

PRACTICAL FOURIER ANALYSIS FOR MULTIGRID METHODS

Numerical Insights

Series Editor

A. Sydow, GMD-FIRST, Berlin, Germany

Editorial Board

P. Borne, École de Lille, France; G. Carmichael, University of Iowa, USA;
L. Dekker, Delft University of Technology, The Netherlands; A. Iserles, University
of Cambridge, UK; A. Jakeman, Australian National University, Australia;
G. Korn, Industrial Consultants (Tucson), USA; G.P. Rao, Indian Institute of
Technology, India; J.R. Rice, Purdue University, USA; A.A. Samarskii, Russian
Academy of Science, Russia; Y. Takahara, Tokyo Institute of Technology, Japan

The Numerical Insights series aims to show how numerical simulations provide valuable insights into the mechanisms and processes involved in a wide range of disciplines. Such simulations provide a way of assessing theories by comparing simulations with observations. These models are also powerful tools which serve to indicate where both theory and experiment can be improved.

In most cases the books will be accompanied by software on disk demonstrating working examples of the simulations described in the text.

The editors will welcome proposals using modelling, simulation and systems analysis techniques in the following disciplines: physical sciences; engineering; environment; ecology; biosciences; economics.

Volume 1

Numerical Insights into Dynamic Systems: Interactive Dynamic System Simulation with Microsoft® Windows™ and NT™

Granino A. Korn

Volume 2

Modelling, Simulation and Control of Non-Linear Dynamical Systems: An Intelligent Approach using Soft Computing and Fractal Theory

Patricia Melin and Oscar Castillo

Volume 3

Principles of Mathematical Modeling: Ideas, Methods, Examples

A.A. Samarskii and A. P. Mikhailov

Volume 4

Practical Fourier Analysis for Multigrid Methods

Roman Wienands and Wolfgang Joppich

PRACTICAL FOURIER ANALYSIS FOR MULTIGRID METHODS

**Roman Wienands
Wolfgang Joppich**



CHAPMAN & HALL/CRC

A CRC Press Company

Boca Raton London New York Washington, D.C.

Library of Congress Cataloging-in-Publication Data

Wienands, R. (Roman)

Practical Fourier Analysis for multigrid methods / R. Wienands and W. Joppich.

p. cm. — (Numerical insights ; v. 4)

Includes bibliographical references and index.

ISBN 1-58488-492-4 (alk. paper)

1. Fourier analysis. 2. Multigrid methods (Numerical analysis) I. Joppich, W. (Wolfgang), 1950- II. Title. III. Series.

QA403 .S.W54

515'.2433--dc22

2004055113

This book contains information obtained from authentic and highly regarded sources. Reprinted material is quoted with permission, and sources are indicated. A wide variety of references are listed. Reasonable efforts have been made to publish reliable data and information, but the author and the publisher cannot assume responsibility for the validity of all materials or for the consequences of their use.

Neither this book nor any part may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, microfilming, and recording, or by any information storage or retrieval system, without prior permission in writing from the publisher.

The consent of CRC Press does not extend to copying for general distribution, for promotion, for creating new works, or for resale. Specific permission must be obtained in writing from CRC Press for such copying.

Direct all inquiries to CRC Press, 2000 N.W. Corporate Blvd., Boca Raton, Florida 33431.

Trademark Notice: Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation, without intent to infringe.

Visit the CRC Press Web site at www.crcpress.com

© 2005 by Chapman & Hall/CRC Press

No claim to original U.S. Government works

International Standard Book Number 1-58488-492-4

Library of Congress Card Number 2004055113

Printed in the United States of America 1 2 3 4 5 6 7 8 9 0

Printed on acid-free paper

Dedicated to Karl

PREFACE

Mathematicians, scientists, and engineers who have some background in numerical analysis and in solving partial differential equations (PDEs) know that multigrid methods are among the fastest solution methods for PDEs. Especially for elliptic equations, they have been proven to be highly efficient. Due to the permanently increasing power of computers, the application of multigrid is expanding to complex systems of PDEs and three-dimensional problems from various fields of application. All these problems require a careful composition of the multigrid algorithm to be applied. However, for problems with nonelliptic and nonsymmetric features—as they often occur in real-life applications—a general rigorous mathematical theory is not yet available and the correct choice of multigrid components is often far from being trivial. For such situations, local Fourier analysis can be considered as the main analysis tool to obtain *quantitative* convergence estimates and to optimize multigrid components like smoothers or intergrid transfer operators.

There are already several tutorials and books on multigrid. Why do we dare to write another book in this context? The answer is simply based on the fact that most books do not provide with software that confirms the written statements about convergence and efficiency of an algorithm and—even more important—that can be easily adapted to the particular application at hand. Usually, paper does not convince the practitioner who has to solve a very concrete problem. Before writing a multigrid code, questions to be posed are: *Does the development pay out? Will multigrid work for my application? What are the numerical properties? How good is convergence?* These questions have one goal in common—to ensure a successful development. Considering multigrid algorithms, there is an enormous degree of freedom in choosing the algorithmic components. This complicates the naive use of multigrid. On the other hand, many of the standard multigrid components work well, even in complex situations. To get rid of the uncertainty—which approach will work and which one will not—a theoretically based technique can provide the practical proof of multigrid applicability or multigrid failure. Based on the framework of local Fourier analysis, it is the intention of this book to describe what is needed for multigrid algorithms in a realistic simulation environment. Our systematic and well-elaborated description of Fourier k -grid ($k = 1, 2, 3$) analysis both for scalar equations and systems of equations including three-dimensional applications provides the required framework to answer the above questions. Especially the recently developed generalization of the classical two-grid analysis to the Fourier three-grid analysis yields additional valuable

insights to real *multigrid* behavior that cannot be captured by a standard two-grid analysis. This might be of great benefit for optimizing a particular algorithm and debugging its implementation.

The Fourier analysis software **LFA** and the graphical user interface (GUI) **xlfa** to use the software efficiently are unique worldwide. Local Fourier analysis is sometimes considered as very technical and tedious, especially for systems of equations and three-dimensional equations. Using our software it can be performed by a simple mouse click. The book and the accompanying software provide both the theoretical background and the necessary link between theory and practice. Because the book and the software establish a bridge from theory to application, the target group is a mixture of theoretically and practically oriented researchers, not limited to multigrid specialists. It is expected to range from interested students on a beginner's level to responsible persons for projects on computer simulation. Summarizing, this book is intended as a companion to the basic multigrid literature with a focus on *quantitative* convergence estimates which are crucial for the development of new multigrid software.

According to the idea of establishing a bridge from theory to application we have structured the presentation of the material such that we combine the learning by reading and the learning by doing (that is, by applying our software) as soon as possible. This is reflected in the overall structure: The first part of the book (**Part I**: Chapters 1-4) provides all the knowledge which is necessary to understand the basic principles of multigrid and local Fourier analysis and to efficiently use the software.

Chapter 1 introduces our notation and the two main ingredients of multigrid: smoothing and coarse-grid correction. The multigrid idea is motivated by a first application of local Fourier analysis to a simple model problem. Moreover, we take a first look at the accompanying software illustrated by the GUI.

Chapter 2 presents the theoretical basis of local Fourier analysis without going into technical details. Possible limits to this approach are discussed. The *practical* relevance and the *reliability* of the analysis for concrete problems are of importance, especially for developers of multigrid software which compose and debug their algorithms using our software. Hence, it seems reasonable to start the discussion of applicability and limits already at this place.

Chapter 3 describes multigrid methods and the difficulty of appropriately choosing their components. In particular, all implemented types of discretization approaches, relaxation methods, coarsening strategies, and transfer operators are discussed in some detail. For a more basic presentation of multigrid, we refer to the well-known literature [7, 13, 28, 60, 65] and especially to [62] for a recent monograph. The descriptions are supported by the GUI, which is presented now in more detail, in order to make the reader familiar with

its functionality. It is shown, how to select the different components of the algorithm which are described in Sections 3.4 and 3.5. This interaction with our software prepares the reader for the following case studies.

Chapter 4 contains case studies for two- and three-dimensional problems. The anisotropic diffusion equation with different discretizations, Helmholtz equation, biharmonic equation, biharmonic system, Stokes equations, Oseen equations, elasticity system, and a shell problem are only a few to mention. For some of them, the effect of differently choosing components is discussed in a systematic manner. The influence of cycle type, discretization, coarsening strategy, or relaxation methods are investigated. All results like convergence rates, norm values of two- and three- level operators, distributions of eigenvalues, etc., are illustrated by the GUI.

So far, the first part provided the skills to exploit the full power of the accompanying software, which has been used up to the end of Chapter 4 as a black-box tool. The second part (**Part II**: Chapters 5-7) describes the theory hidden in the software and is important for those readers who want to understand the details.

More precisely, in **Chapter 5** we focus on the relaxation method in a multigrid cycle. Here the Fourier smoothing analysis is described for general d -dimensional systems of equations including multistage smoothing. Moreover, the measure of h -ellipticity is introduced, which plays an important role for the development of efficient relaxation methods. Chapter 5 can be seen as a one-grid analysis as it solely takes the fine grid operators—discretization and relaxation—into account.

The analysis for the remaining multigrid components is discussed in **Chapter 6** where we present a general description of Fourier two- and three-grid analysis. For example, semicoarsening strategies in two and three dimensions are considered as well as systems of equations. With a three-grid analysis it is possible to investigate real *multigrid* effects, for example, the different behavior of different cycle strategies or pre- and postsmoothing. Moreover, the three-grid analysis allows for a more reliable investigation of possible coarse-grid correction difficulties.

Chapter 7 presents some further applications of local Fourier analysis which are not (yet) implemented in **LFA**, like the analysis of cell-centered multigrid or a “simplified” Fourier k -grid analysis. This chapter is mainly intended for multigrid experts.

The core of this work has been the Ph.D. thesis of R. Wienands [67], which provided both the theory and the LFA software. The practical experience of teaching and applying multigrid by W. Joppich lead to the requirement of supporting the theory by a user-friendly framework as it is provided now by the GUI. This combination of different intentions and approaches to multigrid is reflected by the authors and we have tried to exploit the best of each for this book.

Acknowledgments: We first have to mention Eike Müller, who visited the Institute for practical training before starting his studies. He was encouraged by W. Joppich to learn the standard multigrid methods and to develop a graphical user interface for easy use of the multigrid program he wrote. Exploiting the experience from this exercise the presented GUI for the local Fourier analysis programs of R. Wienands has been built step by step. It was a great pleasure for us to have such a skillful and motivated student within this project. His documentation for installation and use of the GUI is contained on the accompanying CD-ROM to this book.

We greatly admire our families for their patience while we worked on this book. Without their acceptance and support this project would have been impossible.

Köln, Königswinter

Roman Wienands
Wolfgang Joppich

Contents

Symbol Description	xv
I Practical Application of LFA and xlf	1
1 INTRODUCTION	3
1.1 SOME NOTATION	4
1.1.1 Boundary value problems	4
1.1.2 Discrete boundary value problems	5
1.1.3 Stencil notation	6
1.1.4 Systems of partial differential equations	9
1.1.5 Operator versus matrix notation	11
1.2 BASIC ITERATIVE SCHEMES	12
1.3 A FIRST DISCUSSION OF FOURIER COMPONENTS . .	13
1.3.1 Empirical calculation of convergence factors	13
1.3.2 Convergence analysis for the Jacobi method	14
1.3.3 Smoothing properties of Jacobi relaxation	16
1.4 FROM RESIDUAL CORRECTION TO COARSE-GRID CORRECTION	19
1.5 MULTIGRID PRINCIPLE AND COMPONENTS	20
1.6 A FIRST LOOK AT THE GRAPHICAL USER INTERFACE	22
2 MAIN FEATURES OF LOCAL FOURIER ANALYSIS FOR MULTIGRID	29
2.1 THE POWER OF LOCAL FOURIER ANALYSIS	29
2.2 BASIC IDEAS	30
2.2.1 Main goal	30
2.2.2 Necessary simplifications for the discrete problem . . .	31
2.2.3 Crucial observation	31
2.2.4 Arising questions	31
2.3 APPLICABILITY OF THE ANALYSIS	32
2.3.1 Type of partial differential equation	33
2.3.2 Type of grid	33
2.3.3 Type of discretization	34

3	MULTIGRID AND ITS COMPONENTS IN LFA	35
3.1	MULTIGRID CYCLING	35
3.1.1	Coarse-grid correction operator	35
3.1.2	Aliasing of Fourier components	36
3.1.3	Correction scheme	37
3.2	FULL MULTIGRID	40
3.3	xlfa FUNCTIONALITY—AN OVERVIEW	42
3.3.1	Menu bar	42
3.3.2	Button bar	43
3.3.3	Parameter display	43
3.3.4	Problem display	44
3.4	IMPLEMENTED COARSE-GRID CORRECTION COMPONENTS	44
3.4.1	Discretization and grid structure	45
3.4.2	Coarsening strategies	46
3.4.3	Coarse-grid operator	46
3.4.4	Multigrid cycling	48
3.4.5	Restriction	49
3.4.6	Prolongation	50
3.5	IMPLEMENTED RELAXATIONS	51
3.5.1	Relaxation type and ordering of grid points	51
3.5.2	Relaxation methods for systems	54
3.5.3	Multistage (MS) relaxations	55
4	USING THE FOURIER ANALYSIS SOFTWARE	57
4.1	CASE STUDIES FOR 2D SCALAR PROBLEMS	59
4.1.1	Anisotropic diffusion equation: second-order discretization	59
4.1.2	Anisotropic diffusion equation: fourth-order discretization	65
4.1.3	Anisotropic diffusion equation: Mehrstellen discretization	67
4.1.4	Helmholtz equation	69
4.1.5	Biharmonic equation	69
4.1.6	Rotated anisotropic diffusion equation	70
4.1.7	Convection diffusion equation: first-order upwind discretization	73
4.1.8	Convection diffusion equation: higher-order upwind discretization	76
4.2	CASE STUDIES FOR 3D SCALAR PROBLEMS	77
4.2.1	Anisotropic diffusion equation: second-order discretization	77
4.2.2	Anisotropic diffusion equation: fourth-order discretization	82

4.2.3	Anisotropic diffusion equation: Mehrstellen discretization	82
4.2.4	Helmholtz equation	83
4.2.5	Biharmonic equation	83
4.2.6	Convection diffusion equation: first-order upwind discretization	83
4.3	CASE STUDIES FOR 2D SYSTEMS OF EQUATIONS	84
4.3.1	Biharmonic system	84
4.3.2	Stokes equations	86
4.3.3	First-order discretization of the Oseen equations	86
4.3.4	Higher-order discretization of the Oseen equations	91
4.3.5	Elasticity system	93
4.3.6	A linear shell problem	93
4.4	CREATING NEW APPLICATIONS	94

II The Theory behind LFA 97

5	FOURIER ONE-GRID OR SMOOTHING ANALYSIS	99
5.1	ELEMENTS OF LOCAL FOURIER ANALYSIS	100
5.1.1	Basic definitions	100
5.1.2	Generalization to systems of PDEs	102
5.2	HIGH AND LOW FOURIER FREQUENCIES	103
5.2.1	Standard and semicoarsening	103
5.2.2	Red-black coarsening and quadrupling	104
5.3	SIMPLE RELAXATION METHODS	105
5.3.1	Jacobi relaxation	107
5.3.2	Lexicographic Gauss-Seidel relaxation	108
5.3.3	A first definition of the smoothing factor	110
5.4	PATTERN RELAXATIONS	113
5.4.1	Red-black Jacobi (RB-JAC) relaxations	114
5.4.2	Spaces of $2h$ -harmonics	115
5.4.3	Auxiliary definitions and relations	118
5.4.4	Fourier representation for RB-JAC point relaxation	120
5.4.5	General definition of the smoothing factor	123
5.4.6	Red-black Gauss-Seidel (RB-GS) relaxations	127
5.4.7	Multicolor relaxations	128
5.5	SMOOTHING ANALYSIS FOR SYSTEMS	129
5.5.1	Collective versus decoupled smoothing	129
5.5.2	Distributive relaxation	132
5.6	MULTISTAGE (MS) RELAXATIONS	134
5.7	FURTHER RELAXATION METHODS	138
5.8	THE MEASURE OF h -ELLIPTICITY	139
5.8.1	Example 1: anisotropic diffusion equation	141
5.8.2	Example 2: convection diffusion equation	143
5.8.3	Example 3: Oseen equations	145

6	FOURIER TWO- AND THREE-GRID ANALYSIS	147
6.1	BASIC ASSUMPTIONS	148
6.2	TWO-GRID ANALYSIS FOR 2D SCALAR PROBLEMS . .	149
6.2.1	Spaces of $2h$ -harmonics	149
6.2.2	Fourier representation of fine-grid discretization	151
6.2.3	Fourier representation of restriction	151
6.2.4	Fourier representation of prolongation	152
6.2.5	Fourier representation of coarse-grid discretization . .	158
6.2.6	Invariance property of the two-grid operator	160
6.2.7	Definition of the two-grid convergence factor	161
6.2.8	Semicoarsening	163
6.3	TWO-GRID ANALYSIS FOR 3D SCALAR PROBLEMS . .	169
6.3.1	Standard coarsening	169
6.3.2	Semicoarsening	171
6.4	TWO-GRID ANALYSIS FOR SYSTEMS	173
6.5	THREE-GRID ANALYSIS	176
6.5.1	Spaces of $4h$ -harmonics	177
6.5.2	Invariance property of the three-grid operator	179
6.5.3	Definition of three-grid convergence factor	180
6.5.4	Generalizations	181
7	FURTHER APPLICATIONS OF LOCAL FOURIER ANALYSIS	183
7.1	ORDERS OF TRANSFER OPERATORS	184
7.1.1	Polynomial order	184
7.1.2	High- and low-frequency order	185
7.2	SIMPLIFIED FOURIER k -GRID ANALYSIS	187
7.3	CELL-CENTERED MULTIGRID	189
7.3.1	Transfer operators	191
7.3.2	Fourier two- and three-grid analysis	192
7.3.3	Orders of transfer operators	194
7.3.4	Numerical experiments	195
7.4	FOURIER ANALYSIS FOR MULTIGRID PRECONDITIONED BY GMRES	197
7.4.1	Analysis based on the GMRES(m)-polynomial	199
7.4.2	Analysis based on the spectrum of the residual transformation matrix	200
A	FOURIER REPRESENTATION OF RELAXATION	203
A.1	Two-dimensional case	204
A.2	Three-dimensional case	204
	REFERENCES	207

Symbol Description

Boundary value problems:

$\Omega, \Gamma, \Omega_h, \Gamma_h$	d -dimensional domain, boundary, and discrete analogs with mesh size h
L, L_h, B, B_h	continuous and discrete differential and boundary operator
$[\ell_\kappa]_h$	stencil representing L_h , Section 1.1.3
L	Matrix related to L_h
u, u_h	continuous and discrete solution
u	vector related to u_h
$\partial_{j_1 \dots j_n}$	partial differential operator
$D_{h_j}^j, \dots, D_{h_j}^{jjj}$	central discretizations, Example 1.2
$[d_{\kappa_j}^j]_{h_j}, \dots$	corresponding stencils, Example 1.2
u_h, v_h, w_h, f_h	grid functions
$\mathcal{F}(\Omega_h)$	space of grid functions
$\langle v_h, w_h \rangle_{\Omega_h}$	discrete Euclidean inner product
$\ v_h\ _{\Omega_h}$	corresponding norm
$u_h^{(i)}$	approximation of u_h after i steps of an iterative method
$e_h^{(i)}, r_h^{(i)}$	algebraic error and residual

Multigrid:

S_h	relaxation
ω	relaxation or damping parameter
ω_n	multistage parameter
C_h	distributor for distributive relaxation
P_h^h	prolongation
R_h^H	restriction

L_H	coarse-grid discretization
K_h^H	coarse-grid correction operator (3.1)
M_h^H	two-grid operator (3.2)
γ	cycle index

Fourier analysis:

G_h	d -dimensional infinite grid (1.3) with mesh size vector h
i	complex unit $\sqrt{-1}$
$\alpha = (\alpha_1, \dots, \alpha_d)$	multi-index, $\alpha_j = 0, 1$ ($j = 1, \dots, d$)
θ	Fourier frequency, Definition 5.1
$\varphi_h(\theta, \mathbf{x})$	Fourier component, Definition 5.1
$\tilde{L}_h(\theta)$	Fourier symbol for discretization, Definition 5.1
\mathcal{F}	Fourier space (5.4)
\mathcal{F}_{h_θ}	finite dimensional Fourier space (5.11)
$\Theta_{\text{high}}, \Theta_{\text{low}}$	sets of high and low frequencies, Section 5.2
$A(\theta, \omega)$	Fourier symbol for Jacobi- and Gauss-Seidel-type relaxations (5.7)
$\tilde{S}_h(\theta, \omega)$	Fourier representation of relaxation
Q_h^H	ideal coarse-grid correction operator (5.25)
$\rho_1(\nu, \omega)$	general definition of the smoothing factor, Definition 5.9
$\rho(S^\nu Q)$	related LFA output (5.28)

$\rho_1^D(\nu, \omega)$	smoothing factor referring to Dirichlet boundary conditions, Definition 5.9	$\tilde{P}_H^h(\boldsymbol{\theta})$	restriction Fourier symbol for prolongation
$\rho(S^\nu Q)_D$	related LFA output (5.28)	$\rho(M_{2L})$	two-grid convergence factor (6.22)
$E_h(L_h)$	measure of h -ellipticity, Definition 5.10	$ M_{2L} _s$	spectral norm of the two-grid error-reduction operator
$\mathcal{F}_{2h}(\boldsymbol{\theta})$	spaces of $2h$ -harmonics, Definition 5.8	$ M_{2L} _d$	spectral norm of the two-grid residual-reduction operator
$\mathcal{F}_{4h}(\boldsymbol{\theta})$	spaces of $4h$ -harmonics, Definition 6.3	$\rho(M_{3L})$	three-grid convergence factor (6.36)
$\mathcal{F}_{2h}^{4h}(\boldsymbol{\theta})$	spaces of $(2h, 4h)$ -harmonics, Definition 6.4	$ M_{3L} _s$	spectral norm of the three-grid error-reduction operator
$\tilde{R}_h^H(\boldsymbol{\theta})$	Fourier symbol for	$ M_{3L} _d$	spectral norm of the three-grid residual-reduction operator

Part I

Practical Application of
LFA and xlfa

Chapter 1

INTRODUCTION

This introductory chapter attempts to give a first feeling for

- local Fourier analysis,
- multigrid methods,
- and the application of the accompanying software **LFA** with the help of the graphical user interface (GUI) **xlfa**.

In this respect, it can be considered as a *preview* and guideline for the following chapters where the above ingredients are discussed in detail. More precisely, Section 1.3 provides a short discussion of Fourier components and a first convergence analysis is performed for a model problem. In this way, the basic idea of local Fourier analysis is introduced. It is revisited in [Chapter 2](#) in a more general framework with an emphasis on the main principles of local Fourier analysis omitting technical details. A systematic mathematical description of local Fourier k -grid analysis for general situations is finally given in the second part of this book ([Chapters 5-7](#)).

The analysis and observations for the above-mentioned introductory model problem in combination with the coarse-grid correction idea presented in Section 1.4 naturally lead to the development of multigrid algorithms. Section 1.5 presents the multigrid principle and possible components without lots of theoretical derivations but as an obvious consequence of the analysis presented in Section 1.3. A more detailed description of multigrid and the components that are implemented in **LFA** is given in [Chapter 3](#).

The graphical user interface supports the proper understanding of algorithmic details. Consequently, Section 1.6 offers a first look at the GUI and the software behind it. All multigrid components can be selected easily just by using the buttons of the mouse. A first case study for the Poisson equation illustrates how to use our Fourier analysis tool exploring the effect of different multigrid components on the numerical behavior of the resulting algorithms. A large variety of two- and three-dimensional test cases including systems of equations can be found in [Chapter 4](#).

xlfa is written in tcl/tk, an interpreted scripting language, whereas **LFA** is a Fortran77 code. Hence, all you need to start with the experiments is an appropriate interpreter (which is freely available for nearly every UNIX, LINUX, or Windows-like operating system) and a Fortran compiler. More information

on installation requirements and installation steps are given in the corresponding manuals which can be found on the accompanying CD-ROM.

Section 1.1 starts with a short introduction to boundary value problems and their discrete description to clarify our notation. The stencil terminology is introduced for scalar equations and for systems of PDEs. Basic iterative schemes are presented in Section 1.2 as far as it is required for understanding the following motivation of multigrid methods.

1.1 SOME NOTATION

Multigrid is commonly explained and analyzed using operators whereas basic iterative methods are often described with matrices, compare with Section 1.2. For our presentation, it is convenient to combine operator, matrix, and stencil notation. To distinguish between operators and matrices, an operator is denoted by a subscript h in contrast to the corresponding matrix. Similarly, we distinguish grid functions from their related vector representations. A short overview of our notation is given in Section 1.1.5.

1.1.1 Boundary value problems

A (linear) boundary value problem on a d -dimensional domain Ω with boundary $\partial\Omega$ reads

$$\begin{aligned} Lu(\mathbf{x}) &= f(\mathbf{x}) \quad \text{on } \Omega \subset \mathbb{R}^d \\ Bu(\mathbf{x}) &= g(\mathbf{x}) \quad \text{on } \Gamma := \partial\Omega \end{aligned} \tag{1.1}$$

with $\mathbf{x} = (x_1, \dots, x_d)^T$ and given functions

$$f : \Omega \longrightarrow \mathbb{R}, \quad g : \Gamma \longrightarrow \mathbb{R}.$$

L denotes a (linear) differential operator on Ω and B represents one or several boundary operators according to the boundary conditions under consideration. In order to simplify the notation of partial differential operators we use the following terminology:

$$\begin{aligned} \partial_{j_1 \dots j_n} &:= \frac{\partial^n}{\partial x_{j_1} \dots \partial x_{j_n}} \quad \text{and} \quad u_{j_1 \dots j_n}(\mathbf{x}) := \frac{\partial^n u(\mathbf{x})}{\partial x_{j_1} \dots \partial x_{j_n}} \\ &\text{for } j_1, \dots, j_n \in \{1, \dots, d\} \text{ and } n \in \mathbb{N}. \end{aligned}$$

This yields, for example, $\partial^2/\partial x_1^2 = \partial_{11}$ or $\partial^2 u(\mathbf{x})/(\partial x_1 \partial x_2) = u_{12}(\mathbf{x})$. For the well-known classification (*elliptic*, *parabolic*, *hyperbolic*) of partial differential operators, we refer to the literature, e.g., [25]. In this book we focus on purely

elliptic and singularly perturbed problems. Here, a singular perturbation means that the type of the differential operator changes subject to a certain choice of its parameters.

Example 1.1 (Anisotropic diffusion equation). A prominent example for an elliptic boundary value problem is the anisotropic diffusion equation:

$$\begin{aligned} Lu(\mathbf{x}) &:= - \sum_{j=1}^d \varepsilon_j u_{jj}(\mathbf{x}) = f(\mathbf{x}) \quad \text{on } \Omega \quad \text{with } \varepsilon_j \geq 0 \quad (j = 1, \dots, d) \\ Bu(\mathbf{x}) &= g(\mathbf{x}) \quad \text{on } \Gamma. \end{aligned}$$

For $\varepsilon_1 = \dots = \varepsilon_d = 1$, we obtain the Poisson equation which is the standard test case in the classical multigrid literature. If one or several coefficients ε_j tend to zero or to infinity, the type of L changes from elliptic to parabolic. ◀

1.1.2 Discrete boundary value problems

As we are interested in a numerical solution of (1.1), we search for a discrete approximation $u_{\mathbf{h}}$ of the continuous solution u which satisfies the discrete analogue of the boundary value problem (1.1),

$$\begin{aligned} L_{\mathbf{h}} u_{\mathbf{h}}(\mathbf{x}) &= f_{\mathbf{h}}(\mathbf{x}) \quad \text{on } \Omega_{\mathbf{h}} := \Omega \cap G_{\mathbf{h}} \\ B_{\mathbf{h}} u_{\mathbf{h}}(\mathbf{x}) &= g_{\mathbf{h}}(\mathbf{x}) \quad \text{on } \Gamma_{\mathbf{h}} := \partial\Omega \cap G_{\mathbf{h}}, \end{aligned} \tag{1.2}$$

with a d -dimensional vertex-centered infinite grid

$$\begin{aligned} G_{\mathbf{h}} &:= \{ \mathbf{x} = (x_1, \dots, x_d)^T = \boldsymbol{\kappa} \mathbf{h} = (\kappa_1 h_1, \dots, \kappa_d h_d)^T \mid \boldsymbol{\kappa} \in \mathbb{Z}^d \} \\ \text{where } h_1 &= \frac{1}{n_1}, \dots, h_d = \frac{1}{n_d} \quad (n_1, \dots, n_d \in \mathbb{N}). \end{aligned} \tag{1.3}$$

n_1, \dots, n_d are assumed to be even, which is a natural choice in connection with multigrid methods, see [Chapter 3](#). The components of $\mathbf{h} \in \mathbb{R}^d$ denote the mesh sizes of $G_{\mathbf{h}}$ into the different space directions. For many of the problems considered here, the mesh sizes with respect to the different directions are identical. In this case we refer to the scalar quantity $h (= 1/n)$ instead of \mathbf{h} . The notation \mathbf{h} is only used if necessary, for example, in connection with semicoarsening to be defined below. $L_{\mathbf{h}}$ and $B_{\mathbf{h}}$ are discrete differential or boundary operators related to L and B . The grid functions $u_{\mathbf{h}}$, $f_{\mathbf{h}}$, and $g_{\mathbf{h}}$ are the restrictions of the continuous counterparts from (1.1) to the discrete domain $\Omega_{\mathbf{h}}$ (or the discrete boundary $\Gamma_{\mathbf{h}}$), for example, $g_{\mathbf{h}} := g|_{\Omega_{\mathbf{h}}}$. In practice, the discrete boundary conditions are often eliminated from (1.2). Then, we simply replace (1.2) by

$$L_{\mathbf{h}} u_{\mathbf{h}}(\mathbf{x}) = f_{\mathbf{h}}(\mathbf{x}) \quad \text{on } \Omega_{\mathbf{h}}.$$

We define the space of all grid functions that act on the discrete domain Ω_h by

$$\mathcal{F}(\Omega_h) := \{v_h \mid v_h(\mathbf{x}) : \Omega_h \longrightarrow \mathbb{C}\}.$$

With the usual addition and scalar multiplication we have a linear vector space. It can be endowed with a scaled, discrete Euclidean inner product,

$$\langle v_h, w_h \rangle_{\Omega_h} := \frac{1}{\#\Omega_h} \sum_{\mathbf{x} \in \Omega_h} v_h(\mathbf{x}) \overline{w_h(\mathbf{x})} \quad (v_h, w_h \in \mathcal{F}(\Omega_h)), \quad (1.4)$$

where $\#\Omega_h$ is the number of grid points of Ω_h , leading to a Hilbert space. The induced norm is $\|v_h\|_{\Omega_h} := \sqrt{\langle v_h, v_h \rangle_{\Omega_h}}$. The corresponding operator norm is the spectral norm given by $\|L_h\|_S := \sqrt{\rho(L_h L_h^*)}$ where L_h^* is the adjoint of L_h and $\rho(S_h)$ denotes the spectral radius of a discrete operator S_h . In practice, one often applies the infinity norm

$$\|v_h\|_{\infty} := \max \{|v_h(\mathbf{x})| : \mathbf{x} \in \Omega_h\}$$

whereas the discrete Euclidean norm is especially suited for theoretical considerations.

1.1.3 Stencil notation

For a Cartesian or logically rectangular grid G_h , it is appropriate to define discrete differential operators using the stencil terminology [55], which is most easily described for infinite grid functions $u_h : G_h \longrightarrow \mathbb{R}$. This formulation is particularly convenient in the context of the local Fourier analysis, compare with Section 2.2.2. For a fixed grid point $\mathbf{x} \in G_h$, we can define an operator L_h on the space of infinite grid functions by

$$L_h u_h(\mathbf{x}) = \sum_{\kappa \in J} \ell_{\kappa} u_h(\mathbf{x} + \kappa \mathbf{h}) \quad (1.5)$$

with stencil coefficients $\ell_{\kappa} \in \mathbb{R}$ and a certain finite subset $J \subset \mathbb{Z}^d$ containing $(0, \dots, 0)$.

As a two-dimensional example, consider the well-known central approximation of $Lu(\mathbf{x}) = u_{11}(\mathbf{x})$, which is given by

$$\frac{1}{h^2} (u(x_1 - h, x_2) - 2u(x_1, x_2) + u(x_1 + h, x_2)) \quad (1.6)$$

for a fixed grid point $\mathbf{x} = (x_1, x_2) \in G_h$. The central point \mathbf{x} may be written as $\mathbf{x} + \kappa \mathbf{h}$ with $\kappa = (0, 0)$. Similarly, $(x_1 + h, x_2)$ and $(x_1 - h, x_2)$ correspond to $\mathbf{x} + \kappa \mathbf{h}$ with $\kappa = (1, 0)$ and $\kappa = (-1, 0)$, respectively. Hence, we have $\ell_{(-1,0)} = 1/h^2$, $\ell_{(0,0)} = -2/h^2$, and $\ell_{(1,0)} = 1/h^2$. We write this in a more compact way by collecting the coefficients as a one-dimensional “stencil”

$$L_h \hat{=} [\ell_{(-1,0)} \quad \ell_{(0,0)} \quad \ell_{(1,0)}]_h = \frac{1}{h^2} [1 \quad -2 \quad 1]_h.$$

Applying formula (1.5) with $L_h \triangleq \frac{1}{h^2} [1 \quad -2 \quad 1]_h$ yields (1.6).

Thus, a discrete differential operator L_h is uniquely defined for every fixed $\mathbf{x} \in G_h$ by a stencil $[\ell_{\kappa}]_h$, which is commonly written as

$$L_{h_1} \triangleq [\ell_{\kappa_1}]_{h_1} = [\cdots \ell_{-1} \ell_0 \ell_1 \cdots]_{h_1}, \quad L_{h_2} \triangleq [\ell_{\kappa_2}]_{h_2} = \begin{bmatrix} \vdots \\ \ell_1 \\ \ell_0 \\ \ell_{-1} \\ \vdots \end{bmatrix}_{h_2}$$

$$\text{or } L_h \triangleq [\ell_{\kappa}]_h = \begin{bmatrix} \vdots & \vdots & \vdots \\ \cdots \ell_{(-1,1)} \ell_{(0,1)} \ell_{(1,1)} \cdots \\ \cdots \ell_{(-1,0)} \ell_{(0,0)} \ell_{(1,0)} \cdots \\ \cdots \ell_{(-1,-1)} \ell_{(0,-1)} \ell_{(1,-1)} \cdots \\ \vdots & \vdots & \vdots \end{bmatrix}_h$$

in one or two dimensions, respectively. The generalization to higher dimensions is straightforward. For the representation of three-dimensional stencils, we use several two-dimensional stencils in the following way:

$$[\ell_{\kappa}]_h = [\cdots [\ell_{\kappa}]_h^{(-1)} [\ell_{\kappa}]_h^{(0)} [\ell_{\kappa}]_h^{(1)} \cdots].$$

Here, $[\ell_{\kappa}]_h^{(\kappa_3)}$ ($\kappa_3 \in \{\dots, -1, 0, 1 \dots\}$) denotes a two-dimensional stencil where the position in the third space direction (κ_3) is fixed, i.e.,

$$[\ell_{\kappa}]_h^{(\kappa_3)} = \begin{bmatrix} \vdots & \vdots & \vdots \\ \cdots \ell_{(-1,1,\kappa_3)} \ell_{(0,1,\kappa_3)} \ell_{(1,1,\kappa_3)} \cdots \\ \cdots \ell_{(-1,0,\kappa_3)} \ell_{(0,0,\kappa_3)} \ell_{(1,0,\kappa_3)} \cdots \\ \cdots \ell_{(-1,-1,\kappa_3)} \ell_{(0,-1,\kappa_3)} \ell_{(1,-1,\kappa_3)} \cdots \\ \vdots & \vdots & \vdots \end{bmatrix}_h.$$

Only a finite number $\#J$ ($=$ number of elements of J) of stencil elements ℓ_{κ} is nonzero. The zero elements are usually omitted in the above notation.

Example 1.2 (Central discretizations, discrete Laplace operator). If the partial differential operators ∂_1 , ∂_{11} , ∂_{111} , and ∂_{1111} are discretized by central differences, the corresponding discrete differential operators $D_{h_1}^1$, $D_{h_1}^{11}$, $D_{h_1}^{111}$, and $D_{h_1}^{1111}$, respectively, are given by the following stencils:

$$\begin{aligned} [d_{\kappa_1}^1]_{h_1} &= \frac{1}{2h_1} [-1 \ 0 \ 1]_{h_1}, & [d_{\kappa_1}^{11}]_{h_1} &= \frac{1}{h_1^2} [1 \ -2 \ 1]_{h_1}, \\ [d_{\kappa_1}^{111}]_{h_1} &= \frac{1}{2h_1^3} [-1 \ 2 \ 0 \ -2 \ 1]_{h_1}, & [d_{\kappa_1}^{1111}]_{h_1} &= \frac{1}{h_1^4} [1 \ -4 \ 6 \ -4 \ 1]_{h_1}. \end{aligned}$$

The stencil notations of the central discretizations of ∂_j , ∂_{jj} , ∂_{jjj} , and ∂_{jjjj} for $j = 2, \dots, d$ can be obtained straightforwardly. The standard 5- and 7-point discretizations of the two- and three-dimensional Laplace operator with $h = h_1 = h_2 = h_3$ read

$$D_h^1 + D_h^2 = \Delta_h \triangleq \frac{1}{h^2} [1 \ -2 \ 1]_h + \frac{1}{h^2} \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}_h = \frac{1}{h^2} \begin{bmatrix} 1 & & \\ & -4 & \\ & & 1 \end{bmatrix}_h \quad (1.7)$$

$$\text{and } D_h^1 + D_h^2 + D_h^3 = \Delta_h \triangleq \frac{1}{h^2} \begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix}_h \begin{bmatrix} 1 & -6 & 1 \\ & 1 & \\ & & 1 \end{bmatrix}_h \begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix}_h \quad (1.8)$$

respectively. The discrete operators (1.7) and (1.8) are of second-order accuracy, i.e., with Taylor's expansion one can establish a consistency relation of the form

$$Lu(\mathbf{x}) - L_h u(\mathbf{x}) = \sum_{j=1}^d O(h_j^2) \quad \text{for } h_1, \dots, h_d \rightarrow 0 \quad (\mathbf{x} \in \Omega_h)$$

and a sufficiently smooth function u . ◀

Of course, the introduced stencil notation is not limited to symmetric cases as shown above, where the central point referring to $\kappa = (0, 0)$ is located at the central position of the stencil. To indicate the position of the central point within a nonsymmetric stencil, the corresponding coefficient is usually underlined. For example, a one-sided upwind discretization of $Lu(\mathbf{x}) = u_1(\mathbf{x})$ is given by

$$\frac{1}{h} (u(x_1, x_2) - u(x_1 - h, x_2)).$$

The corresponding stencil reads

$$L_h \triangleq \frac{1}{h} [\ell_{(-1,0)} \ \underline{\ell}_{(0,0)}] = \frac{1}{h} [-1 \ \underline{1}].$$

Remark 1.1 (Calculation with stencils). Suppose that the stencil notations of two discrete operators S_h and T_h are given by $[s_\kappa]_h$ and $[t_\kappa]_h$ for a fixed $\mathbf{x} \in G_h$. Then, the stencil elements of $\alpha S_h \pm \beta T_h$ ($\alpha, \beta \in \mathbb{R}$) and $S_h T_h$ can be calculated by

$$\alpha s_\kappa \pm \beta t_\kappa \quad \text{and} \quad \sum_{\nu \in \mathbb{Z}^d} s_\nu t_{\kappa - \nu} \quad (\kappa \in \mathbb{Z}^d), \quad \text{respectively.} \quad (1.9)$$

Furthermore, it is possible to define a tensor product \otimes between one-dimensional stencils $[s_{\kappa_j}]_{h_j}$ corresponding to certain one-dimensional discrete infinite grid operators S_{h_j} ($j = 1, \dots, d$) by

$$\bigotimes_{j=1}^d [s_{\kappa_j}]_{h_j} := [s_\kappa]_h \quad \text{with } s_\kappa = \prod_{j=1}^d s_{\kappa_j} \quad (\kappa \in \mathbb{Z}^d).$$

Obviously, the resulting stencil $[s_{\kappa}]_h$ represents a d -dimensional discrete operator S_h .

The stencil that describes the adjoint of an operator $S_h \triangleq [s_{\kappa}]_h$ reads $S_h^* \triangleq [s_{-\kappa}]_h$. ►

Example 1.3 (Stencils for Mixed Derivatives). Stencils for differential operators with mixed derivatives may be calculated by tensor products of one-dimensional stencils. For example, the stencils corresponding to the central discretizations D_h^{12} of ∂_{12} and D_h^{1122} of ∂_{1122} are given by

$$\begin{aligned} [d_{\kappa}^{12}]_h &= [d_{\kappa_1}^1]_{h_1} \otimes [d_{\kappa_2}^2]_{h_2} = \frac{1}{4h_1h_2} \begin{bmatrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{bmatrix}_h, \\ [d_{\kappa}^{1122}]_h &= [d_{\kappa_1}^{11}]_{h_1} \otimes [d_{\kappa_2}^{22}]_{h_2} = \frac{1}{h_1^2h_2^2} \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}_h \end{aligned}$$

with $d_{\kappa}^{12} = d_{\kappa_1}^1 d_{\kappa_2}^2$ ($\kappa \in \mathbb{Z}^2$) and $d_{\kappa}^{1122} = d_{\kappa_1}^{11} d_{\kappa_2}^{22}$ ($\kappa \in \mathbb{Z}^2$), respectively.

For the representation of $[d_{\kappa_1}^1]_{h_1}$, $[d_{\kappa_2}^2]_{h_2}$, $[d_{\kappa_1}^{11}]_{h_1}$, and $[d_{\kappa_2}^{22}]_{h_2}$, see Example 1.2. ◀

Note that most of the discrete differential operators occurring in this monograph can be expressed by sums and tensor products of the central one-dimensional discrete operators $D_{h_j}^j, D_{h_j}^{jj}, D_{h_j}^{jjj}$, and $D_{h_j}^{jjjj}$ ($j = 1, \dots, d$) from Example 1.2.

Remark 1.2 (Operators with variable coefficients). If a discrete operator L_h depends on \mathbf{x} , then its stencil does as well, leading to $[\ell_{\kappa}(\mathbf{x})]_h$. In practice, discrete operators are usually given on the *finite* grid Ω_h . To identify such operators with a stencil, one has to restrict $[\ell_{\kappa}(\mathbf{x})]_h$ to Ω_h , which often forces a modification of $[\ell_{\kappa}(\mathbf{x})]_h$ near the boundaries. Such modifications are obvious for rectangular domains. ►

Apart from discrete differential operators it is, furthermore, possible to specify certain multigrid components (like intergrid transfer operators) using stencil notation, compare with Sections 3.4.5, 6.2.3, and 6.2.4.

1.1.4 Systems of partial differential equations

The generalization of (1.1) to linear $(q \times q)$ -systems of PDEs reads

$$\begin{aligned} Lu(\mathbf{x}) &= \mathbf{f}(\mathbf{x}) \text{ on } \Omega \\ Bu(\mathbf{x}) &= \mathbf{g}(\mathbf{x}) \text{ on } \Gamma \end{aligned} \tag{1.10}$$

with $\mathbf{u} = (u^1, \dots, u^q)^T$, $\mathbf{f} = (f^1, \dots, f^q)^T$, $\mathbf{g} = (g^1, \dots, g^r)^T$,

$$\mathbf{L} = \begin{pmatrix} L^{1,1} & \dots & L^{1,q} \\ \vdots & \dots & \vdots \\ L^{q,1} & \dots & L^{q,q} \end{pmatrix}, \text{ and } \mathbf{B} = \begin{pmatrix} B^{1,1} & \dots & B^{1,q} \\ \vdots & \dots & \vdots \\ B^{r,1} & \dots & B^{r,q} \end{pmatrix}.$$

The $L^{i,j}$ ($i, j = 1, \dots, q$) are scalar differential operators and the $B^{i,j}$ ($i = 1, \dots, r$; $j = 1, \dots, q$) are scalar boundary operators. Note that in general the number of boundary conditions r does not necessarily have to match the number of equations q (for more details see, for example, [62]). The discrete counterpart of (1.10) is given by

$$\begin{aligned} \mathbf{L}_h \mathbf{u}_h(\mathbf{x}) &= \mathbf{f}_h(\mathbf{x}) \quad \text{on } \Omega_h \\ \mathbf{B}_h \mathbf{u}_h(\mathbf{x}) &= \mathbf{g}_h(\mathbf{x}) \quad \text{on } \Gamma_h \end{aligned} \quad (1.11)$$

with $\mathbf{u}_h = (u_h^1, \dots, u_h^q)^T$, $\mathbf{f}_h = (f_h^1, \dots, f_h^q)^T$, $\mathbf{g}_h = (g_h^1, \dots, g_h^r)^T$,

$$\mathbf{L}_h = \begin{pmatrix} L_h^{1,1} & \dots & L_h^{1,q} \\ \vdots & \dots & \vdots \\ L_h^{q,1} & \dots & L_h^{q,q} \end{pmatrix}, \text{ and } \mathbf{B}_h = \begin{pmatrix} B_h^{1,1} & \dots & B_h^{1,q} \\ \vdots & \dots & \vdots \\ B_h^{r,1} & \dots & B_h^{r,q} \end{pmatrix}. \quad (1.12)$$

The vector valued functions \mathbf{u}_h , \mathbf{f}_h , and \mathbf{g}_h consist of scalar grid functions u_h^1, \dots, u_h^q , $f_h^1, \dots, f_h^q \in \mathcal{F}(\Omega_h)$ and $g_h^1, \dots, g_h^r \in \mathcal{F}(\Gamma_h)$. The systems \mathbf{L}_h and \mathbf{B}_h are composed of discrete scalar differential and boundary operators, respectively:

$$\begin{aligned} L_h^{i,j} &: \mathcal{F}(\Omega_h) \longrightarrow \mathcal{F}(\Omega_h) \quad (i, j = 1, \dots, q), \\ B_h^{i,j} &: \mathcal{F}(\Gamma_h) \longrightarrow \mathcal{F}(\Gamma_h) \quad (i = 1, \dots, r; j = 1, \dots, q). \end{aligned}$$

Example 1.4 (Biharmonic system). The central approximation of the biharmonic system

$$\mathbf{L} \mathbf{u}(\mathbf{x}) = \begin{pmatrix} \Delta & 0 \\ -I & \Delta \end{pmatrix} \begin{pmatrix} u^1(\mathbf{x}) \\ u^2(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} f^1(\mathbf{x}) \\ 0 \end{pmatrix}$$

reads

$$\mathbf{L}_h \mathbf{u}_h(\mathbf{x}) = \begin{pmatrix} \Delta_h & 0 \\ -I_h & \Delta_h \end{pmatrix} \begin{pmatrix} u_h^1(\mathbf{x}) \\ u_h^2(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} f_h^1(\mathbf{x}) \\ 0 \end{pmatrix}$$

with discrete identity $I_h \stackrel{\Delta}{=} [1]_h$. ◀

A generalization of the discrete Euclidean inner product (1.4) for vector valued functions

$$\mathbf{v}_h, \mathbf{w}_h \in \mathcal{F}(\Omega_h) := \{\mathbf{v}_h \mid \mathbf{v}_h(\mathbf{x}) : \Omega_h \longrightarrow \mathbb{C}^q\},$$

is defined by

$$\langle \mathbf{v}_h, \mathbf{w}_h \rangle_{\Omega_h}^q := \sum_{j=1}^q \langle v_h^j, w_h^j \rangle_{\Omega_h}.$$

The stencil notation for a $(q \times q)$ -system of scalar discrete operators, denoted by \mathbf{L}_h , is given by $[\ell_{\kappa}]_h$, where $[\ell_{\kappa}]_h$ consists of $q \cdot q$ scalar stencils $[\ell_{\kappa}^{i,j}]_h$ representing the $q \cdot q$ scalar discrete operators $L_h^{i,j}$ ($i, j = 1, \dots, q$). The scalar stencils $[a_{\kappa}^{i,j}]_h$ constituting the adjoint system \mathbf{L}_h^* are given by $[a_{\kappa}^{i,j}]_h = [\ell_{-\kappa}^{j,i}]_h$ for $i, j = 1, \dots, q$. This representation is used in the accompanying software to construct Kaczmarz relaxation methods; see below.

1.1.5 Operator versus matrix notation

After a brief summary of our operator/function notation for scalar equations, we introduce the related matrix/vector terminology. The respective notation is collected in Table 1.1.

Starting from a (linear) boundary value problem (1.1) with continuous differential operator L , boundary operator B , given functions f, g , and solution u , the discrete analogs are denoted by L_h, B_h, f_h, g_h , and discrete solution u_h . The discrete operator will often be defined by a stencil

$$L_h \stackrel{\Delta}{=} [\ell_{\kappa}]_h,$$

as introduced in Section 1.1.3. Then, the discrete problem (with eliminated boundary conditions) can be converted into a large linear system of equations where the discrete operator L_h is represented by a matrix \mathbf{L} and the unknown discrete function u_h by a vector \mathbf{u} . More precisely, the unknowns $u_h(\mathbf{x})$ with $\mathbf{x} \in \Omega_h$ are collected in the vector \mathbf{u} and each matrix row represents the connections of one unknown to its neighbors governed by the discrete partial differential operator. The generalization to $(q \times q)$ systems of PDEs is straightforward. Here, we have q unknowns at each grid point \mathbf{x} .

TABLE 1.1: Operator, matrix, and stencil notation

continuous operator		discrete operator		matrix
L	\longrightarrow	$L_h \stackrel{\Delta}{=} [\ell_{\kappa}]_h$	\longrightarrow	\mathbf{L}
continuous function		discrete function		vector
u	\longrightarrow	u_h	\longrightarrow	\mathbf{u}

1.2 BASIC ITERATIVE SCHEMES

For the solution of linear systems of equations governed by large sparse matrices, iterative methods like Gauss-Seidel (GS), Jacobi (JAC), weighted Jacobi (ω -JAC), or successive overrelaxation (SOR) methods are a natural and popular choice. The Jacobi- or simultaneous-displacement method is of theoretical importance for explaining some principles.

Such *classical* iterative methods are constructed in the following way [63]. For solving a linear system of $\mathbf{L}\mathbf{u} = \mathbf{f}$ the matrix \mathbf{L} is split into $\mathbf{L} = \mathbf{C} - \mathbf{A}$, where \mathbf{C} is assumed to be regular.

From $\mathbf{L}\mathbf{u} = \mathbf{f}$ we get

$$(\mathbf{C} - \mathbf{A})\mathbf{u} = \mathbf{f} \quad \text{or}$$

$$\mathbf{C}\mathbf{u} = \mathbf{A}\mathbf{u} + \mathbf{f} \quad \text{and with iteration count}$$

$$\mathbf{u}^{(i+1)} = \mathbf{C}^{-1}\mathbf{A}\mathbf{u}^{(i)} + \mathbf{C}^{-1}\mathbf{f}.$$

The convergence of the iterative method is determined by the matrix $\mathbf{S} := \mathbf{C}^{-1}\mathbf{A}$, which is called the iteration matrix. For the description of a particular iteration scheme one simply has to specify the corresponding splitting of the matrix \mathbf{L} . More precisely, it is sufficient to specify \mathbf{C} since \mathbf{A} can be computed using $\mathbf{A} = \mathbf{C} - \mathbf{L}$. Obviously \mathbf{C} should be chosen in such a way that it is easily invertible.

Example 1.5 (Jacobi and Gauss-Seidel relaxation). In order to define some well-known classical iterative schemes it is convenient to split the matrix \mathbf{L} into $\mathbf{L} = \mathbf{D} - \mathbf{E} - \mathbf{F}$. \mathbf{D} is the diagonal part of \mathbf{L} , while \mathbf{E} and \mathbf{F} are the strictly lower and upper nondiagonal parts of \mathbf{L} with reverse sign, respectively. The Jacobi and the Gauss-Seidel relaxation are given by $\mathbf{C} = \mathbf{D}$ and $\mathbf{C} = \mathbf{D} - \mathbf{E}$, respectively. The corresponding iteration matrices read:

$$\mathbf{S}^{\text{JAC}} := \mathbf{D}^{-1}(\mathbf{E} + \mathbf{F}) \quad \text{and} \quad \mathbf{S}^{\text{GS}} := (\mathbf{D} - \mathbf{E})^{-1}\mathbf{F}. \quad (1.13)$$

There are modifications of these methods which do not add the complete increment to $\mathbf{u}^{(i)}$ to obtain $\mathbf{u}^{(i+1)}$ but only a weighted one. These are the weighted Jacobi and the weighted Gauss-Seidel method. They are given by the following two iteration matrices:

$$\begin{aligned} \mathbf{S}^{\text{JAC}}(\omega) &:= \omega\mathbf{D}^{-1}(\mathbf{E} + \mathbf{F}) + (1 - \omega)\mathbf{I} \quad \text{and} \\ \mathbf{S}^{\text{GS}}(\omega) &:= (\mathbf{D} - \omega\mathbf{E})^{-1}((1 - \omega)\mathbf{D} + \omega\mathbf{F}). \end{aligned}$$

Obviously $\mathbf{S}^{\text{JAC}}(\omega = 1) = \mathbf{S}^{\text{JAC}}$ and $\mathbf{S}^{\text{GS}}(\omega = 1) = \mathbf{S}^{\text{GS}}$ hold and the undamped methods are recovered. \blacktriangleleft

Note that the weighted Gauss-Seidel relaxation is usually called the successive overrelaxation (SOR) method. However in connection with multigrid methods—where it is used as a smoother and not as a solver, see below—it is often referred to as weighted Gauss-Seidel relaxation (ω -GS).

1.3 A FIRST DISCUSSION OF FOURIER COMPONENTS

When dealing with the solution of the discrete problem there are three quantities which have to be considered. The approximation to the solution (produced by any iterative method), the algebraic error, and the residual. The residual and especially the residual equation are of importance when deriving multigrid schemes.

If the exact solution u_h of the discrete problem solves

$$L_h u_h = f_h \quad \text{on } \Omega_h, \quad (1.14)$$

an approximation $u_h^{(i)}$ after i iterations corresponds to some algebraic error $e_h^{(i)} = u_h - u_h^{(i)}$ and to a residual (or defect), because the approximation will not satisfy (1.14) exactly: $r_h^{(i)} := f_h - L_h u_h^{(i)}$.

1.3.1 Empirical calculation of convergence factors

The quantities like $u_h^{(i)}$, $e_h^{(i)}$, and $r_h^{(i)}$ can be measured by standard vector norms like the infinity norm or the discrete Euclidean norm defined in Section 1.1.

The above-mentioned classical iterative methods show an unsatisfactory convergence, especially when the mesh sizes h_1, \dots, h_d are small. A convergence history both for the development of the error and the residual applying the Jacobi method four hundred times to the two-dimensional Poisson equation

$$-\Delta_h u_h = f_h \quad \text{on } (0, 1)^2, \quad u_h = g_h \quad \text{on } [0, 1]^2 \setminus (0, 1)^2 \quad (1.15)$$

with fixed f_h and g_h , discretized on a square grid with $h = h_1 = h_2 = \frac{1}{128}$ is given in Figure 1.1 (where f_h and g_h are chosen such that the exact solution is $u(x_1, x_2) = x_1^3 + x_2^2$). The convergence history of Figure 1.1 shows that the development of error and residual becomes very similar after a certain number of iterations. The empirical error-reduction $\|e_h^{(i)}\| / \|e_h^{(i-1)}\|$ and residual-reduction $\|r_h^{(i)}\| / \|r_h^{(i-1)}\|$ for varying mesh size h are given in Table 1.2 revealing an h -dependent behavior: The asymptotic convergence clearly deteriorates for $h \rightarrow 0$. This behavior can be explained for model

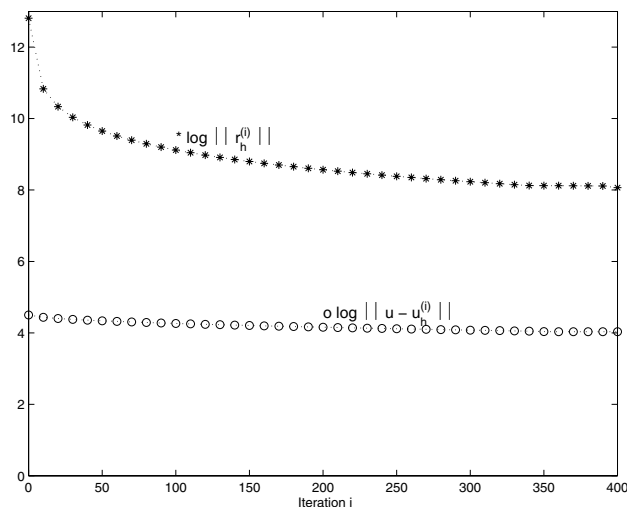


FIGURE 1.1: Convergence history for Jacobi relaxation, $\omega = 1$, $h = \frac{1}{128}$.

TABLE 1.2: Error- and defect-reduction after 1000 iterations, Poisson problem, Jacobi iteration

	error-reduction	defect-reduction
$h = \frac{1}{16}$	0.98079	0.98079
$h = \frac{1}{32}$	0.99518	0.99518
$h = \frac{1}{64}$	0.99877	0.99867
$h = \frac{1}{128}$	0.99959	0.99903

problems and basic relaxation schemes. *The proper understanding of this fact will help to motivate the functionality of multigrid methods and to introduce the basic principles of local Fourier analysis.*

1.3.2 Convergence analysis for the Jacobi method

The poor and h -dependent convergence of iterative schemes has been demonstrated by numerical experiments in the previous subsection. The theoretical explanation now follows.

The asymptotic convergence property of an iterative method is characterized by the spectral radius of its corresponding operator

$$\rho(S_h) = \max \{ |\lambda| \mid \lambda \text{ is eigenvalue of } S_h \},$$

which is also called the asymptotic convergence factor. The iterative process

converges if and only if $\rho(S_h)$ is smaller than one. It can be shown that this is equivalent to an asymptotic error or residual reduction smaller than one. We would like to emphasize that these *asymptotic* reduction factors are usually observed after only a few iterations for elliptic problems and multi-grid methods. As an alternative, one might consider norms of the iteration operator which yield upper bounds for the error reduction for *each* iteration step including the very first iterations. However, norm estimates are often too pessimistic for practical problems where the spectral radius is usually a sharper approximation for the actual convergence behavior.

The spectral radius of the iteration operator for the Jacobi method (1.13),

$$\rho(S_h^{\text{JAC}}) = \rho(D_h^{-1}(E_h + F_h)),$$

can be easily determined analytically for the discrete model problem from the previous subsection. The first step is to verify that

$$\begin{aligned}\varphi^{\ell,m}(\mathbf{x}) &= \sin(\ell\pi x_1) \sin(m\pi x_2) \quad (\mathbf{x} = (x_1, x_2) \in \Omega) \quad \text{and} \\ \lambda^{\ell,m} &= \ell^2 + m^2 \quad \text{for all } \ell, m = 1, 2, \dots\end{aligned}$$

are eigenfunctions, respectively eigenvalues of the two-dimensional Laplacian on the unit square with Dirichlet boundary conditions. The corresponding discrete eigenvalue problem

$$\begin{aligned}-\Delta_h u_h &= \lambda_h u_h \quad \text{in } \Omega_h = (0, 1)^2 \cap G_h \\ u_h &= 0 \quad \text{at } \Gamma_h\end{aligned}$$

on a square grid with $h = 1/n = h_1 = 1/n_1 = h_2 = 1/n_2$ has discrete eigenfunctions

$$\varphi_h^{\ell,m}(\mathbf{x}) = \sin(\ell\pi x_1) \sin(m\pi x_2) \quad (\mathbf{x} \in \overline{\Omega}_h) \quad (1.16)$$

with $1 \leq \ell, m \leq n - 1$. The respective discrete eigenvalues are given by

$$\lambda_h^{\ell,m} = \frac{1}{h^2} [4 - 2 \cos(\ell\pi h) - 2 \cos(m\pi h)] = \frac{4}{h^2} \left[\sin^2 \left(\frac{\ell\pi h}{2} \right) + \sin^2 \left(\frac{m\pi h}{2} \right) \right].$$

Note, that the discrete eigenfunctions and eigenvalues might be rewritten as

$$\begin{aligned}\varphi_h^{\text{D}}(\boldsymbol{\theta}, \mathbf{x}) &:= \sin(\theta_1 x_1 / h_1) \sin(\theta_2 x_2 / h_2) \quad (= \varphi_h^{\ell,m}(\mathbf{x})) \quad \text{and} \\ \lambda_h(\boldsymbol{\theta}) &:= \frac{4}{h^2} \left[\sin^2 \left(\frac{\theta_1}{2} \right) + \sin^2 \left(\frac{\theta_2}{2} \right) \right] \quad (= \lambda_h^{\ell,m}) \quad \text{with} \\ \boldsymbol{\theta} \in \Theta^{\text{D}} &:= \{ \boldsymbol{\theta} \mid \boldsymbol{\theta} = (\theta_1, \theta_2) \text{ with } \theta_1 = \pi\ell/n_1, \theta_2 = \pi m/n_2 \}.\end{aligned} \quad (1.17)$$

The superscript “D” refers to “Dirichlet” boundary conditions. $\varphi_h^{\text{D}}(\boldsymbol{\theta}, \mathbf{x})$ are called Fourier components, Fourier modes or simply modes [4]. $\boldsymbol{\theta}$ are known as Fourier frequencies; compare with the general description of local Fourier analysis in the second part of this monograph.

The special structure of the operators both for the discrete Poisson equation and for the Jacobi scheme allows for a direct analysis of the model problem. From

$$S_h^{\text{JAC}} = D_h^{-1} (E_h + F_h) = D_h^{-1} (D_h - L_h) = I_h - D_h^{-1} L_h$$

we calculate the eigenvalues of S_h^{JAC} as

$$\begin{aligned} \lambda_h^{\ell,m}(S_h^{\text{JAC}}) &= 1 - \frac{h^2}{4} \lambda_h^{\ell,m} = 1 - \left(1 - \frac{1}{2} \cos \ell \pi h - \frac{1}{2} \cos m \pi h \right) \\ &= \frac{1}{2} (\cos \ell \pi h + \cos m \pi h), \end{aligned}$$

and we obtain the spectral radius $\rho(S_h^{\text{JAC}}) = \cos \pi h$ for the Jacobi scheme applied to the Poisson equation. It is shown in Table 1.3 for different mesh sizes. The asymptotic convergence factors presented are in good accordance

TABLE 1.3: Asymptotic convergence factors $\rho(S_h^{\text{JAC}}) = \cos \pi h$ for the Jacobi relaxation applied to the Poisson equation

h	1/16	1/32	1/64	1/128
$\rho(S_h^{\text{JAC}})$	0.98079	0.99518	0.99880	0.99970

with the values of Table 1.2 after one thousand relaxations.

The *weighted* Jacobi scheme also allows for a direct analysis of our discrete model problem. Again, the eigenvalues $\lambda_h^{\ell,m}(\omega)$ of the iteration operator are determined from the representation of $S_h^{\text{JAC}}(\omega)$, using the fact that the eigenfunctions given by (1.16) of the operators $-\Delta_h$ and $S_h^{\text{JAC}}(\omega)$ coincide. This yields

$$\lambda_h^{\ell,m}(\omega) = 1 - \omega \frac{h^2}{4} \lambda_h^{\ell,m} = 1 - \omega \left(\sin^2 \frac{\ell \pi h}{2} + \sin^2 \frac{m \pi h}{2} \right) \quad (1.18)$$

for all $1 \leq \ell, m \leq n-1$. $\omega \in (0, 1]$ implies convergence. However, the spectral radius is very close to one if h is small and there is no satisfactory convergence for any ω . We can easily spot the “ h -dependent convergence”: the smaller h becomes the worse the convergence is.

1.3.3 Smoothing properties of Jacobi relaxation

The eigenfunctions $\varphi_h^{\ell,m}$ of the iteration operator form a basis for all grid functions having zero boundary values. Therefore, an initial error $e_h^{(0)}$ for the

Poisson equation with Dirichlet boundary conditions is given by

$$e_h^{(0)} = \sum_{\ell,m=1}^{n-1} c_{\ell,m}^{(0)} \varphi_h^{\ell,m}$$

with some coefficients $c_{\ell,m}^{(0)}$. (Note that the error or the residual are generally zero at a Dirichlet boundary.) For the weighted Jacobi scheme, the error $e_h^{(i)}$ after i iterations is $e_h^{(i)} = (S_h^{\text{JAC}}(\omega))^i (e_h^{(0)})$. Inserting the representation of $e_h^{(0)}$ gives

$$e_h^{(i)} = \sum_{\ell,m=1}^{n-1} c_{\ell,m}^{(0)} \left(S_h^{\text{JAC}}(\omega) \right)^i \varphi_h^{\ell,m} = \sum_{\ell,m=1}^{n-1} c_{\ell,m}^{(0)} \left(\lambda_h^{\ell,m}(\omega) \right)^i \varphi_h^{\ell,m}. \quad (1.19)$$

Again, the identity of eigenfunctions of both $S_h^{\text{JAC}}(\omega)$ and $-\Delta_h$ has been used. The representation of $e_h^{(i)}$ is governed by coefficients

$$c_{\ell,m}^{(i)} := c_{\ell,m}^{(0)} \left(\lambda_h^{\ell,m}(\omega) \right)^i.$$

That is, the coefficients of the initial error are damped by the i -th power of the corresponding eigenvalue.

Selected frequencies are used as initial guesses to illustrate the behavior of the Jacobi iteration. The modes with $(\ell, m) \in \{(1, 1); (3, 3); (6, 6)\}$ on a grid with $h = \frac{1}{128}$ are reduced faster when the wave numbers are higher, see [Figure 1.2](#). In general, an initial guess will be composed of several modes with different wave numbers. The previous experiment, but now with a linear combination of $\varphi_h^{1,1}, \varphi_h^{6,6}$ and $\varphi_h^{64,64}$, shows a well-known result. After an initially fast decrease the reduction becomes worse because the contributions to the error referring to modes with high wave numbers are already smoothed out and the iteration scheme slowly reduces the now dominating components with small wave numbers; see [Figure 1.3](#). Many basic relaxation methods behave like this. Regarding this matter it seems reasonable to distinguish between modes with small and high wave numbers. Hence, the above Fourier modes are subdivided into two disjoint subsets. Modes with wave numbers $1 \leq \ell, m < \frac{n}{2}$ are called low-frequency modes. Those with $\max(\ell, m) \geq \frac{n}{2}$ are high-frequency ones. (The reason for this particular splitting becomes more obvious below. It is related to the “coarsening strategy” within a multi-grid method; compare with Sections 3.1 and especially 5.2.) For the related Fourier frequencies $\boldsymbol{\theta} = (\pi\ell/n, \pi m/n)$ this means that low frequencies are characterized by $\boldsymbol{\theta} \in (0, \pi/2)^2$ and high frequencies by $\boldsymbol{\theta} \in (0, \pi)^2 \setminus (0, \pi/2)^2$. A wave number ℓ represents $\frac{\ell}{2}$ full sine waves on the unit interval, each having a wavelength of $\frac{2}{\ell}$. The wavelength of a low-frequency mode ranges from $4h$ to 2 and the wavelength of high-frequency ones varies between $2h$ and $4h$.

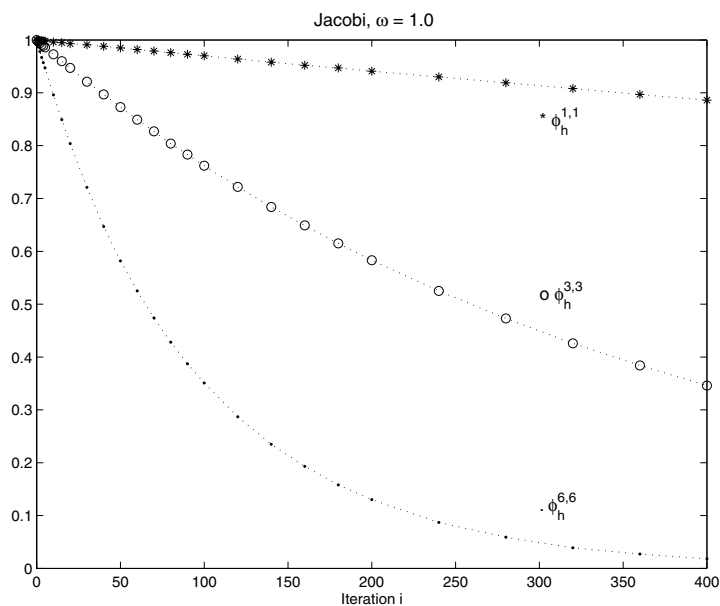


FIGURE 1.2: Damping of $\varphi_h^{1,1}$, $\varphi_h^{3,3}$, and $\varphi_h^{6,6}$.

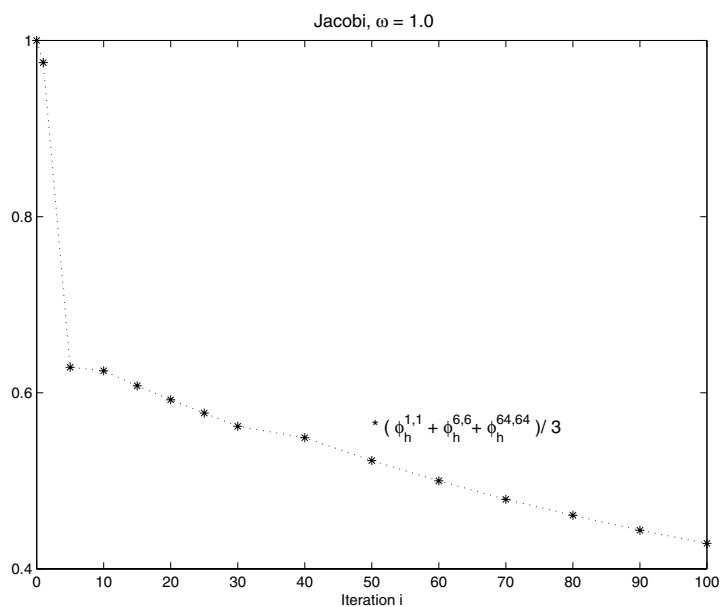


FIGURE 1.3: Damping of $\frac{1}{3} (\varphi_h^{1,1} + \varphi_h^{6,6} + \varphi_h^{64,64})$.

Modes with $\ell, m \geq n$ cannot be represented on an h -grid, because their wave lengths are smaller than $2h$. They coincide with modes with wave numbers $0 < m, \ell < n$ due to the periodicity of the sine function. For instance, modes with $\ell = 13$ alias with modes related to wave number $2n - \ell = 3$, considering the mesh size $h = 1/n = 1/8$.

A reasonable question now might be how a certain relaxation method will work on high-frequency modes, i.e., how good the *smoothing* is. This is combined with the question of whether this property can be quantified. It turns out that there is a number, the “smoothing factor,” which measures the smoothing property of relaxation schemes. The smoothing factor ρ_1 is defined to be the worst damping factor of all high-frequency modes. For instance, weighted Jacobi relaxation with $\omega = 0.8$ on a square grid has a smoothing factor $\rho_1(\omega) = 0.6$ assuming the above distinction between high- and low-frequency components. This means that high-frequency modes are reduced per relaxation step at least by a factor of 0.6 which might be easily verified using formula (1.18). A motivation for the particular choice of the relaxation parameter ($\omega = 0.8$)—which is in fact optimal for this particular problem—is given in Example 5.17. For Gauss-Seidel relaxation with a lexicographical ordering of grid points one obtains $\rho_1 = 0.5$ and with a checkerboard ordering even $\rho_1 = 0.25$. Details and a thorough description of this kind of “smoothing analysis” are given in [Chapter 5](#).

1.4 FROM RESIDUAL CORRECTION TO COARSE-GRID CORRECTION

Solving the discrete problem

$$L_h u_h = f_h \tag{1.20}$$

iteratively yields a sequence of approximations $u_h^{(i)}$ ($i = 0, 1, 2, \dots$) of the discrete solution u_h . The algebraic error $e_h^{(i)} = u_h - u_h^{(i)}$ measures the actual quality of the approximation. Due to the definition of the residual after i iterations, $r_h^{(i)} := f_h - L_h u_h^{(i)}$, it is easy to verify that we have

$$r_h^{(i)} = f_h - L_h u_h^{(i)} = f_h - L_h(u_h - e_h^{(i)}) = f_h - L_h u_h + L_h e_h^{(i)}.$$

Together with (1.20) this gives the defect equation

$$L_h e_h^{(i)} = r_h^{(i)}. \tag{1.21}$$

From this relationship it turns out that solving (1.21) immediately yields the desired solution by $u_h = u_h^{(i)} + e_h^{(i)}$. Hence, an iterative process for solving (1.21) can be considered as an iterative method to solve (1.20).

An approximate solution of the defect equation $L_h e_h^{(i)} = r_h^{(i)}$ can be obtained by solving a “similar problem”

$$\widehat{L}_h \widehat{e}_h^{(i)} = r_h^{(i)} \quad (1.22)$$

with an operator \widehat{L}_h being “similar” to L_h . If the inverse operator \widehat{L}_h^{-1} exists, we use the solution

$$\widehat{e}_h^{(i)} = \widehat{L}_h^{-1} r_h^{(i)}$$

of the above equation to compute a new approximation

$$\begin{aligned} u_h^{(i+1)} &= u_h^{(i)} + \widehat{e}_h^{(i)} = u_h^{(i)} + \widehat{L}_h^{-1} r_h^{(i)} = u_h^{(i)} + \widehat{L}_h^{-1} f_h - \widehat{L}_h^{-1} L_h u_h^{(i)} \\ &= (I_h - \widehat{L}_h^{-1} L_h) u_h^{(i)} + \widehat{L}_h^{-1} f_h, \end{aligned} \quad (1.23)$$

with an iteration operator $M_h := I_h - \widehat{L}_h^{-1} L_h$ where I_h represents the identity operator. Because the solution of the residual equation has been used to “correct” the approximation, this procedure is called residual correction.

Now, the crucial point is to find a problem similar to the defect equation which is cheaper to solve in order to benefit from the above rewriting of the original equation (1.20). Regarding this matter, a coarser grid Ω_H ($H > h$) and an appropriate approximation L_H to L_h on Ω_H are introduced to formulate an H -defect equation $L_H e_H^{(i)} = r_H^{(i)}$ which can be solved fast for a small number of grid points in Ω_H . Applying a restriction operator R_H^H we define $r_H^{(i)} := R_H^H r_h^{(i)}$. Then, the solution of the coarse-grid equation $L_H e_H^{(i)} = r_H^{(i)}$ is transferred back to the fine grid by an interpolation operator (prolongation) P_H^h . This interpolated solution of the coarse-grid defect equation is an approximation to that of the fine-grid defect equation. It is used to correct the fine-grid approximation. This approach makes sense if the coarse-grid quantities $e_H^{(i)}$ and $r_H^{(i)}$ are reasonable approximations to the corresponding fine-grid quantities. Especially for the error we know from the previous section that after some relaxation steps the high-frequency components have been smoothed out and low-frequency components dominate. Such smooth quantities can be approximated well on coarser meshes.

1.5 MULTIGRID PRINCIPLE AND COMPONENTS

The development of multigrid methods for the numerical solution of an elliptic boundary value problem

$$L_h u_h = f_h \text{ on } \Omega_h, \quad B_h u_h = g_h \text{ at } \partial\Omega_h$$

is motivated by two basic observations which have been discussed in Section 1.3.3 and Section 1.4, respectively:

1. Smoothing principle: Many classical iterative methods (like Jacobi, Gauss-Seidel relaxation, etc.) have a strong error smoothing effect if they are applied to discrete, elliptic problems.
2. Coarse-grid correction principle: A smooth error term, on the other hand, can be represented well on a coarser grid Ω_H where its approximation is substantially less expensive.

If the grid coarsening is performed by doubling the mesh size into each direction, i.e., $\Omega_H = \Omega_{2h}$, we speak of standard coarsening. This is particularly straightforward if n_1, \dots, n_d from (1.3) are even numbers. The above principles suggest the following structure of a two-grid cycle for a linear problem called the correction scheme (CS), compare with Algorithm 3.1:

- Presmoothing: Perform ν_1 steps of a classical iterative relaxation method S_h on the fine grid Ω_h .
- Coarse-grid correction:
 - Compute the defect (or the residual) of the current fine-grid approximation.
 - Restrict the defect to the coarse grid Ω_H using the fine-to-coarse transfer operator R_h^H .
 - Solve the coarse-grid defect equation on Ω_H .
 - Interpolate the correction using the coarse-to-fine transfer operator P_H^h .
 - Add the interpolated correction to the current fine-grid approximation.
- Postsmoothing: Perform ν_1 steps of a classical iterative relaxation method S_h on the fine grid Ω_h .

The *multigrid* idea is based on the observation that it is not necessary to solve the coarse-grid defect equation exactly. Instead, one could approximate its solution by γ two-grid cycles with zero initial approximation. A recursive application of this idea yields a multigrid or k -grid method involving k different grids. Summarizing, for the unique definition of a multigrid cycle one has to specify its components: The number k of involved grids, the smoothing procedures on each grid, the number of pre- and postrelaxation steps on each grid, the coarsening strategy which determines the coarser grids, the discretization operators on each grid, the transfer operators, and the cycle index γ . Note that at least a three-grid analysis is necessary to analyze the different behaviour of varying cycle indices or pre- and postsmoothing. Examples are given in [Chapter 4](#).

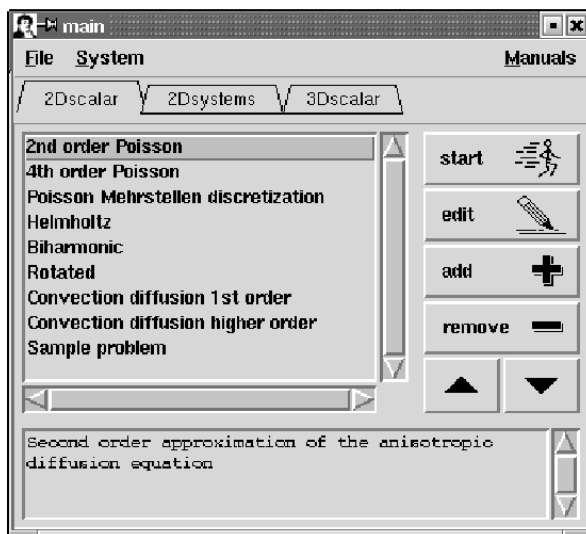


FIGURE 1.4: The initial window of *xlfa*.

1.6 A FIRST LOOK AT THE GRAPHICAL USER INTERFACE

Having learned how multigrid is composed of different components which have to be put together in a proper way, the local Fourier analysis software **LFA** should be able to analyze these pieces of the puzzle. Therefore, the graphical user interface (GUI) *xlfa* offers an easy way to select the multigrid components. This will shortly be shown here for a model problem.

When starting the GUI main function (`wish main`) the collection of pre-defined applications is offered to select one. Here we choose our model problem from the class of Poisson-type equations with standard second-order discretization discussed in Section 1.3.

Pressing the **start**-button in Figure 1.4 makes the main window (Figure 1.5) of *xlfa* appear. This window supports the selection of all previously mentioned components. Pre- and postsmoothing can be chosen in the same way as the numbers ν_1 and ν_2 of pre-, and postsmoothing steps, respectively (left and right boxes in the part which we denote as **Parameter display** in Figure 1.5, see also Figure 3.3). When clicking the bars with **Presmoothing** or **Postsmoothing** a window appears where the number of smoothing steps can be modified from default values to desired ones by moving sliders. Pressing the bar **Presmoothing** opens a selection box which offers all the possible smoothing methods. We select red-black Gauss-Seidel for one pre- and one postsmoothing step as shown in Figure 1.6.

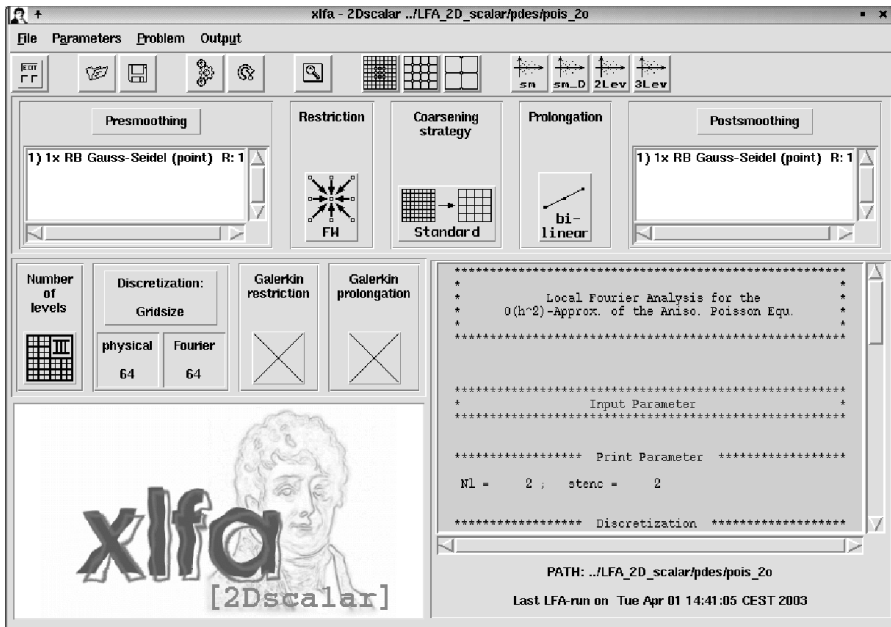


FIGURE 1.5: The main window of xlfA.



FIGURE 1.6: Relaxation methods for pre- and postsmoothing.

The coarse-grid correction step is composed of transfer operators both from fine to coarse and vice versa and the solution of the coarse-grid equation. Restriction and interpolation (prolongation) depend on the coarsening strategy. Pressing the button of any of these parameters opens a more detailed window (see Figure 1.7), where restriction, coarsening strategy, and prolongation can be selected. We choose full weighting and bilinear interpolation for the transfer of defects and corrections, respectively (compare with Sections 3.4.5 and 3.4.6). This is a proper choice as these methods are adjoint operators up to a scaling factor. The cycle parameter γ determines the complexity of the cycle and to some extent controls the amount of work to be invested on coarser meshes. The default value is $\gamma = 1$, representing the V-cycle. Standard coarsening and natural analogs of the fine-grid discretization on coarser grids are used within the V-cycle.

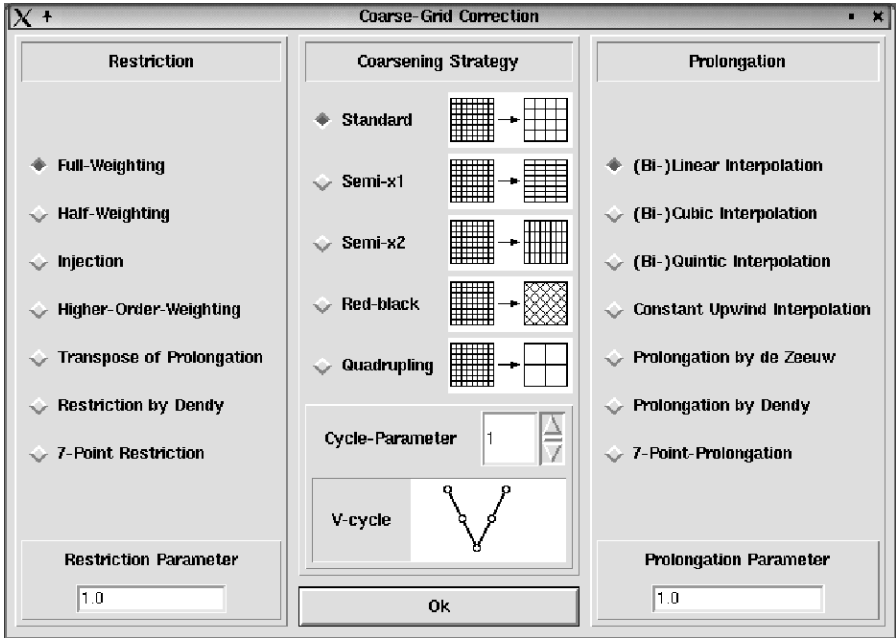


FIGURE 1.7: Possible parameters for coarse-grid correction.

Below the `Parameter display` block for the coarse-grid correction it is possible to select the type of analysis to be performed. The single-grid analysis essentially delivers the smoothing factor of the chosen relaxation. If the number of levels has been fixed to two, a two-grid analysis for the scheme is performed. With a three-level analysis we can expect the results to be more

realistic and closer to the final multigrid performance. When all the parameters have been selected, **LFA** may be started either from the **Problem**-menu and the **run program** selection or by pressing the **run**-button in the button bar (top area in the GUI). Running the program delivers the smoothing factor and two- or three-grid convergence predictions, respectively. Different norms of the operators are given as well (see Figure 1.8).

Due to the chosen multigrid components (V(1,1), standard-coarsening, full-weighting, bilinear interpolation, red-black Gauss-Seidel, $\nu = \nu_1 + \nu_2 = 1 + 1 = 2$), a multigrid program applied to the Poisson equation with Dirichlet boundary conditions (solution $u(x_1, x_2) = x_1^3 + x_2^2$) delivers empirical defect-reduction rates of about 0.10 for the cycles 5 to 14. The corresponding empirical error-reduction rate of 0.11 is only slightly worse. To determine the asymptotic behavior of an algorithm it is a standard approach to specify zero boundary conditions and zero right-hand side, thus looking for the zero function as solution. Then, from a nonzero initial guess many cycles can be evaluated and the empirical reduction factors should reach the asymptotic behavior. Performing such an experiment with the above components gives 0.1193 both for error reduction and for residual reduction. This value is obtained when using a finest mesh size both with 64 subdivisions of the unit interval and with 128 subdivisions performing more than one hundred cycles.

In Figure 1.8, upper left window, the smoothing factor $\rho(S^\nu Q)$ for the selected relaxation method (Gauss-Seidel red-black) is shown. The given value of 0.0625 for two smoothing steps is nothing else than the information that the chosen component has good smoothing properties. It is not necessarily a good approximation of the multigrid convergence, because no grid-transfer component is considered for its calculation. The upper-right window in Figure 1.8 additionally shows values for the two-grid operator. There is the spectral radius $\rho(M_{2L})$ giving asymptotic information, and the norms of the two-grid method both for error reduction $\|M_{2L}\|_s$ and for residual (defect) reduction $\|M_{2L}\|_d$. The asymptotic estimation of a *two*-grid method is too optimistic compared to the observed behavior of a *multigrid* program. The norm values shown give some information about the improvement which can be reached at least per iteration step. From this it is not surprising that the estimation of *what can be reached at least* for each iteration step (including the very first ones) is worse than that which has been reached asymptotically. As the two-level method is still an artificial limitation when thinking in terms of many levels for multigrid, the three-grid values may be a better prediction of possible multigrid convergence. In the lower-left window of Figure 1.8 the value $\rho(M_{3L})$ now comes close to the empirically observed asymptotic convergence of 0.1193 (independent on the mesh size, using six and seven levels, respectively). In a similar way as mentioned before, the norms $\|M_{3L}\|_s$ and $\|M_{3L}\|_d$ indicate that the improvement per step should be better than 0.15. This has been verified in practice considering the complete convergence history.

If desired, the distribution of the eigenvalues of the analyzed operators can be displayed, too. We show results for the two-level method using Gauss-

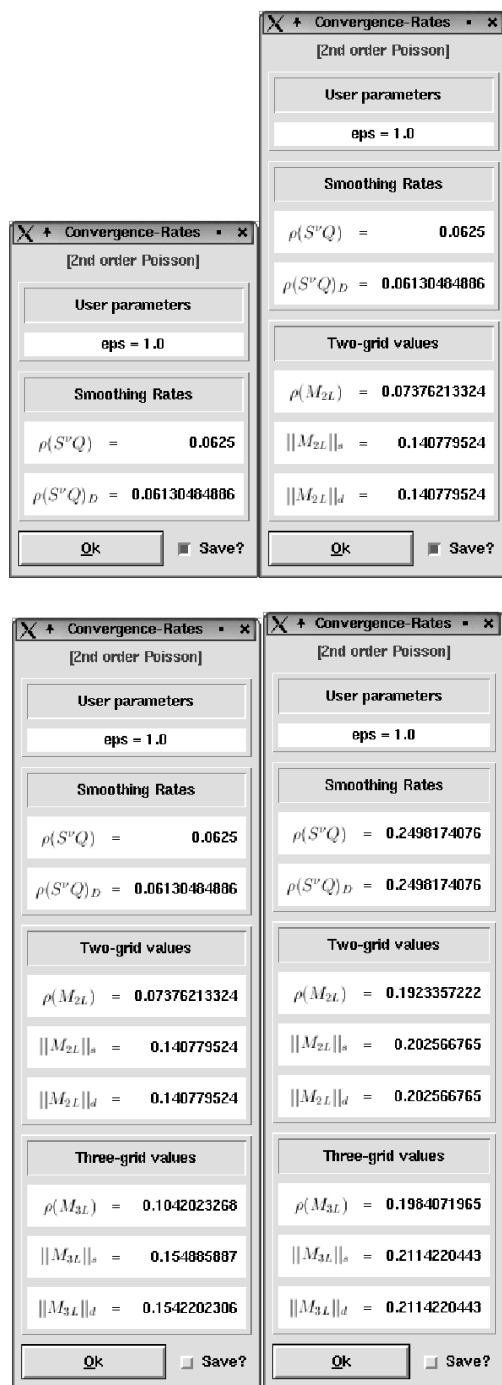


FIGURE 1.8: Smoothing factors, two-, and three-grid convergence estimates.

Seidel with red-black ordering (left picture in Figure 1.9) and additionally the eigenvalues of the two-level method when selecting lexicographic ordering for the Gauss-Seidel smoothing (right picture in Figure 1.9). The plot of such distributions is selected again from the button bar (right part). For comparison, the three-grid values for this selection of ordering are shown in the lower-right window of Figure 1.8. The experiment for the asymptotic behavior delivers values of about 0.184, being close to the predicted value. Both practice and theoretical prediction show the influence of ordering on the final convergence behavior.

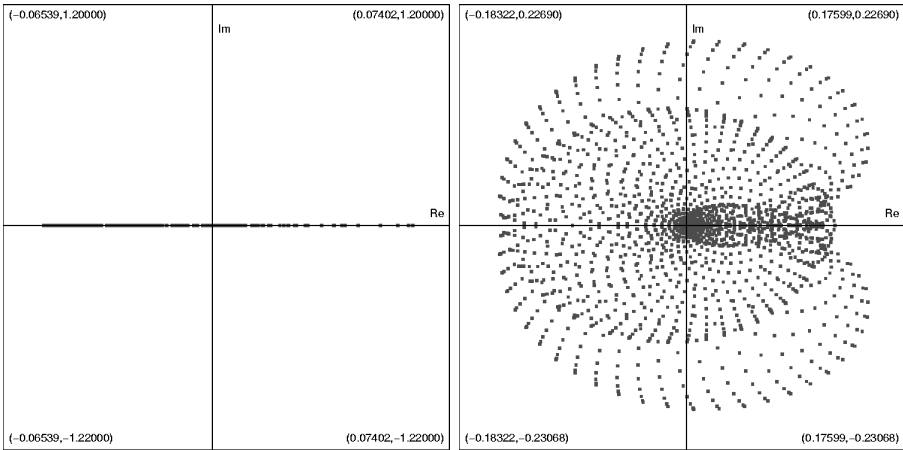


FIGURE 1.9: Distribution of eigenvalues for three-grid methods applying Gauss-Seidel relaxation with different orderings; red-black (left), lexicographic (right).

Another selection made from the button bar concerns the stencils which are used on the different levels. It is possible to visualize the stencils at all levels involved in the selected Fourier analysis. An important quantity in the framework of local Fourier analysis is the measure of h -ellipticity of a discrete operator L_h . A certain amount of h -ellipticity is a sufficient condition for the existence of an efficient point smoother. More details are given in Section 5.8. Figure 1.10 shows the discrete Laplacian and the corresponding measure of h -ellipticity on three different levels. Because we use standard coarsening and coarse-grid analogs of the fine-grid discretization, the five-point stencils look very similar—up to the factor 4 because of doubling the mesh size. Below we will show what happens when nonstandard coarsening strategies or coarse-grid operators have been chosen.

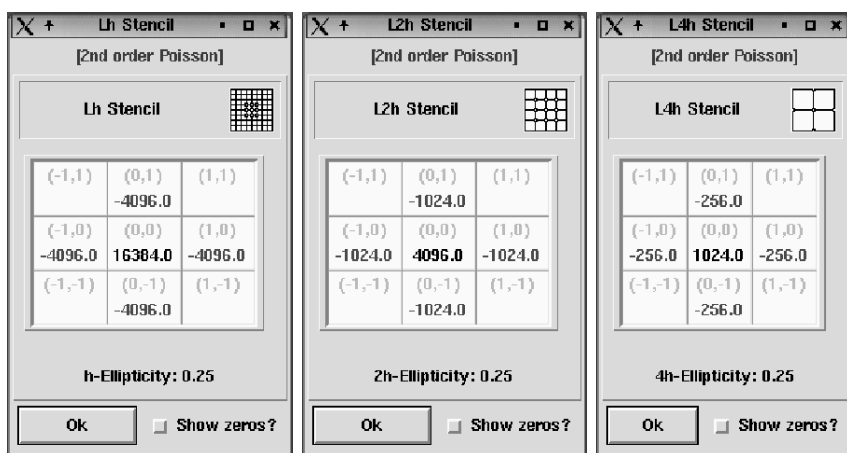


FIGURE 1.10: Stencil representation for the central approximation of the Poisson equation on three different levels.

Chapter 2

MAIN FEATURES OF LOCAL FOURIER ANALYSIS FOR MULTIGRID

In this chapter, we summarize the basic features of local Fourier analysis for multigrid methods without going into technical details. It is intended to give a short overview about the main ideas, the power, and the range of applicability of local Fourier analysis. In this respect it should be particularly worthwhile for the practitioner who is mainly interested in the accompanying software. Here he gets a qualitative insight into the principles of local Fourier analysis without wading through the theoretical chapters from the second part of this monograph. Moreover, for the theoretically oriented researcher it may serve as a preparation and guideline for the (more technical) description of local Fourier analysis for multigrid presented in the second part.

The application of Fourier analysis to multigrid methods has been introduced by Achi Brandt [4] who prefers the expression local *mode* analysis instead of local *Fourier* analysis. Comprehensive surveys including further contributions are given in [7, 59, 60, 62, 65]. Our presentation is related to the notation and philosophy from [59, 60, 62]. The recent developments concerning the k -grid analysis are based on [67, 68].

2.1 THE POWER OF LOCAL FOURIER ANALYSIS

A standard choice for a multigrid method is to apply a classical pointwise relaxation method as a smoother, to repeatedly double the mesh size in each spatial direction for the determination of coarser grids, to use straightforward geometric transfer operators from fine to coarse grids and vice versa, and to apply coarse-grid operators that are natural analogs of the fine-grid discretization. For “nicely” elliptic PDEs like the Poisson equation, these standard multigrid methods have been proven to be highly efficient using local Fourier analysis techniques. However, the basic multigrid structure described above is often not appropriate for more complex problems. The crucial point for the efficiency of any multigrid method is the correct choice of its components for

an efficient interplay between smoothing and coarse-grid correction.

The difficulties in connection with standard multigrid are mainly divided into two categories. The first one being that the error cannot be *uniformly* smoothed by the relaxation method. The second class of difficulties is due to the fact that smooth error components cannot be sufficiently reduced by the coarse-grid correction. For example, for anisotropic problems pointwise relaxation exhibits good smoothing properties only “into the directions of strong couplings” as it is shown in Sections 4.1.1 and 4.2.1. In the case of nonelliptic or singularly perturbed operators, there are smooth “characteristic” error terms that are poorly approximated on coarse grids leading to a severe deterioration of the multigrid convergence for standard components; compare with Section 7.2 and [2, 6, 12, 75]. Consequently, one has to switch to more sophisticated smoothers or one has to switch to nonstandard coarse-grid correction schemes with a satisfactory approximation of smooth error terms. Examples are given in Sections 4.1.6, 4.1.7, and 4.3.3.

Over the last three decades it has been demonstrated that carefully tuned multigrid methods could be successfully applied to a much larger class of problems than the nicely elliptic ones. During this development, the standard multigrid components were replaced by block relaxations, ILU smoothers, operator-dependent transfer operators, Galerkin coarse-grid operators, (multiple) semicoarsening techniques, etc. (compare with Sections 3.4 and 3.5) in order to overcome the particular difficulty at hand; see, for example, [62] for a recent monograph. In many cases this tuning was motivated and optimized with the help of local Fourier analysis. *In this respect, local Fourier analysis can be regarded as the main quantitative analysis tool for the development of new efficient multigrid methods.*

2.2 BASIC IDEAS

We start with a discrete linear boundary value problem

$$L_h u_h = f_h \text{ on } \Omega_h, \quad B_h u_h = g_h \text{ at } \partial\Omega_h$$

which is iteratively solved by a k -grid method.

2.2.1 Main goal

The main goal is to *estimate* the spectral radius or certain norms of the k -grid operator which are quantitative measures for the error (or defect) reduction, see Section 1.3.

For an *exact* calculation of these quantities, it is necessary that there exists a unitary basis of periodic eigenfunctions of L_h which generate the whole space

of grid functions and which are compatible with the boundary conditions. Then it is possible to expand the error (or the residual) after the i -th k -grid cycle into a Fourier series by a unitary basis transformation, and different multigrid methods can be analyzed by evaluating their effect on the eigenfunctions. An example is given in Section 1.3 for the Laplacian on the unit square with homogeneous Dirichlet boundary conditions. However, for more complicated domains, operators L_h , or (combinations of) boundary conditions an appropriate basis of unitary eigenfunctions does usually not exist.

2.2.2 Necessary simplifications for the discrete problem

Therefore, we assume that L_h has only constant coefficients, neglect boundary conditions and extend the discrete operator to an infinite grid. The eigenfunctions of such a constant coefficient infinite-grid operator are given by simple exponential functions—called the *Fourier components*—generating the whole space of bounded infinite-grid functions [9, 29, 65]. Thus, each infinite-grid function can be written as a linear combination of Fourier components. In particular an error or residual function can be expanded into a series with respect to the Fourier components.

So far, we described the necessary simplifications only with respect to the discrete operator L_h and the underlying domain. In order to apply the local Fourier analysis to the whole k -grid operator, we furthermore assume that all operators involved in the k -grid method have constant coefficients (which will usually hold if already L_h is assumed to have constant coefficients) and that they are extended to infinite grids (G_h , G_H , etc.) as well.

2.2.3 Crucial observation

Due to the above simplifications, it turns out that the complete k -grid error (or defect) reduction operator for large classes of multigrid components (to be specified in Chapters 5, 6 for the two- and three-dimensional cases) leaves certain *low-dimensional* subspaces of Fourier components invariant. (More precisely, these subspaces are low-dimensional for small k , for example, $k = 2, 3$.) This fact can be considered as *the crucial observation from the local Fourier analysis* and means that the k -grid operator is unitarily equivalent to a block-diagonal matrix with small blocks. From this block-diagonal representation it is easy to obtain the corresponding spectral radius or norm value which is usually done by a computer program, compare with Chapter 4 where the usage of the accompanying software to perform these computations is described for many test cases.

2.2.4 Arising questions

From the previous discussion it is obvious that we have to clarify two main questions:

1. What are the implications of the above simplifications for “my real-world application” which is nonlinear, has nonconstant coefficients and a complex domain? In other words, we have to specify the range of applicability of local Fourier analysis.
2. How do these block-diagonal representations (the so-called *Fourier representations*) of the k -grid operator look like in practice?

The explicit Fourier representations for practical k -grid methods ($k = 2, 3$) are derived in [Chapters 5](#) and [6](#). Since the Fourier representations are calculated automatically by the accompanying software, these chapters are mainly interesting for the theoretician. The answer of the practically very important first question is given in the next section.

2.3 APPLICABILITY OF THE ANALYSIS

At first sight, the necessary simplifications in the framework of local Fourier analysis, i.e.,

- L_h is a linear operator with constant coefficients,
- boundary conditions are neglected,

seem to be very restrictive and very unrealistic w.r.t. real applications. To resolve this objection we would like to emphasize the following crucial observation.

Under general assumptions, any general discrete operator, nonlinear with nonconstant coefficients can be linearized locally and can be replaced locally (by freezing the coefficients) by an operator with constant coefficients.

This means if we deal with linear operators $L_h(\mathbf{x})$ that are characterized by variable coefficients, we are able to apply the analysis to the locally frozen operator at a fixed grid point ξ . Replacing the variable \mathbf{x} by a constant ξ , one obtains an operator $L_h(\xi)$ with *constant frozen* coefficients. In [8, 57] it is shown that the convergence factor for an elliptic operator with varying coefficients can be bounded by the supremum over the estimates from local Fourier analysis for the locally frozen operators in case of smoothly varying coefficients. Moreover, practical experience demonstrates that estimates for linearized versions of the original moderately nonlinear problem are in good accordance with the actual performance. In this way it is possible to treat a very large class of discrete difference operators—including real-life applications—by local Fourier analysis. This will be validated by a large variety of different examples given in [Chapter 4](#). For example, in [Section 4.3.3](#)

the validity of local Fourier analysis for an application governed by the incompressible Navier Stokes equations is demonstrated, representing a nonlinear system of PDEs with nonconstant coefficients.

The neglect of boundary conditions does usually not affect the validity of local Fourier analysis as well, assuming a “correct” treatment of the boundaries. This important observation is proved in [8, 57]. This means that the predicted convergence factors *should always be obtained in the numerical experiments* through the expense of some negligible work (at least in the limit of small mesh size) at and near the boundaries. However, this “correct” treatment necessitates a certain expertise for difficult applications and complicated domains. Often it can be accomplished by some additional boundary relaxations.

2.3.1 Type of partial differential equation

Local Fourier analysis is rigorously proven to be valid for elliptic problems [8, 57]. However, for certain nonelliptic and singularly perturbed problems it can be very useful as well (see, for example [62, 67, 75] and the references therein) which will be demonstrated in the following by several examples like anisotropic diffusion or convection-dominated flow problems. Especially for hyperbolic equations, there are certain variants of the local Fourier analysis like the first differential approximation or the half-space analysis; compare with [7, 21]. Moreover, the (time) discretization of parabolic equations often leads to a sequence of elliptic problems which may be solved by multigrid methods and analyzed with the help of local Fourier analysis.

2.3.2 Type of grid

Local Fourier analysis is restricted to uniform grids but it is valid for general domains. (More precisely it is valid for general domains assuming a correct treatment of the boundary, see above.) In this book, we focus on vertex-centered grids G_h , i.e., the coarse grids are constructed by deleting grid points so that the nodes on every coarse grid form a subset of the nodes on finer grids. A popular alternative approach is to divide the computational domain into cells and to locate the unknowns at cell centers. This approach is most natural in connection with finite volume discretizations. Coarse grids are formed by merging fine-grid cells. Thus, the cell centers of a coarse grid do not belong to the next finer grid. Local Fourier analysis as presented in [Chapter 6](#) for the vertex-centered case can be straightforwardly adapted to the cell-centered case [44] which is briefly discussed in Section 7.3. The accompanying software is developed for multigrid methods on vertex-centered grids. However, certain parts of the analysis (in particular smoothing analysis or the investigation of the h -ellipticity to be defined below) do not distinguish between vertex- or cell-centered grids and can be applied to both cases.

For systems of PDEs arising from computational fluid dynamics so-called staggered grids are often applied. Here certain unknowns are placed at cell

centers whereas others are placed at their borders. Even for such complicated grid structures it is possible to perform local Fourier analysis; see [10, 46, 70].

2.3.3 Type of discretization

In principle, finite-difference (FD), finite-volume (FV), and finite-element (FEM) discretizations can be used in the context of local Fourier analysis as long as they lead to difference stencils with constant coefficients on uniform grids. However for FEM, the applicability is rather limited since there is no application to irregular grids which often arise from FEM discretizations. Thus, we will typically deal with FD or FV discretizations.

Summarizing, we have discussed the huge range of applicability and its practical relevance for the development of efficient multigrid methods for a large variety of problems. These observations will be illustrated and come to life in [Chapter 4](#) by considering a lot of case studies.

Chapter 3

MULTIGRID AND ITS COMPONENTS IN LFA

In this chapter we give a short introduction into multigrid methods for two- and three-dimensional problems including systems of equations. For a detailed description we refer to the literature, e.g., [7, 28, 62, 65]. Here, our main focus is to describe the multigrid components which can be selected in the accompanying software **LFA**. Sections 3.1 and 3.2 are for those readers who are not yet familiar with multigrid principles. In Sections 3.3, 3.4, and 3.5, the implemented multigrid components of **LFA** are detailed and it is explained how to select them via the graphical user interface **xlfa**.

3.1 MULTIGRID CYCLING

In the first chapter a special property of basic relaxation schemes has been worked out theoretically and practically: the fast damping of high-frequency Fourier modes. Such a behavior is typical for many iterative schemes. It is important to note that this fast reduction of high-frequency modes usually does not depend on the mesh size—in contrast to convergence. This property is a fundamental one for the construction of multigrid algorithms. A second basic idea has already been mentioned, the coarse-grid correction. The combination of both ideas leads to the multigrid correction scheme.

3.1.1 Coarse-grid correction operator

Recall from Section 1.4 that the coarse-grid correction is based on the solution of a coarse-grid residual equation

$$e_H^{(i)} = L_H^{-1} r_H^{(i)} = L_H^{-1} R_h^H r_h^{(i)}$$

with some restriction operator $R_h^H : \Omega_h \rightarrow \Omega_H$. Then, the coarse-grid quantity $e_H^{(i)}$ is transferred back to the fine grid by an interpolation or prolongation operator $P_H^h : \Omega_H \rightarrow \Omega_h$. This interpolated quantity $\hat{e}_h^{(i)} := P_H^h e_H^{(i)} =$

$P_H^h L_H^{-1} R_h^H r_h^{(i)}$ is applied to correct the current fine grid approximation, i.e.,

$$\begin{aligned}\hat{u}_h^{(i)} &= u_h^{(i)} + \hat{e}_h^{(i)} = u_h^{(i)} + P_H^h L_H^{-1} R_h^H r_h^{(i)} \\ &= u_h^{(i)} + P_H^h L_H^{-1} R_h^H f_h - P_H^h L_H^{-1} R_h^H L_h u_h^{(i)} \\ &= K_h^H u_h^{(i)} + P_H^h L_H^{-1} R_h^H f_h\end{aligned}$$

with the coarse-grid correction operator

$$K_h^H = I_h - P_H^h L_H^{-1} R_h^H L_h. \quad (3.1)$$

Note that the coarse-grid correction procedure does not converge. There exist h -grid functions $r_h^{(i)}$, which are restricted to zero. As a consequence, the corresponding correction, computed from $L_H e_H^{(i)} = 0$, is zero as well. Therefore, the spectral radius $\rho(K_h^H)$ is larger or equal to one. In this regard it makes no sense to apply the coarse-grid correction as a standalone solver.

3.1.2 Aliasing of Fourier components

Because the coarse grid receives fine-grid quantities we have to answer the question of what happens to functions when transferring them. The transfer of Fourier components like $\varphi_h^{\ell,m}(x_1, x_2)$ from Section 1.3 is particularly illustrative. For simplicity we assume standard coarsening ($H = 2h$) and injection of defects from the fine grid onto the coarse grid. Then

$$\Omega_{2h} = \{(x_1, x_2) \mid x_1 = \kappa_1 H = 2\kappa_1 h, x_2 = \kappa_2 H = 2\kappa_2 h; \kappa_1, \kappa_2 \in \mathbb{Z}\}$$

consists of all even-numbered points of Ω_h . Injection of fine-grid functions is nothing else than the evaluation of these functions at coarse-grid places. For a fine-grid component $\varphi_h^{\ell,m}$ at $(x_1, x_2) = (2\kappa_1 h, 2\kappa_2 h) \in \Omega_{2h}$ we have

$$\varphi_h^{\ell,m}(x_1, x_2) = \sin(\ell\pi 2\kappa_1 h) \cdot \sin(m\pi 2\kappa_2 h) = \varphi_{2h}^{\ell,m}(x_1, x_2),$$

showing that modes maintain their wave number.

The next question concerns the representation of high-frequency h -grid modes (i.e., $\max\{\ell, m\} \geq n/2$; compare with Section 1.3.3) on the coarse grid. It turns out that they cannot be represented on the coarse grid, because they coincide with certain low-frequency components. More precisely, for low-frequency modes ($1 \leq \ell, m < n/2$) we have

$$\varphi_h^{\ell,m}(x_1, x_2) = \varphi_h^{n-\ell, n-m}(x_1, x_2) = -\varphi_h^{n-\ell, m}(x_1, x_2) = -\varphi_h^{\ell, n-m}(x_1, x_2)$$

for $(x_1, x_2) \in \Omega_{2h}$. Hence the high-frequency modes cannot be distinguished on Ω_{2h} . They are not “visible” on the coarse grid.

Example 3.1 (Aliasing of Fourier modes). Any h -grid mode with ℓ (or m) equal to $\frac{n}{2}$ is mapped onto the coarse-grid zero function, because we have

$$\varphi_h^{\frac{n}{2}, m}(\kappa_1 H, \kappa_2 H) = \varphi_h^{\frac{n}{2}, m}(2\kappa_1 h, 2\kappa_2 h) = \sin(\pi\kappa_1) \cdot \sin(m\pi 2\kappa_2 h) = 0$$

for all nodes of Ω_{2h} .

Let $n = 32$, $h = \frac{1}{32}$, $H = 2h = \frac{1}{16}$. W.r.t. Ω_h , $\ell = 10$ refers to a low-frequency component. Regarding Ω_{2h} , this function is a high-frequency mode, because high-frequency wave numbers vary between 8 and 15 on Ω_{2h} .

The Fourier mode with wave numbers $(\ell, m) = (27, 2)$ is a high-frequency component w.r.t. Ω_h . Due to the aliasing phenomenon this function appears on the $2h$ -grid as a mode with wave numbers $(\ell, m) = (5, 2)$, which obviously refers to a low-frequency component. ◀

If we discuss visibility now with respect to wavelength it has to be remembered that on a grid with given mesh size those modes are visible which have a wavelength larger or equal to twice that mesh size. Consequently, on the H -mesh only modes with a wavelength larger or equal to $2H = 4h$ are visible. These are low-frequency fine-grid modes. On the coarse grid the Fourier modes with wavelengths smaller than $2H$ are invisible. They correspond to high-frequency modes on the fine mesh.

3.1.3 Correction scheme

From the above discussion it is a straightforward idea to suppress the transfer of invisible modes to the coarse grid. This can be achieved by eliminating high-frequency components on the fine grid by relaxation. The principles of smoothing and coarse-grid correction naturally lead to a new two-grid iteration procedure, which is called the two-grid Correction Scheme (CS) [4].

Algorithm 3.1 Correction scheme (CS)

- | | |
|--|--|
| (1) Presmoothing | $\bar{u}_h^{(i)} := S_h^{\nu_1} u_h^{(i)}$ |
| (2) Computation of residuals | $r_h^{(i)} := f_h - L_h \bar{u}_h^{(i)}$ |
| (3) Restriction of residuals | $r_H^{(i)} := R_h^H r_h^{(i)}$ |
| (4) Exact solution of the
coarse-grid problem | $L_H e_H^{(i)} = r_H^{(i)}$ |
| (5) Transfer of the correction | $\hat{e}_h^{(i)} := P_H^h e_H^{(i)}$ |
| (6) Correction | $\hat{u}_h^{(i)} := \bar{u}_h^{(i)} + \hat{e}_h^{(i)}$ |
| (7) Postsmoothing | $u_h^{(i+1)} = S_h^{\nu_2} \hat{u}_h^{(i)}$ |

The iteration operator of this two-grid method reads

$$M_h^H = S_h^{\nu_2} K_h^H S_h^{\nu_1}. \quad (3.2)$$

The iterative use of this operator finally leads to the formal description of the error reduction by the correction scheme:

$$e_h^{(i+1)} = (S_h^{\nu_2} (I_h - P_H^h L_H^{-1} R_h^H L_h) S_h^{\nu_1}) e_h^{(i)}. \quad (3.3)$$

A similar formula can be derived for the defect reduction using (1.21):

$$r_h^{(i+1)} = (L_h (S_h^{\nu_2} (I_h - P_H^h L_H^{-1} R_h^H L_h) S_h^{\nu_1}) L_h^{-1}) r_h^{(i)}. \quad (3.4)$$

The two-grid method is easily extended to a multigrid method by applying γ two grid cycles again to step (4) of Algorithm 3.1. It is only necessary to introduce a coarser mesh, for instance, a $4h$ -grid. Such a recursive use of coarser and coarser meshes can be continued until the coarsest problem has reached a size which allows a direct solution according to step (4). In many cases the coarsest grid may consist of a single interior point only. For example, a k -grid method using standard coarsening involves k different grids with uniform mesh sizes

$$h_j(m) = 2^{k-m} h =: h(m) \quad (m = 1, \dots, k; \quad j = 1, \dots, d). \quad (3.5)$$

Obviously, the mesh size becomes finer with an increasing index m and we have $h(k) = h$. Then it can be easily established by induction that the error transformation of a k -grid cycle is given by the following recursion [28, 60, 62]:

$$M_2^1 = S_2^{\nu_2} K_2^1 S_2^{\nu_1} = S_2^{\nu_2} (I_2 - P_1^2 (L_1)^{-1} R_2^1 L_2) S_2^{\nu_1} \quad (3.6)$$

$$\begin{aligned} M_{\ell+2}^1 &= S_{\ell+2}^{\nu_2} K_{\ell+2}^1 S_{\ell+2}^{\nu_1} \\ &= S_{\ell+2}^{\nu_2} (I_{\ell+2} - P_{\ell+1}^{\ell+2} (I_{\ell+1} - (M_{\ell+1}^1)^\gamma) (L_{\ell+1})^{-1} R_{\ell+2}^{\ell+1} L_{\ell+2}) S_{\ell+2}^{\nu_1} \\ &\text{for } \ell = 1, \dots, k-2, \end{aligned} \quad (3.7)$$

where the sub- and superscripts of the different operators are abbreviations for the related mesh sizes $h(m)$ of the k involved grids. γ denotes the cycle index.

The two-grid analysis provides the spectral radii and the norms of the error and defect-reduction operators:

$$\rho(M_h^H) = \rho(L_h M_h^H L_h^{-1}), \quad \|M_h^H\|_S, \quad \|L_h M_h^H L_h^{-1}\|_S.$$

In the accompanying software these values are named

$$\rho(M_{2L}), \quad \|M_{2L}\|_s, \quad \|M_{2L}\|_d.$$

The corresponding three-grid values (see (3.7) with $k = 3$) are

$$\rho(M_{3L}), \quad \|M_{3L}\|_s, \quad \|M_{3L}\|_d,$$

respectively. Details on how to derive these estimates are given in [Chapter 6](#).

The effect of this algorithm is understood best when thinking in terms of frequencies. By smoothing on the h -grid, high-frequency fine-grid modes are damped by some smoothing steps. The remaining modes are visible on the next coarser mesh. They are transferred to the coarse grid. Again, the coarse-grid high-frequency components are damped by smoothing. Then low-frequency coarse-grid components remain, but they are visible on the next

coarser mesh. The same arguments can be applied further and further. Summarizing, most Fourier components are smoothed out by the relaxation procedure on that particular mesh where they are visible high-frequency modes. The remaining error modes are removed on the coarsest grid where an exact solution is assumed.

Having introduced the correction scheme it is appropriate to apply the algorithm to exactly that problem (compare Figure 1.1) where we started from with our analysis of basic iterative schemes. Different mesh sizes $h \in \{\frac{1}{64}, \frac{1}{128}, \frac{1}{256}\}$ have been used for the finest grid when applying the CS iteration. The coarsest grid always consists of only a single interior point. We choose the same components as in our previous discussion in Section 1.6. There we have shown convergence predictions and empirical convergence factors for an algorithm with two sweeps of red-black Gauss-Seidel smoothing, full weighting and bilinear interpolation for grid transfers, standard coarsening and V-cycling. Figure 3.1 shows the multigrid convergence history for both the errors and the residuals.

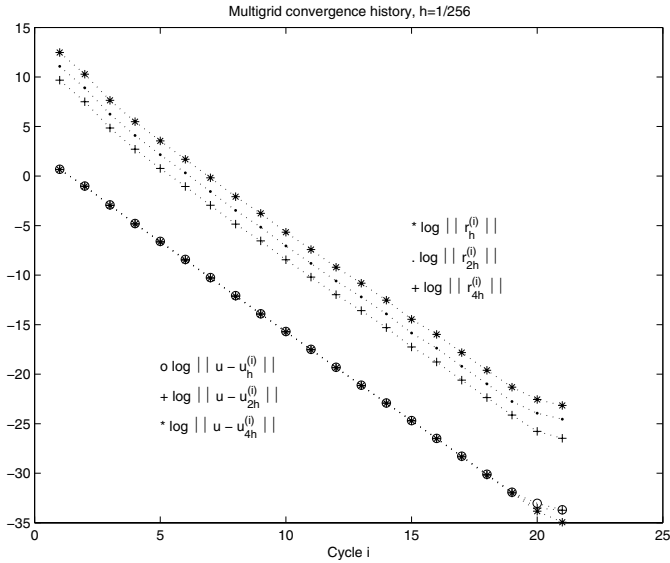


FIGURE 3.1: Multigrid convergence history, cycling with a finest mesh size $h = \frac{1}{128}$.

The most significant facts are:

1. Residual and error are reduced by almost the same factor of approximately 0.1 per multigrid iteration.
2. The reduction does not depend on the mesh size of the finest grid level.

3. Two of these numerically cheap multigrid cycles reduce errors and residuals to about the same extent as four hundred steps of the single-grid method (Jacobi relaxation) applied in Section 1.3.1. The numerical work for one cycle is about that of four iterations of the single-grid method applied on the finest grid.

3.2 FULL MULTIGRID

We continue our brief survey of multigrid principles with a description of full multigrid (FMG).

To solve $L_h u_h = f_h$ by multigrid, a hierarchy of k grids, $\Omega_h = \Omega_k, \dots, \Omega_1$, with decreasing mesh sizes (3.5) is introduced. (For ease of notation, we use the indexing with level numbers instead of using the mesh size directly, compare with Section 3.1). The cycling starts on grid level k and proceeds to coarser grids. Within this iterative approach the different grids are applied to compute corrections to the approximation on the next finer level.

Moreover, as for many other iterative approaches, the coarser grids can be used to compute an initial guess. It is straightforward to combine this idea known as “nested iteration” with multigrid cycling. The resulting FMG algorithm is easily described.

Algorithm 3.2 Full multigrid

With the hierarchy of grids $\Omega_h = \Omega_k, \Omega_{k-1}, \dots, \Omega_1$ perform

1. *on the coarsest level Ω_1 solve $L_1 u_1 = f_1$*
2. *for $m = 2, \dots, k$*
 - (a) *interpolate the initial guess $u_m^{(0)} = \mathcal{I}_{m-1}^m u_{m-1}$*
 - (b) *start from this initial value on level m with n_m multigrid cycles to calculate the new approximation u_m .*

To clearly distinguish the interpolation operator \mathcal{I}_{m-1}^m from the interpolation of corrections it is called FMG-interpolation.

Note that FMG is no fixed algorithm. It offers a wide range of choices. Instead of V-cycles any other cycle type might be used. It is not necessary to start the nested iteration on the coarsest grid which is used for multigrid. A proper choice for the number of multigrid cycles n_m to be applied on each level depends on the convergence factor of the multigrid cycle under consideration. Assuming “typical” multigrid convergence, $n_m = 1, 2$ is usually sufficient; see, for example, [62].

There are two main features of FMG which make it superior to pure multigrid cycling. Firstly, it can be shown that FMG computes an approximation

to the solution of the PDE up to the accuracy of the discretization error. This desirable feature mainly depends on the “polynomial order” of the FMG interpolation, the cycle index γ , and on a sufficiently small norm of the two-grid operator [60, 62]. (For the definition and a discussion of the order of transfer operators, refer to Section 7.1.) In practice, the designer of multigrid methods will look for small two- or three-level values with the help of **LFA** and then he can be sure that FMG will work properly. The second property of FMG is that it reaches this accuracy with an amount of work which is *linearly* dependent on the number N of unknowns characterizing an “optimal order” method or “ $O(N)$ -method”.

The potential of FMG can be easily demonstrated when solving our standard problem. Within just a single FMG step errors as well as defects are reduced up to the same amount which is obtained by the multigrid correction scheme after seven to ten iterations. Figure 3.2 shows the multigrid con-

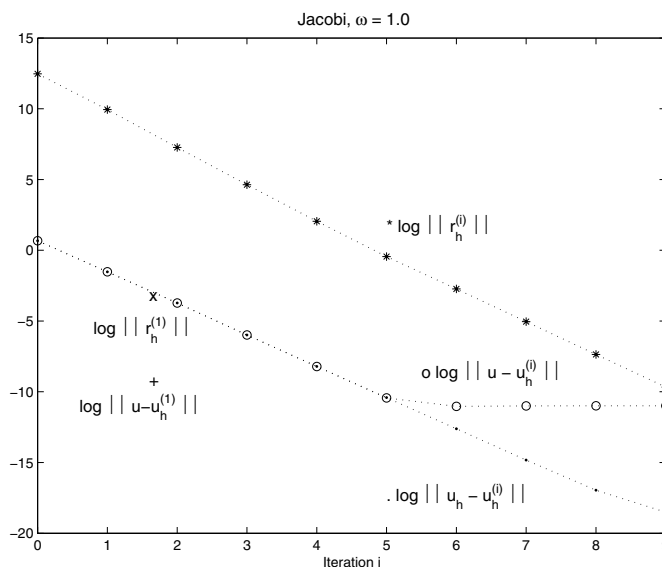


FIGURE 3.2: Multigrid convergence history, cycling (*, o) versus FMG (\times , +), finest mesh size $h = \frac{1}{128}$.

vergence history in a logarithmic scale for both the errors and the residuals computing the solution of an example with nonvanishing discretization error ($u(\mathbf{x}) = x_1^5 + x_2^6$). Multigrid iteration reduces the algebraic error $u_h - u_h^{(i)}$ well, but after some iterations, the total error $u - u_h^{(i)}$ has reached the size of the discretization error and cannot be reduced any further. FMG starts from the coarsest mesh, always performing just a single cycle per level. Because

we use $V(1,1)$ -cycles, the numerical effort for such an FMG step is about 1.6 times larger than that for one cycle starting on the finest grid. The resulting norm of both residual and error after one FMG cycle is marked by \times and $+$, respectively. This shows the potential of FMG: the accuracy of discretization error is reached with substantially less numerical effort compared to pure multigrid cycling.

3.3 **xlfa** FUNCTIONALITY—AN OVERVIEW

The user interface **xlfa** has been designed to allow for an easy modification of input parameters for the Fourier analysis program **LFA**. The multigrid components and postprocessing functions like plotting selected results can be chosen by either using the mouse (standard case) or the keyboard. **xlfa** is written in tcl/tk, an interpreted scripting language. Thus, if an appropriate interpreter is present **xlfa** can be run on nearly every UNIX-, LINUX-, or Windows-like operating system. More detailed information on functionality, installation requirements, and installation steps are given in the user's manual which can be found on the accompanying CD-ROM.

The initial module (Figure 1.5) of the graphical environment is a tabbed register panel which supports the selection of the problem class (2Dscalar, 2Dsystems, 3Dscalar) and the choice of a concrete PDE problem. There is a representative set of standard problems to work with. Nevertheless, the user is allowed to define and include new applications into the list.

The main window consists of four sections: the *menu bar*, the *button bar*, the *parameter display*, and the *problem display*; see Figure 3.3. To select any function the mouse is used, but there are shortcuts to use the keyboard, too.

3.3.1 Menu bar

The **File** menu supports loading previously used options or saving them. This menu is also used to quit **xlfa**. When working with the **Parameters**-menu the input information for the Fourier analysis program **LFA** is created. There are submenus where the output of **LFA** is specified, where relaxation-related selections can be made, and where components for the coarse-grid correction are chosen. Also, the type of discretization and the grid size can be specified here. There are default **xlfa**-parameters and options for each of the predefined applications. They can be saved as another option file to be loaded via the **File** menu above.

The **Problem** menu can be used to compile and run the current application. Moreover, it may serve to create a new application. To do so, one simply has to specify the related stencil of the new application; compare with Section 4.4.

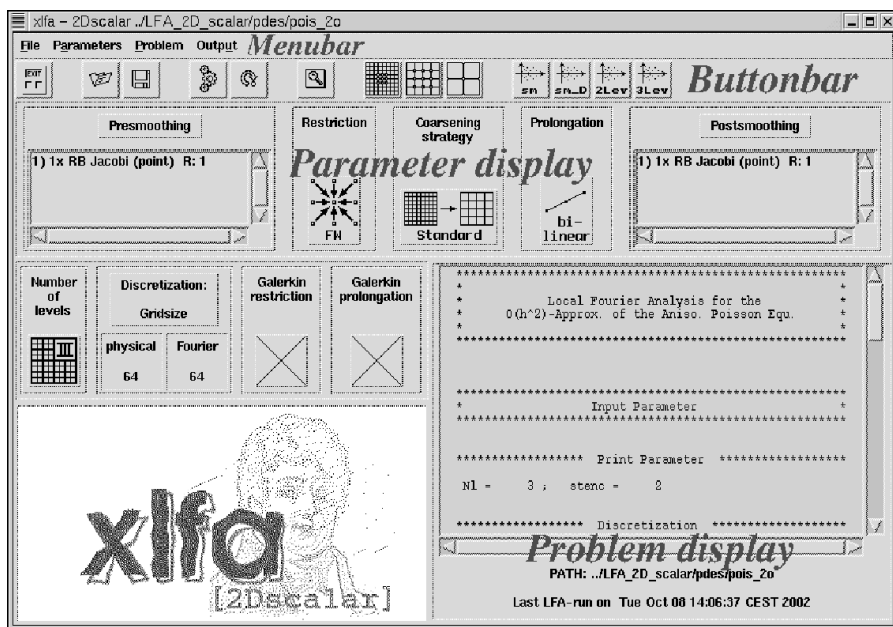


FIGURE 3.3: The main window of **xlf a** revisited.

The **Problem** menu supports editing such stencil files as well as incorporating them into **LFA**. Finally, the linked application may be started from this menu. Usually, the **Output** menu will be used to select the style of **LFA**-related output information. Internal files with detailed information and compact data about convergence rates and operator norms can be shown. Eigenvalues of different operators may be selected for visualization purposes here, too.

3.3.2 Button bar

The **Button bar** (Figure 3.4) provides a selection of often-used actions to be initiated by a single click with the mouse button. The first category concerns the preparation (definition of options, compilation, and execution) of the special application. The second class of buttons allows a fast selection of interactively displayed information such as convergence rates, stencils for selected levels, and distributions of eigenvalues.

3.3.3 Parameter display

The **Parameter display** (Figure 3.5) shows most of the actually chosen multigrid-related parameters. The graphical representations are self-explanatory. When clicking on list boxes or icons in this part of the display, a



FIGURE 3.4: The Button bar.

dialog is opened and the selection of different components is offered. The new selection is displayed by an appropriate icon in this part of the display, again.

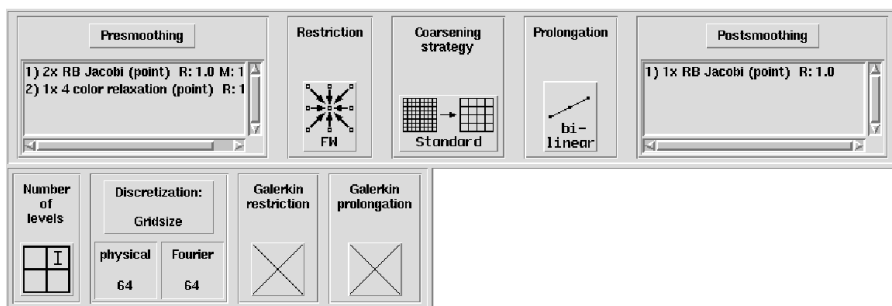


FIGURE 3.5: The Parameter display.

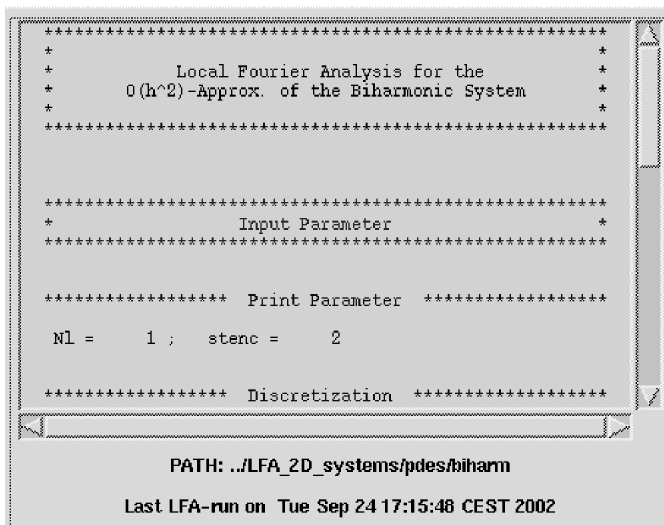
The number of levels to be used within the analysis is of importance for the type of analysis. If only one level is chosen, the single grid (or smoothing) analysis is carried out. Otherwise a two- or three- grid analysis is performed. In principle, the results are the closer to multigrid results when more levels are involved. The selection of components has to be in accordance with the selected number of levels.

3.3.4 Problem display

The Problem display (Figure 3.6) shows the output of the last **LFA** execution. The path of the currently selected problem run is displayed, too.

3.4 IMPLEMENTED COARSE-GRID CORRECTION COMPONENTS

The description of Algorithm 3.1 already shows that there are several algorithmic details to be chosen properly. An appropriate selection for obtaining



```

*****
*                               *
*   Local Fourier Analysis for the   *
*   O(h^2)-Approx. of the Biharmonic System *
*                               *
*****

*****
*                               *
*   Input Parameter                 *
*                               *
*****

***** Print Parameter *****

Nl =    1 ;   stenc =    2

***** Discretization *****

PATH: ../LFA_2D_systems/pdes/biharm
Last LFA-run on Tue Sep 24 17:15:46 CEST 2002

```

FIGURE 3.6: The Problem display.

an efficient multigrid strategy for a given problem is not trivial. Nevertheless, by theory and experience there exist recommendations for save and robust components.

In the following we consider each of the components separately which can be selected in **LFA**. The description of multigrid has not yet offered any feature which would limit the use of multigrid to scalar equations. Comparable to the scalar case the transfer of quantities and the proper selection of relaxation schemes becomes crucial. Thus it is convenient to describe the multigrid components for scalar equations and for a system of equations simultaneously. However, although the multigrid principles remain, the efficient treatment of systems of partial differential equations is definitely *not* straightforward and cannot be deduced from pure intuition or experience. Especially in these cases, **LFA** yields valuable and reliable information about the performance of different multigrid methods.

3.4.1 Discretization and grid structure

Discretization technique and grid structure strongly depend on the differential operator and the domain under consideration. In most cases the underlying geometric information is used. For a large class of problems vertex-centered discretization schemes are favored, where the unknown quantities are located at grid points. As previously mentioned, for many systems of PDEs in fluid-flow calculations, cell-oriented and staggered approaches are used. This does not limit the use of multigrid, but programming the compo-

nents becomes technically more demanding. The accompanying software is mainly designed for vertex-centered grids. However, local Fourier analysis for cell-centered [44] grids is addressed in Section 7.3 whereas the staggered case can be found in [10, 46, 70].

3.4.2 Coarsening strategies

The coarsening defines the hierarchy of grids to be used within the multigrid method. Regular grids allow coarsening rules which are easy to use. Doubling the mesh size from G_h to G_H with $H = 2h$, is known as standard coarsening. A second uniform coarsening is red-black coarsening where the next coarser mesh is created by omitting every other point. The new grid corresponds to a rotated grid with $H = \sqrt{2}h$.

A basic nonuniform procedure doubles the mesh size only with respect to a subset of spatial directions. For instance, x_1 - and x_2 -semicoarsening for two-dimensional applications yield $\mathbf{H} = (H_1, H_2) = (2h, h)$ and $\mathbf{H} = (h, 2h)$, respectively. In three dimensions, the number of semicoarsening strategies increases. The different coarsening strategies implemented in **LFA** are illustrated in the middle part of the **coarse grid correction** dialogue box shown in [Figure 3.7](#).

Restriction and prolongation have to be selected in accordance with the coarsening strategy, see below.

3.4.3 Coarse-grid operator

The discretization dialogue ([Figure 3.8](#)) serves to adapt the discretization approach. This concerns both the size of the stencil and the choice of the coarse-grid operator. Additionally, the size of the discrete meshes in physical space and in Fourier space (wave numbers) can be selected. For the non-experienced user, we always recommend selecting the same value for both parameters. The finer the discretization in the frequency space the higher the computational cost of the **LFA** calculations will be.

The coarse-grid operator L_H has been introduced up to now by the only condition that it is “similar” to the fine grid operator L_h . The most natural definition of L_H is to directly apply the same discretization technique as on the finest grid. This will be referred to as *discretization coarse-grid approximation* (DCA) as in [65]. Alternatively, L_H can be chosen using the Galerkin operator $L_H = R_h^H L_h P_H^h$, denoted by *Galerkin coarse-grid approximation* (GCA). Prolongation operator P_H^h and restriction operator R_h^H are often selected to be adjoint to each other. The possible choices for R_h^H and P_H^h shown in [Figure 3.8](#) are explained in Sections 3.4.5 and 3.4.6. Another possibility supported by the accompanying software is the selection of a *user-defined coarse-grid approximation* (UCA) which has to be specified in a subroutine called **starH.f**. An example is given in Section 4.1.2.

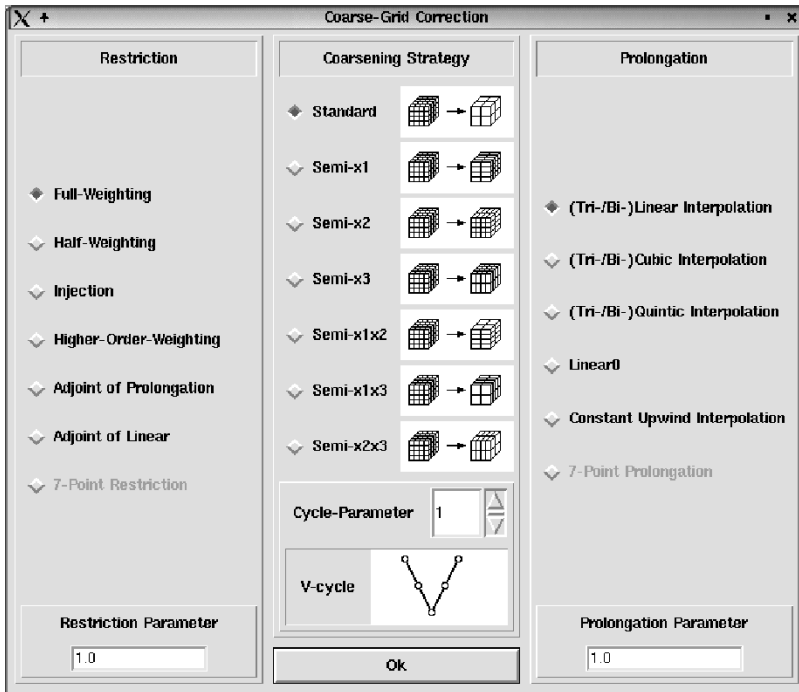
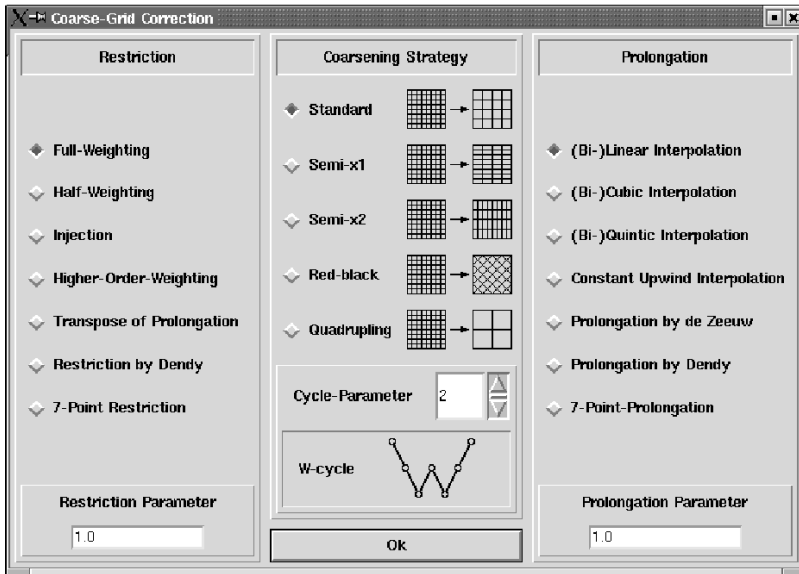


FIGURE 3.7: Coarse-grid correction dialogue for two- and three-dimensional cases.

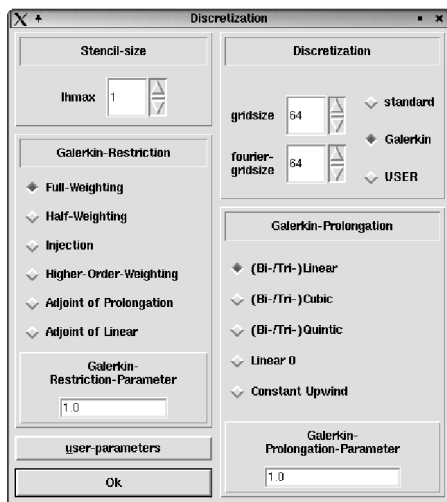


FIGURE 3.8: Discretization dialogue.

3.4.4 Multigrid cycling

The cycle parameter γ of recursive calls of the two-grid method to solve the coarse-grid problem determines how often a certain grid level is used. Usually γ equals one or two. $\gamma = 1$ leads to the so-called V-cycle and $\gamma = 2$ generates the W-cycle. Figure 3.9 motivates the names when using three levels.

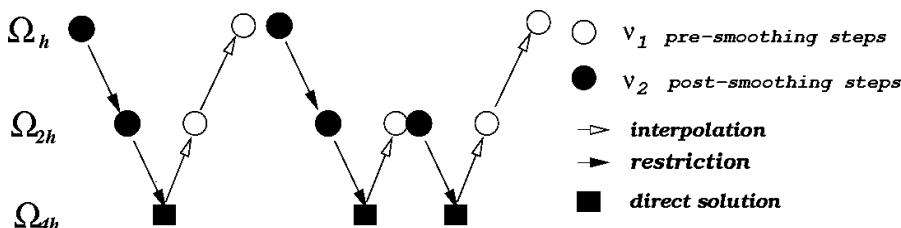


FIGURE 3.9: V- and W-cycles.

V- and W-cycle are the standard cycle types. Obviously, W-cycles are computationally more expensive than V-cycles. On the other hand, W-cycles are more robust and a general convergence theory exists; see, for example [28].

3.4.5 Restriction

For the description of the transfer operators listed in Fig. 3.7, we have to consider their dimensionality, which depends on the dimension of the underlying application *and* on the coarsening strategy. For example, x_1 -semicoarsening requires a one-dimensional restriction procedure (independent of the dimension of the underlying application) whereas standard coarsening requires a two-dimensional restriction procedure for two-dimensional applications and a three-dimensional restriction for three-dimensional problems. The practical application of transfer operators close to boundaries has to be done carefully for complicated domains; see, for example [62]. However, for our purpose such modifications are of minor interest since in the framework of local Fourier analysis we are considering infinite grids; compare with Section 2.2.2.

For $\Omega_H \subset \Omega_h$ the pure injection (INJ) first comes to mind. With standard coarsening we have $R_h^H v_h = v_h|_{\Omega_H}$. For half injection (HI) the injected value is multiplied by $\frac{1}{2}$. This particular restriction has been proven to be highly efficient for Poisson-type problems in connection with red-black Gauss-Seidel relaxation; compare with Section 4.1.1. Injection is easily programmed but averaging transfer operators are more robust for general problems. Full weighting (FW) is an often-recommended restriction. Higher-order weighting is rarely applied but might be useful in connection with higher-order differential equations like the biharmonic equation. If the grid coarsening is done in one space direction only—say x_1 —the corresponding stencils read as

$$\frac{1}{4} [1 \ 2 \ 1]_h^{2h} \quad \text{for full weighting and} \quad (3.8)$$

$$\frac{1}{32} [-1 \ 0 \ 9 \ 16 \ 9 \ 0 \ -1]_h^{2h} \quad \text{for higher-order weighting.} \quad (3.9)$$

The two- and three-dimensional analogues of (3.8) and (3.9) can be obtained by simple tensor products as described in Example 1.3. For example, the two-dimensional FW operator, given by

$$\frac{1}{4} [1 \ 2 \ 1]_h^{2h} \otimes \frac{1}{4} \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}_h^{2h} = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}_h^{2h} \quad (3.10)$$

might be applied in the case of standard coarsening for two-dimensional problems or in the case of (x_1, x_2) -semicoarsening for three-dimensional applications. Interpreting these stencils due to (1.5), the coarse-grid function $v_{2h}(x_1, x_2)$ ($(x_1, x_2) \in \Omega_{2h}$) after, e.g., full weighting in connection with standard coarsening and a two-dimensional application is given by

$$\begin{aligned} v_{2h}(x_1, x_2) = & \frac{1}{16} [v_h(x_1 - h, x_2 - h) + 2v_h(x_1, x_2 - h) + v_h(x_1 + h, x_2 - h) \\ & + 2v_h(x_1 - h, x_2) + 4v_h(x_1, x_2) + 2v_h(x_1 + h, x_2) \\ & + v_h(x_1 - h, x_2 + h) + 2v_h(x_1, x_2 + h) + v_h(x_1 + h, x_2 + h)]. \end{aligned}$$

Moreover, there are some restriction operators which cannot be described by tensor products of one-dimensional operators. That is, half weighting (HW) in two and three dimensions,

$$\frac{1}{8} \begin{bmatrix} 1 \\ 1 & 4 & 1 \\ 1 \\ 1 \end{bmatrix}_h^{2h}, \quad \frac{1}{12} \left[\begin{bmatrix} 1 \\ 1 \end{bmatrix}_h^{2h} \begin{bmatrix} 1 & 6 & 1 \\ 1 \end{bmatrix}_h^{2h} \begin{bmatrix} 1 \\ 1 \end{bmatrix}_h^{2h} \right],$$

seven-point restriction in two dimensions,

$$\frac{1}{8} \begin{bmatrix} 1 & 1 \\ 1 & \underline{2} & 1 \\ 1 & 1 \end{bmatrix}_h^{2h}, \quad (3.11)$$

the adjoint of linear interpolation in three dimensions due to [65]

$$\frac{1}{16} \left[\begin{bmatrix} 1 & 1 \\ 1 & \underline{1} \end{bmatrix}_h^{2h} \begin{bmatrix} 1 & 1 \\ 1 & \underline{2} & 1 \\ 1 & 1 \end{bmatrix}_h^{2h} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}_h^{2h} \right], \quad (3.12)$$

and the operator-dependent restrictions $R_h^{2h}(L_h)$ from [18, 77]. In addition, it is possible to apply the transpose of the selected prolongation operator as a restriction.

The generalization of the above restrictions to systems of equations is trivial (except for matrix-dependent restrictions, discussed in the following subsection). Here, each variable has to be treated separately by the selected restriction operator.

3.4.6 Prolongation

The transfer of coarse-grid information to the next finer one, especially prolongating the smooth correction, is often done by (bi-, tri-)linear interpolation. Other possible choices are (bi-, tri-)cubic or even (bi-, tri-)quintic interpolation. A particular simple (but usually not very efficient) choice is to use constant upwind interpolation only. Linear interpolation in two dimensions is adjoint to a seven-point restriction (3.11). It is exact for functions $v(x_1, x_2) = 1, x_1, x_2$, and takes place in triangles. In three dimensions, linear interpolation is adjoint to (3.12) and takes place in tetrahedra [65].

Depending on the problem and on the quantity to be transferred more sophisticated interpolation methods are needed. Popular alternatives are matrix-dependent prolongations $P_{2h}^h(L_h)$ [18, 32, 65, 77] which are based on the discretization operator under consideration. These matrix-dependent transfer operators are defined for two-dimensional *scalar* equations which can be represented by compact 9-point stencils in connection with standard coarsening. However for $(q \times q)$ *systems* of equations, one can use matrix-dependent

prolongations $P_{2h}^h(L_h^{i,i})$ and restrictions $R_h^{2h}(L_h^{i,i})$ based on the diagonal scalar operators $L_h^{i,i}$ ($i = 1, \dots, q$) as long as they are given by compact 9-point stencils. Then, for the i -th unknown discrete function, $R_h^{2h}(L_h^{i,i})$ and $P_{2h}^h(L_h^{i,i})$ are used as transfer operators from coarse to fine grids and vice versa.

Similarly, one applies $P_{2h}^h(L_h^{i,i})$ and $R_h^{2h}(L_h^{i,i})$ ($i = 1, \dots, q$) if Galerkin coarsening is selected. More precisely, \mathbf{L}_{2h} is then composed of

$$L_{2h}^{i,j} = R_h^{2h}(L_h^{i,i}) L_h^{i,j} P_{2h}^h(L_h^{j,j}) \quad \text{for } i, j = 1, \dots, q.$$

Then, $P_{4h}^{2h}(L_{2h}^{i,i})$ and $R_{2h}^{4h}(L_{2h}^{i,i})$ ($i = 1, \dots, q$) are applied to construct the next coarser Galerkin discretization \mathbf{L}_{4h} , and so on. Note that such an application of matrix-dependent transfer operators is closely related to the “unknown approach” in an algebraic multigrid (AMG) setting for systems of PDEs; see, for example, [51].

Because interpolation creates high-frequency modes on the fine grid, post-smoothing is recommended after coarse-grid correction.

3.5 IMPLEMENTED RELAXATIONS

Relaxation methods play a dominant role in multigrid algorithms. There is an extensive list of iterative schemes which have been applied as smoothers in a multigrid algorithm [62, 65] ranging from basic relaxations as presented in Section 1.2 to more sophisticated methods like “ILU-type” smoothers based on an incomplete matrix decomposition. Here, we focus on those relaxation methods implemented in **LFA**.

3.5.1 Relaxation type and ordering of grid points

Iterative schemes require a systematic way to resolve the equations for the corresponding unknowns. Because an unknown is associated with some grid node, this scanning of equations is also called ordering of grid points. Jacobi iteration (JAC) does not depend on the ordering whereas for Gauss-Seidel-type relaxations (GS) different orderings yield different numerical results. Hence it can be expected that the ordering of equations may have some influence on the quality of smoothing. To distinguish the type of ordering we append the information in which sequence the grid points are visited. Standard orderings for two-dimensional applications are lexicographic (LEX), red-black (RB) in a checkerboard (point relaxation), or zebra (line relaxation) manner [62, 65].

Using a lexicographical ordering, the point $(\kappa_1 h, \kappa_2 h)$ is called “smaller” than $(\kappa'_1 h, \kappa'_2 h)$ if $\kappa_2 \leq \kappa'_2$ and in case of $\kappa_2 = \kappa'_2$ if $\kappa_1 < \kappa'_1$. For the checkerboard ordering the grid points are split into a set of red points $\Omega_h^R = \{(\kappa_1 h, \kappa_2 h) \mid \kappa_1 + \kappa_2 \text{ even}\}$ and into the black ones $\Omega_h^B = \Omega_h \setminus \Omega_h^R$. In a

similar way the four-color ordering (4C) requires splitting Ω_h into four disjoint subsets; compare with Section 5.4.7. Within every color the points are numbered lexicographically, again. An x_1 -zebra line ordering first collects all points belonging to red (even-numbered) lines $\Omega_h^R = \{(\kappa_1 h, \kappa_2 h) \mid \kappa_2 \text{ even}\}$ and then the black (odd-numbered) ones.

The possibility for theoretical analysis is the main reason to use Jacobi-type relaxation, although they often do not provide efficient smoothing. A first modification of standard Jacobi iteration is red-black Jacobi relaxation (RB-JAC), which is realized by two half-steps. The first one applies Jacobi iteration to all grid points (respectively equations) belonging to Ω_h^R . With new red values the second Jacobi step determines new values for Ω_h^B . A straightforward generalization is the four-color Jacobi relaxation (4C-JAC). Gauss-Seidel relaxation with lexicographic ordering of grid points (GS-LEX) works satisfactorily in many cases. Nevertheless, Gauss-Seidel relaxation with red-black ordering (RB-GS) yields better smoothing properties for Poisson-like equations.

Block-oriented schemes become attractive as soon as there occur anisotropies, for instance caused by strongly varying coefficients of the PDE. Block-relaxation schemes collect all equations corresponding to the block (for instance a column or a line) and solve this subsystem directly. Passing through all the columns or lines lexicographically and updating the values immediately establishes the lexicographic column or line Gauss-Seidel relaxation (x_1 -, x_2 -line GS-LEX). The denomination for red-black variants (x_1 -, x_2 -line RB-JAC) are self-explanatory.

The alternating direction line RB-JAC relaxation (AD-RB-JAC) performs four different steps in a “symmetric” manner: x_1 -line RB-JAC for odd-numbered lines followed by x_1 -line RB-JAC for even-numbered lines followed by x_2 -line RB-JAC for even-numbered columns followed by x_2 -line RB-JAC for odd-numbered columns.

Apart from these standard smoothing methods, it is also possible to choose the so-called KAPPA-smoother [49] which has been especially designed for higher-order upwind discretizations of convection-dominated flow problems.

In three dimensions the possible number of variants is increased by using plane relaxations, compare with Fig. 3.10. The orientation of the plane should be governed by the coefficients of the discrete problem, which will be specified by several case studies given in Section 4.2.1. A plane relaxation, directly solving for all unknowns belonging to this plane, is realized in many cases by a two-dimensional multigrid method itself.

The selection of one of the described relaxation methods and the number of pre- and postsmoothing steps can be made with the help of a dialog box pictured in Figure 3.10. Also particular parameters like weighting factors (see Example 1.5) or multistage parameters (defined in Section 3.5.3) may be modified. For both, pre- and postsmoothing up to 10 different selections (**Presmoothing 1-10**, **Postsmoothing 1-10**) can be made. Moreover, a Kaczmarz variant of the selected relaxation method might be applied,

which is based on the normal equations of the underlying application; compare with [62, 65] and Remarks 5.12 and 5.16.

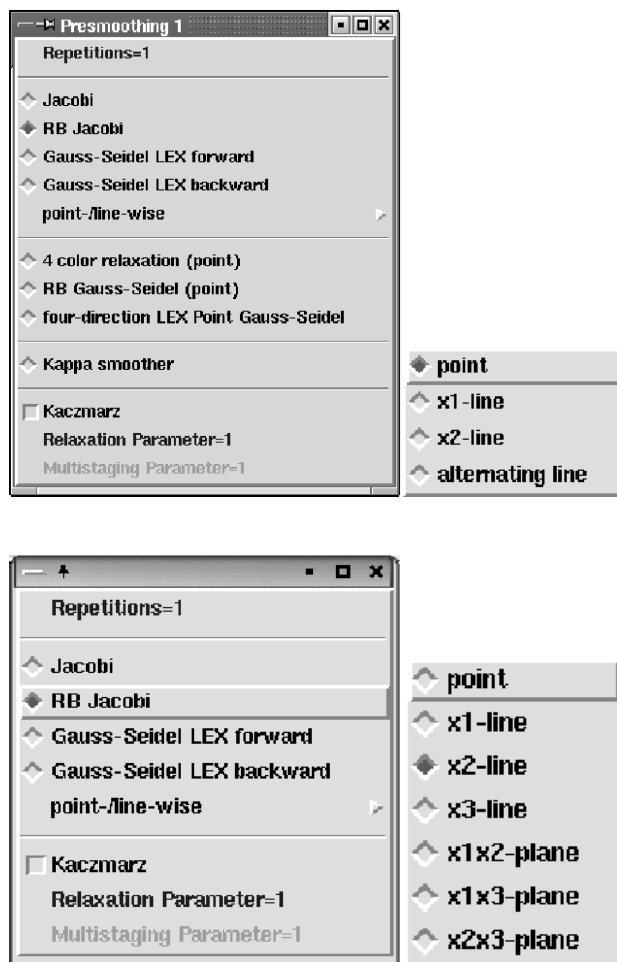


FIGURE 3.10: Possible smoothing methods for two- and three-dimensional scalar equations.

A general theoretical description of point- and block-relaxation methods of JAC-, GS-, and RB-JAC-type for d -dimensional problems will be given in [Chapter 5](#) in connection with local Fourier smoothing analysis. For implementation purpose and further details on the above-mentioned relaxation methods, we refer to the well-known basic multigrid literature, e.g., [62, 65].

3.5.2 Relaxation methods for systems

In order to characterize different types of relaxations for systems of PDEs, it is important to distinguish between *unknowns*, u^1, \dots, u^q , *grid points*, \mathbf{x} , and *variables*, $u_h^1(\mathbf{x}), \dots, u_h^q(\mathbf{x})$. An unknown is one of the functions being approximated, e.g., the pressure or the velocity components in the Stokes equations discussed in Section 4.3.2. The meaning of grid point is evident. The value representing the unknown at a particular grid point is called a variable.

The straightforward generalization of a *scalar* smoothing method described above to a *system* of PDEs is a collective (COL) or coupled relaxation. Like its scalar counterpart, this relaxation method sweeps over all grid points in a certain order, for example, in a lexicographic order or in a red-black manner. At each grid point \mathbf{x} in pointwise collective smoothing, all difference equations are solved simultaneously, i.e., the corresponding variables $u_h^1(\mathbf{x}), \dots, u_h^q(\mathbf{x})$ are updated simultaneously. This means that a linear $(q \times q)$ -system has to be solved at each grid point. The necessary adaptations for collective line or plane smoothers are straightforward.

Relaxation schemes which do not collectively solve the q equations at one grid point are called decoupled smoothers. We distinguish between two decoupled relaxation methods. Both variants consist of an inner and an outer sweep to update the variables. In the first variant (DEC1), the grid points are visited in an outer sweep where at each grid point the variables $u_h^1(\mathbf{x}), \dots, u_h^q(\mathbf{x})$ are updated one after the other (inner sweep). In the second variant (DEC2), each grid point is visited q times. In the first sweep over the grid points, only the variables $u_h^1(\mathbf{x})$ are calculated, in the second sweep the variables $u_h^2(\mathbf{x})$ are updated, etc., and finally the variables $u_h^q(\mathbf{x})$ are actualized. In other words, the variables are updated with the help of an outer sweep over the unknowns and an inner sweep over the grid points. For both variants, we are free to choose the order (lexicographic, red-black, etc.) in which the grid points are visited (outer sweep in the first variant, inner sweep in the second variant; see Fig. 3.11).

Distributive relaxations [10, 71] represent an important class of smoothing methods for systems of equations. They can be described as follows. To relax a system of PDEs $\mathbf{L}_h \mathbf{u}_h = \mathbf{f}_h$, we introduce a new variable \mathbf{v}_h by $\mathbf{u}_h = \mathbf{C}_h \mathbf{v}_h$ and consider the transformed system $\mathbf{L}_h \mathbf{C}_h \mathbf{v}_h = \mathbf{f}_h$. For example, \mathbf{C}_h is chosen in such a way that the resulting transformed system is suited for decoupled smoothing. The implemented distributor for the predefined Stokes equations is constructed like this (compare with Section 4.3.2). If $\mathbf{C}_h = \mathbf{L}_h^*$ is applied, the distributive version of the relaxation under consideration coincides with its Kaczmarz variant. For details on distributive relaxations, see [7, 10, 62, 71] and Section 5.5.2. Next to the predefined distributive relaxations, one might easily implement new variants by simply specifying the distributor stencil in the file `dist.f` explained in Section 4.4.

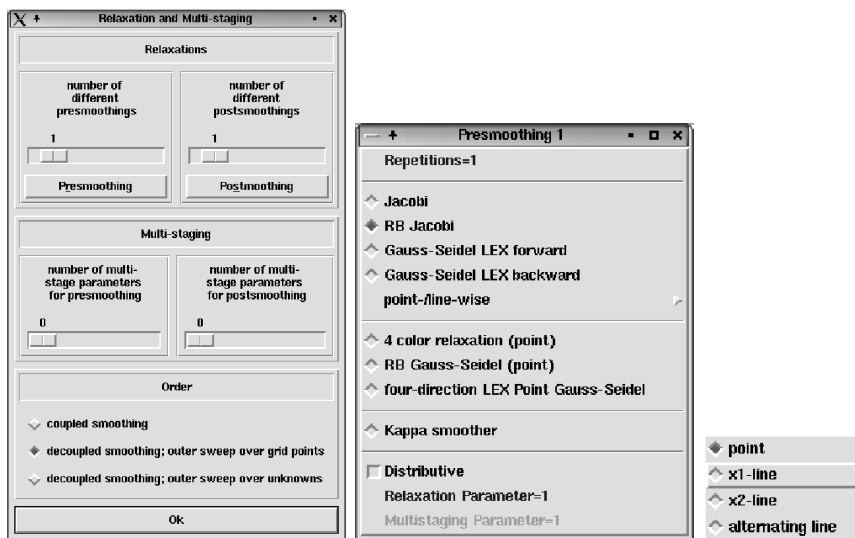


FIGURE 3.11: Selection of smoothing methods for two-dimensional systems.

3.5.3 Multistage (MS) relaxations

For any given smoothing method S_h , it is possible to construct an m -stage (MS) relaxation by

$$P_m(S_h) := \prod_{n=1}^m ((1 - \omega_n)I_h + \omega_n S_h) \quad (3.13)$$

with multistage parameters ω_n ($n = 1, \dots, m$) and grid identity I_h . The most popular multistage methods are of point Jacobi-type [31, 20, 15, 40]. The use of block- or red-black-type multistage methods is not very common, but it can be useful as well; see [67] and Section 4.1.5.

Note that the consecutive application of m damped point/block Jacobi relaxations ($n = 1, \dots, m$) is equivalent to point/block multistage Jacobi relaxation with multistage parameters $\omega_1, \dots, \omega_m$. For pattern or damped Gauss-Seidel relaxations it is not possible to express the relaxation parameters in terms of the multistage parameters. More details about MS relaxation are given in Section 5.6.

Chapter 4

USING THE FOURIER ANALYSIS SOFTWARE

Assuming that an application has been selected and that the compilation has created an **LFA**-executable we may start the experiments, for instance, by clicking the **run**-icon of the button bar (Fig. 3.4). Any error or warning of **LFA** is reported in the problem display. For each experiment the convergence predictions can be saved yielding a sequence of results in order to document the way to the most promising selection of components. We would like to point out that **LFA** does not only present some numbers (the expected multigrid convergence rates), it also provides the possibility to optimize algorithms and to debug multigrid software. The optimization aspect is easily understood. The debugging feature is a second aspect. Whenever the desired components for a given problem class have been selected and programming has started, **LFA** can be used to analyze the current status of the development step by step. As soon as a new component has been finished and is activated in the program, numerical experiments should be made to investigate the empirical behavior of the program. If this does not coincide with the predicted values for the given problem, there might be an error. In this way the analysis software may help in debugging all the components one after another. To use the **LFA** software in such a debugging mode is not only of value for multigrid beginners.

If a one-level (smoothing) analysis was requested, only the smoothing factors $\rho(S^\nu Q)$ and $\rho(S^\nu Q)_D$ will be shown where $\rho(S^\nu Q)_D$ refers to Dirichlet boundary conditions. Of course, local Fourier analysis usually neglects boundary conditions since it is based on infinite grids. However, it is possible to heuristically mimic the influence of Dirichlet boundary conditions in Fourier smoothing analysis which is motivated and explained in Section 5.3.3. The maximum information to be shown is listed in Table 4.1. The stencils of the chosen discretization and the corresponding measure of h -ellipticity (compare with Section 5.8) might be displayed as well. Considering systems of PDEs it is furthermore possible to take a look at a possible distributor and the resulting transformed system.

The visualization of the eigenvalues for the different operators as determined by the Fourier analysis program **LFA** is either requested by the main menu or by activating the corresponding icon of the button bar. The complex eigenvalues are represented in the complex plane (see, for instance, Figure 1.9

TABLE 4.1: Maximum information displayed by **LFA**

$\rho(S^\nu Q)$	smoothing factor
$\rho(S^\nu Q)_D$	smoothing factor for Dirichlet boundary conditions
$\rho(M_{2L})$	two-level asymptotic convergence factor
$\ M_{2L}\ _s$	spectral norm of the two-level error reduction operator
$\ M_{2L}\ _d$	spectral norm of the two-level residual reduction operator
$\rho(M_{3L})$	three-level asymptotic convergence factor
$\ M_{3L}\ _s$	spectral norm of the three-level error-reduction operator
$\ M_{3L}\ _d$	spectral norm of the three-level residual-reduction operator

in Section 1.6). A feature for zooming and moving around in order to look at the results in detail is available. For documentation purposes, an option to print and save as a postscript file is provided.

Experienced users may look at the input file **LFADAT** which has been generated by **xlfa** and at the output files of **LFA** itself. This option is chosen from the **Output** menu. Besides the already mentioned parameter input file there are files describing the discretization stencils and the different eigenvalue distributions.

In this chapter we will present both theoretical and corresponding numerical results for two-dimensional scalar, three-dimensional scalar, and two-dimensional systems of PDEs in Sections 4.1, 4.2, and 4.3, respectively. The convergence rates of the numerical experiments are denoted by $\rho_n(kL)$ which is the average defect reduction after 100 iterations for the corresponding solution method involving k levels. We choose such a large number of iterations because the theoretical predictions $\rho(M_{2L})$ and $\rho(M_{3L})$ refer to asymptotic convergence factors. Unless stated otherwise we use a zero right-hand side, homogeneous Dirichlet boundary conditions, and a random initial guess to avoid round-off errors.

Each of the predefined discrete applications of the accompanying software **LFA** is separated in its own subsection. Several interesting features of **LFA** like

- the superiority of the three-grid analysis over the classical two-grid analysis
- benefits of Galerkin coarse-grid approximation (GCA) versus discretization coarse-grid approximation (DCA)
- the application of user-defined coarse-grid approximations (UCA)
- the usefulness of multistage and relaxation parameters
- the comparison of coupled and decoupled smoothing for systems of PDEs
- the application of distributive relaxations for systems

- implications of operator-dependent transfer operators

are discussed by studying an assortment of the given examples in some detail.

Section 4.4 is dedicated to the implementation of a new application.

In the following, we will frequently apply a multigrid method with a fixed set of coarse-grid correction components and a varying smoother. We denote this method by MGO(“smoother”). It is characterized in the following table.

MGO(“smoother”):	<ul style="list-style-type: none"> • standard coarsening and DCA • bi- (2D) or trilinear (3D) interpolation and full weighting • smoother under consideration
------------------	--

For the different abbreviations of the particular relaxation at hand—like point GS-LEX or x_1 -line RB-JAC—refer to Section 3.5.

In order to get familiar with the accompanying software we encourage the reader to perform the same experiments as presented in the following sections.

4.1 CASE STUDIES FOR 2D SCALAR PROBLEMS

Two-dimensional scalar equations represent the first category of case studies under consideration. Typical difficulties concerning an efficient multigrid treatment like anisotropies, higher-order discretizations, mixed derivatives, or convection-dominated flow problems are discussed and possible remedies are presented. The investigated test cases are of increasing complexity: starting with a detailed discussion of the anisotropic diffusion equation—the most prominent example in connection with multigrid methods—we end up with a higher-order upwind discretization of the convection diffusion equation. Sections 4.1.1–4.1.5 are dedicated to basic multigrid topics whereas Sections 4.1.6–4.1.8 are more relevant for multigrid experts.

4.1.1 Anisotropic diffusion equation: second-order discretization

A discrete version using central differences of the anisotropic diffusion equation

$$Lu(\mathbf{x}) = -\varepsilon u_{11}(\mathbf{x}) - u_{22}(\mathbf{x}) \quad (\varepsilon > 0), \quad (4.1)$$

is represented by the stencil

$$L_h \triangleq \frac{\varepsilon}{h_1^2} [-1 \ 2 \ -1]_{h_1} + \frac{1}{h_2^2} \begin{bmatrix} -1 \\ 2 \\ -1 \end{bmatrix}_{h_2}. \quad (4.2)$$

By setting $\varepsilon = 1$, the Poisson equation is obtained representing the class of *isotropic* problems. The first algorithm coming into mind for the solution of the Poisson equation—denoted by **MG1**—is composed of

-
- | | |
|-------------|--|
| MG1: | <ul style="list-style-type: none"> • standard coarsening and DCA • bilinear interpolation and injection • point GS-LEX relaxation |
|-------------|--|
-

These selections are simply motivated by the fact that they can be most easily implemented. The empirically determined convergence factors for a V(2,1)-cycle involving six grids with finest mesh size $h = 1/64$ are satisfying: the error reduction is about 0.08, whereas the residual reduction is a little bit worse. This convergence behavior is nicely confirmed by **LFA**; compare with the left part of [Figure 4.1](#). The predicted smoothing rate for **MG1** is about 0.125. Due to a total number of three smoothing steps, this could have been estimated by taking the third power of 0.5, which is known to be the smoothing rate of lexicographic Gauss-Seidel relaxation (compare with Section 1.3.3 and Example 5.9). The predicted asymptotic behavior is about 0.09, both for two- and three-grid cycles. Note that even for an asymptotically converging algorithm the error (or defect) reduction norm might be larger than one. This is a common observation in connection with a simple injection of defects; compare with Remark 6.4 and Example 6.6. It can be avoided by an application of full weighting instead of injection.

Regarding the importance of smoothing and its influence on the multigrid convergence, a beginner's next step could be (the easiest one to program) to replace *lexicographic* Gauss-Seidel with *red-black* Gauss-Seidel yielding

-
- | | |
|-------------|---|
| MG2: | <ul style="list-style-type: none"> • standard coarsening and DCA • bilinear interpolation and injection • point GS-RB relaxation |
|-------------|---|
-

As a result one obtains a *diverging* algorithm. At first sight this might be surprising because red-black Gauss-Seidel relaxation has even better smoothing properties than lexicographic Gauss-Seidel as it has already been reported in Section 1.3.3. What does local Fourier analysis tell us about this experience? Replacing the lexicographical ordering by red-black ordering only the smoothing factor shows some improvement. Whenever more algorithmic details are incorporated for the prediction, the resulting estimates indicate divergence. The predicted values and the three-level eigenvalue distributions are shown in the right part of [Figure 4.1](#). The explanation for the diverging algorithm is that after one red-black Gauss-Seidel step the defect is zero at all black points. Hence, applying simple injection yields a wrongly scaled representation of the fine-grid residual on the coarser mesh. It is known that half injection is the remedy, leading to one of the most efficient multigrid algorithms for the Poisson equation [60, 62]: **MG3** which is detailed in the following table.

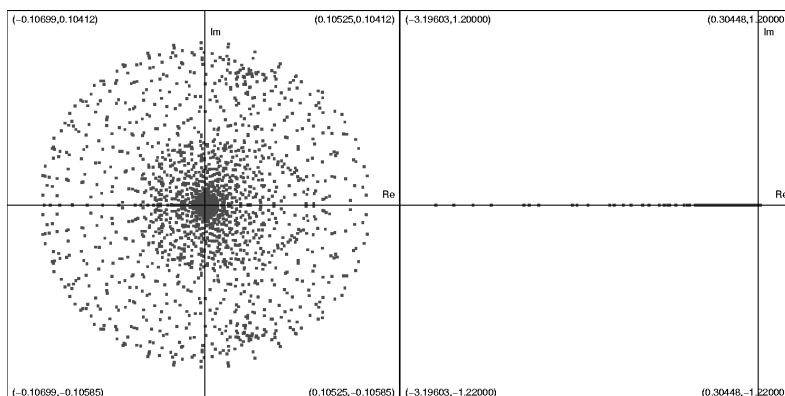
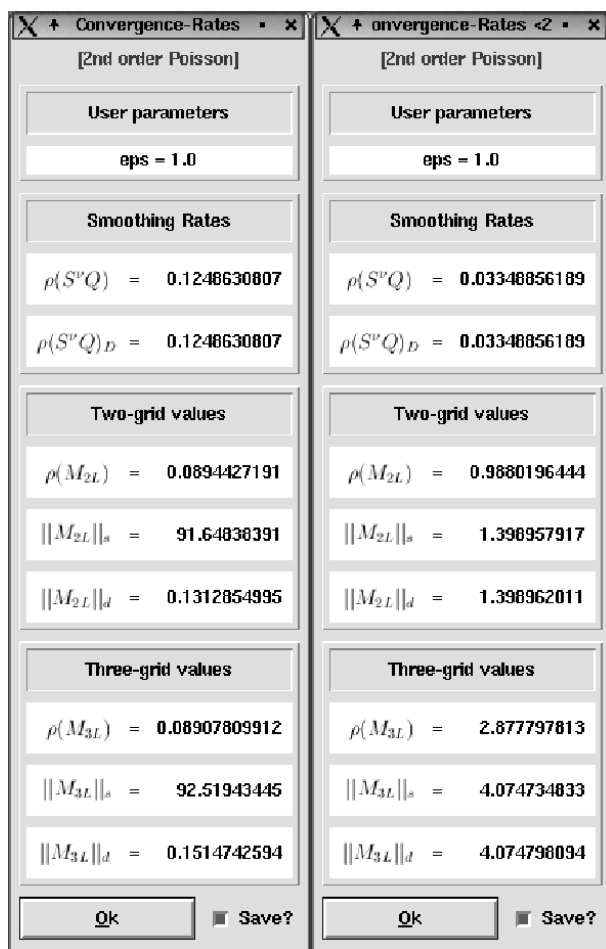


FIGURE 4.1: Local Fourier analysis estimates and distributions of eigenvalues for MG1 (left) and MG2 (right), $V(2,1)$ -cycles.

	• standard coarsening and DCA
MG3:	• bilinear interpolation and half injection
	• point GS-RB relaxation

Figure 4.2 shows these parameter selections for the two-dimensional Poisson equation. The corresponding W(2,1)-cycle (right part) offers an improved convergence compared to the V(2,1)-cycle (left part), both by prediction and in practice. The empirically determined asymptotic reduction factor involving six grids is $\rho_n(6L) = 0.0334$. But this gain which is proved by comparing the predicted and the empirical convergence speed has to be paid for by the invested numerical effort on the coarser levels.

Next, we discuss another efficient multigrid method, MG0(point RB-JAC), for the Poisson equation to demonstrate the different behavior of different cycle strategies and the ability of the three-grid analysis to predict these differences. Table 4.2 compares the analytical predictions from the one-, two-, and three-grid analyses with numerical reference values $\rho_n(3L)$ and $\rho_n(7L)$ for several 3- and 7-grid cycles. This table illustrates that even for such a simple and well-understood problem there is a difference between the performance of a V-cycle and a W-cycle and pre- and postsmoothing which cannot be displayed by Fourier two-grid analyses, whereas the different behavior of the cycle variants is very accurately predicted by the three-grid estimates $\rho(M_{3L})$. As it is seen in Table 4.2, one has to choose the multigrid W-cycle to retain

TABLE 4.2: Fourier values and numerically obtained asymptotic convergence factors for MG0(point RB-JAC) applied to the Poisson equation involving 3 and 7 levels; $h = 1/128$

Cycle	$\rho(S^2Q)$	$\rho(M_{2L})$	$\rho(M_{3L})$	$\rho_n(3L)$	$\rho_n(7L)$
V(1,1)	0.063	0.074	0.106	0.105	0.119
V(2,0)	0.063	0.074	0.133	0.132	0.170
V(0,2)	0.063	0.074	0.140	0.138	0.179
W(1,1), W(2,0), W(0,2)	0.063	0.074	0.074	0.073	0.073

the two-grid convergence factor. This is indicated by the Fourier analysis. The theoretical predictions for the two- and three-grid factors are equal only for the W-cycle. If we replace the 5-point discretization on the coarser grids by operators based on Galerkin coarsening with full weighting and bilinear interpolation, the V-cycle also leads to k -independent fast convergence. Using 4C-JAC relaxation for the resulting symmetric 9-point operators, we get for a V(1,1)-cycle: $\rho(S^2Q) = \rho(M_{2L}) = \rho(M_{3L}) = 0.063$. This value is validated by the corresponding numerical calculation for a 7-grid method.

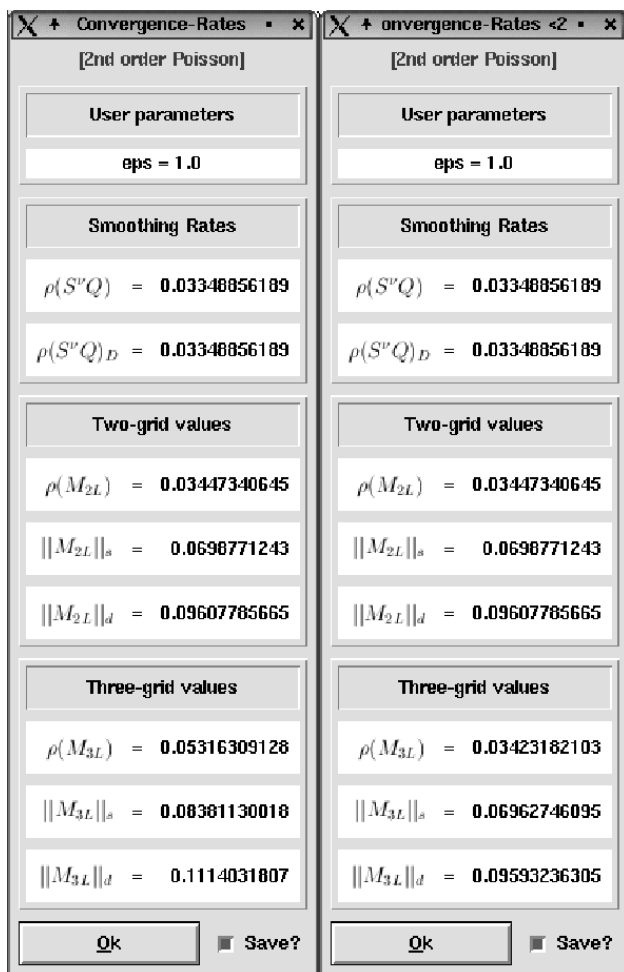


FIGURE 4.2: V(2,1)- (left) and W(2,1)-cycle of MG3 for the two-dimensional Poisson equation.

After the detailed discussion of the multigrid treatment for the isotropic diffusion equation (Poisson equation) we continue with the anisotropic case. As soon as the coefficients of the differential operator vary considerably, the multigrid strategy has to be adapted to this situation. The anisotropic model problem (4.1) with $\varepsilon \ll 1$ is used to explain the principles. First of all, we observe a severe deterioration of convergence factors for multigrid algorithms involving standard coarsening and point-relaxation methods. The failure of MG0(point RB-JAC) is due to the fact that point relaxation leads to smooth error terms only in the direction of “strong couplings” (i.e., in the x_2 direction in the present example). Hence it is not possible to represent the error that is highly oscillating in the x_1 direction on a coarser mesh. The first obvious approach to overcome this difficulty is based on a modification of the coarsening strategy, i.e., coarsening should only take place in the direction of smooth error terms. Instead of using an equispaced coarse grid to discretize (4.1) we can use a grid with mesh size $\mathbf{H} = (H_1, H_2) = (h, 2h)$. The corresponding stencil

$$L_{\mathbf{H}} \triangleq \frac{1}{4h^2} \begin{bmatrix} & -1 & \\ -4\varepsilon & 2(1+4\varepsilon) & -4\varepsilon \\ & -1 & \end{bmatrix}$$

shows a reduced anisotropy because of the factor 4 into the x_1 direction. This is the basis for the first strategy for anisotropic problems: maintaining the point relaxation but modifying the coarsening strategy from standard to semicoarsening into the x_2 direction (the direction of strong couplings) until the stencil entries are of approximately the same size, again. Assuming $\varepsilon \gg 1$, x_1 -semicoarsening has to be selected.

The second strategy is to work on the relaxation scheme instead, but fixing the coarsening strategy. This guarantees easy-to-handle grid structures. Errors which are smooth into both the x_1 and x_2 direction can be obtained by treating all strongly coupled unknowns together. For $\varepsilon \ll 1$, x_2 -line RB-JAC relaxation is recommended whereas for $\varepsilon \gg 1$, x_1 -line relaxation should be applied. This approach is often used because it maintains the simplicity of the algorithm and it can be expanded easily towards alternating direction line RB-JAC relaxation (AD-RB-JAC) which is an appropriate smoother if the size of the coefficients considerably varies over the domain (i.e., values $\varepsilon \ll 1$ and $\varepsilon \gg 1$ are realized in different regions of the domain under consideration). The smoothing factors of these relaxation schemes obtained by **LFA** are shown in Table 4.3 for different values of ε .

The robustness of the algorithm using alternating line relaxation for varying ε is obvious. Table 4.4 shows corresponding asymptotic three-grid factors for the particular choice $\varepsilon = 0.01$ using the same mesh size $h = \frac{1}{128}$. All values have been confirmed by numerical test calculations.

Figure 4.3 demonstrates the almost identical effect of applying the two different strategies for the above equation (4.1) using a V(2,1)-cycle with $\varepsilon = 0.01$. With MG4 given in the following table:

MG4:	• x_2 -semicoarsening and DCA
	• linear interpolation and full weighing
	• point GS-LEX relaxation

we have almost the same convergence predictions as for MG0(x_2 -line GS-LEX).

TABLE 4.3: Smoothing factors for anisotropic diffusion equation, standard coarsening

ε	point RB-JAC	x_1 -line RB-JAC	x_2 -line RB-JAC	AD-RB-JAC
100	0.980	0.125	0.980	0.122
10	0.826	0.125	0.826	0.102
1	0.250	0.250	0.250	0.048
0.1	0.826	0.826	0.125	0.102
0.01	0.980	0.980	0.125	0.122

TABLE 4.4: $\rho(M_{3L})$ for the anisotropic diffusion equation, $\varepsilon = 0.01$

	MG0(point RB-JAC)		MG0(x_2 -line RB-JAC)		MG0(AD-RB-JAC)
ε	V(2,1)	W(2,1)	V(2,1)	W(2,1)	W(1,1)
0.01	0.945	0.942	0.045	0.030	0.048

4.1.2 Anisotropic diffusion equation: fourth-order discretization

A fourth-order approximation of the anisotropic diffusion equation (4.1) reads

$$L_h \triangleq \frac{\varepsilon}{12h_1^2} [1 \ -16 \ 30 \ -16 \ 1]_{h_1} + \frac{1}{12h_2^2} \begin{bmatrix} 1 \\ -16 \\ 30 \\ -16 \\ 1 \end{bmatrix}_{h_2}. \quad (4.3)$$

We apply a V(1,1)-cycle of MG0(point RB-JAC) for the solution of (4.3) with finest mesh size $h = 1/64$. The resulting output of **xlfa** for this algorithm is shown in the left part of Figure 4.4. The asymptotic convergence rate of the two-grid method ($\rho(M_{2L}) = 0.122$) is not recovered by the three-grid analysis

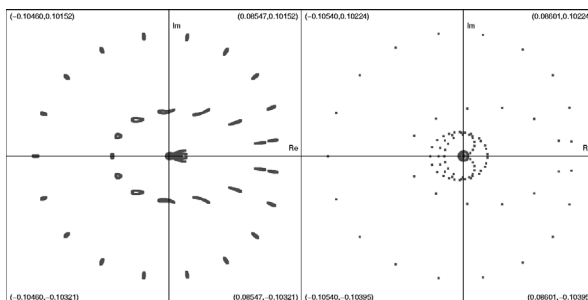
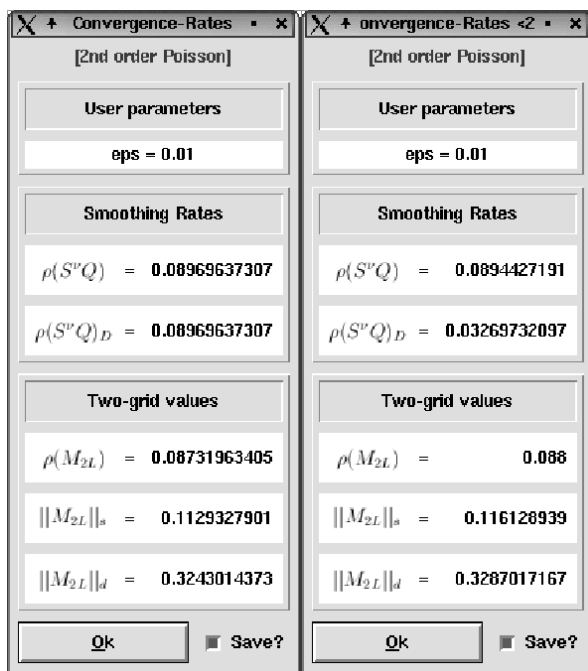
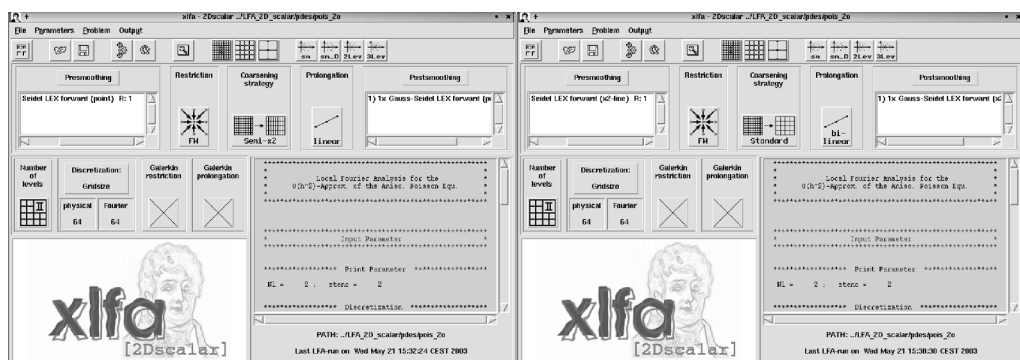


FIGURE 4.3: Exploiting different strategies for the anisotropic diffusion equation. MG4: (left) and MG0(x_2 -line GS-LEX) (right).

($\rho(M_{3L}) = 0.172$). This observation also holds for the corresponding norm values. It has to be expected (and has been validated by numerical experiments) that this deterioration gets even worse if more levels are involved. Switching to the more expensive W-cycle, it is possible to avoid this difficulty. As a remedy to overcome this undesirable feature of the V-cycle one could change the discretizations on coarser grids. Applying the second-order discretization from (4.2) leading to MG5:

	• standard coarsening and UCA
MG5:	• bilinear interpolation and full weighting
	• point RB-JAC relaxation

yields improved three-grid values as can be seen in the right part of Fig. 4.4. This particular choice for `starH.f` (see Section 3.4.3) has already been implemented in `xlfa`. The `problem` menu can be used in order to edit `starH.f`.

4.1.3 Anisotropic diffusion equation: Mehrstellen discretization

Another fourth-order approximation of (4.1) is given by the Mehrstellen discretization

$$L_h u_h(x) = R_h f_h(x)$$

represented by the stencils

$$L_h \triangleq \frac{\varepsilon}{12h_1^2} \begin{bmatrix} -1 & 2 & -1 \\ -10 & 20 & -10 \\ -1 & 2 & -1 \end{bmatrix}_h + \frac{1}{12h_2^2} \begin{bmatrix} -1 & -10 & -1 \\ 2 & 20 & 2 \\ -1 & -10 & -1 \end{bmatrix}_h \quad \text{and} \quad R_h \triangleq \frac{1}{12} \begin{bmatrix} 1 \\ 1 & 8 & 1 \\ 1 \end{bmatrix}_h.$$

With $\varepsilon = 1, h_1 = h_2 = h$ the Mehrstellen discretization for the Poisson equation is given by

$$L_h \triangleq \frac{1}{6h^2} \begin{bmatrix} -1 & -4 & -1 \\ -4 & 20 & -4 \\ -1 & -4 & -1 \end{bmatrix}_h. \quad (4.4)$$

Point JAC relaxation with an optimal overrelaxation parameter $\omega = 10/11$ leads to a particular simple but also efficient multigrid algorithm for (4.4). A V(1,1)-cycle of MG0(point ω -JAC) with finest mesh size $h = 1/64$ gives an asymptotic three-grid convergence factor of $\rho(M_{3L}) = 0.229$.

The default multigrid components as provided by the software distribution yield a V(1,1)-cycle of MG0(4C-JAC). This leads to very fast multigrid convergence demonstrated by an asymptotic three-grid factor of $\rho(M_{3L}) = 0.057$ for the Poisson equation.

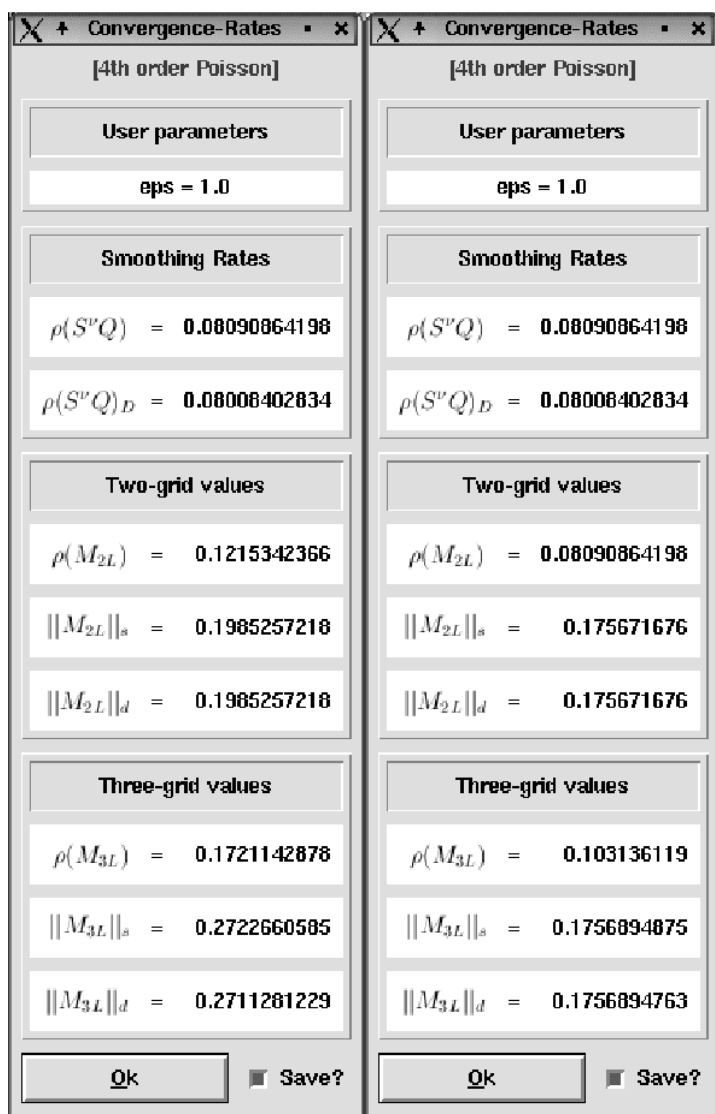


FIGURE 4.4: The effect of different coarse-grid discretizations; fourth-order MG0(point RB-JAC) (left), second-order MG5 (right) for the anisotropic diffusion equation.

4.1.4 Helmholtz equation

A second-order approximation of the Helmholtz equation,

$$Lu(\mathbf{x}) = -u_{11}(\mathbf{x}) - u_{22}(\mathbf{x}) + cu(\mathbf{x}) = f(\mathbf{x}) \quad (c \in \mathbb{R}),$$

is described by the stencil

$$L_h \triangleq \frac{1}{h^2} \begin{bmatrix} & -1 & \\ -1 & 4 + ch^2 & -1 \\ & -1 & \end{bmatrix}_h.$$

The Helmholtz equation is of high relevance because implicit time-discretization schemes for linear parabolic PDEs result in such types of equations to be solved within every time-step Δt (see, for example, [33]). The Helmholtz parameter c then usually depends on h and Δt : $c := c(h, \Delta t)$. The Poisson equation ($c = 0$) can be considered as the limit for $\Delta t \rightarrow \infty$.

The convergence predictions for the Helmholtz equation with $c > 0$ should be better than those for the Poisson equation, because of its increasing diagonal dominance which is a desirable feature for any iterative method [63].

Note that the typical smoothing and coarse-grid approximation difficulties known for the indefinite case (i.e., $c \ll 0$) [22, 62] cannot be investigated by local Fourier analysis based on infinite grids. These convergence difficulties are connected to the boundary conditions which have to be taken into account for a proper evaluation of multigrid convergence for the indefinite Helmholtz equation. Alternatively, an analysis based on the sine functions from Section 1.3 might be performed assuming Dirichlet boundary conditions. Details can be found, for example, in [22, 50].

4.1.5 Biharmonic equation

The biharmonic equation,

$$Lu(\mathbf{x}) = -(\partial_{11} + \partial_{22})(\partial_{11} + \partial_{22})u(\mathbf{x}), \quad (4.5)$$

is studied in order to demonstrate the usefulness of one- (or even multi-) stage parameters within a relaxation method. Equation (4.5) is sometimes transformed into a system of two Poisson-type equations in order to achieve better smoothing properties [4] and smaller stencils; see Section 4.3.1. Here, we keep the scalar problem and apply one-stage point RB-JAC to a second-order discretization given by the following stencil:

$$L_h \triangleq \frac{1}{h_1^4} [1 \ -4 \ 6 \ -4 \ 1]_{h_1} + \frac{1}{h_1^2 h_2^2} \begin{bmatrix} 2 & -4 & 2 \\ -4 & 8 & -4 \\ 2 & -4 & 2 \end{bmatrix}_h + \frac{1}{h_2^4} \begin{bmatrix} 1 \\ -4 \\ 6 \\ -4 \\ 1 \end{bmatrix}_{h_2}.$$

Figure 4.5 demonstrates the positive influence of an optimal one-stage parameter $\omega = 1.4$ taken from [67] (upper right part of the picture) compared to ordinary point RB-JAC relaxation (upper left part of the picture). Here, W(1,1)-cycles of MG0(point RB-JAC) are applied. Using this optimal one-stage relaxation leads to a complex eigenvalue distribution of the three-grid error-reduction operator (lower picture in Figure 4.5) in contrast to the real-valued spectrum for regular red-black Jacobi smoothing.

It is possible to construct *multistage* variants of RB-JAC relaxation (see, for example, [67]) with even better smoothing properties. However, it turned out in the Fourier three-grid analysis and in the numerical tests that it does not pay off to invest too much work into smoothing because the coarse-grid correction cannot reduce the low-frequency error components equally well. Therefore, we focus on one-stage RB-JAC smoothing.

4.1.6 Rotated anisotropic diffusion equation

Next we discuss the rotated anisotropic diffusion equation,

$$Lu(\mathbf{x}) = -(c^2 + \varepsilon s^2)u_{11}(\mathbf{x}) - 2(\varepsilon - 1)csu_{12}(\mathbf{x}) - (\varepsilon c^2 + s^2)u_{22}(\mathbf{x}) = 1 \quad (4.6)$$

on $\Omega = (0, 1)^2$ with $\varepsilon \geq 0$, $c = \cos \beta$, $s = \sin \beta$, $\beta \in [0^\circ, 360^\circ)$ and homogeneous Dirichlet boundary conditions $u(\mathbf{x}) = 0$ on $\Gamma = [0, 1]^2 \setminus \Omega$. This differential operator corresponds to $-u_{\xi\xi} - \varepsilon u_{\eta\eta}$ in a (ξ, η) -coordinate system, which can be obtained by rotating the (x_1, x_2) -system with an angle of β [65]. A second-order approximation of (4.6) reads

$$L_h \triangleq \frac{c^2 + \varepsilon s^2}{h_1^2} \begin{bmatrix} -1 & 2 & -1 \end{bmatrix}_{h_1} + \frac{2(\varepsilon - 1)c}{4h_1h_2} \begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}_h + \frac{\varepsilon c^2 + s^2}{h_2^2} \begin{bmatrix} -1 \\ 2 \\ -1 \end{bmatrix}_{h_2}.$$

We set $\beta = 45^\circ$. For $\varepsilon \rightarrow 0, \infty$ this equation is no longer elliptic. Using standard coarsening and Jacobi- or Gauss-Seidel-type smoothing, this leads to coarse-grid correction difficulties which limit the two-grid convergence to the factor 0.75; see Section 7.2. A simple recursive argument yields that the asymptotic V-cycle convergence on k grids is limited by $\rho_n(kL) \geq 1 - 4^{-k+1}$ ($\rho_n(2L) \geq 0.75$, $\rho_n(3L) \geq 0.9375, \dots$). These bounds can be established by Fourier two- and three-grid analysis as can be seen from Table 4.5, where MG0(point RB-JAC) is applied to the rotated anisotropic diffusion equation. Switching to a W-cycle leads to a slight improvement, which is also predicted by the three-grid analysis. In the context of Galerkin coarsening for the rotated anisotropic diffusion equation, it is interesting to investigate the multigrid method MG6 which is composed of

-
- | | |
|------|---|
| | <ul style="list-style-type: none"> • standard coarsening and GCA |
| MG6: | <ul style="list-style-type: none"> • matrix-dependent prolongation and restriction, de Zeeuw [77] • 4C-JAC relaxation |
-

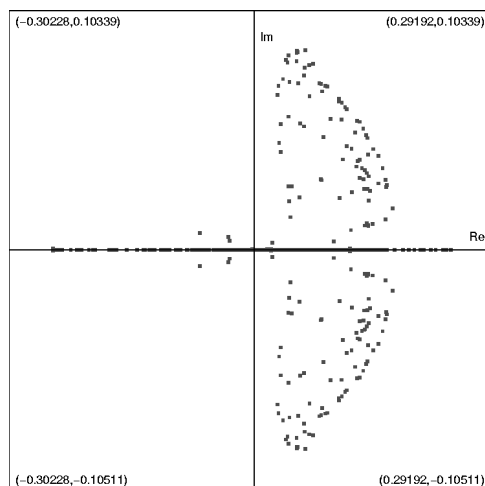
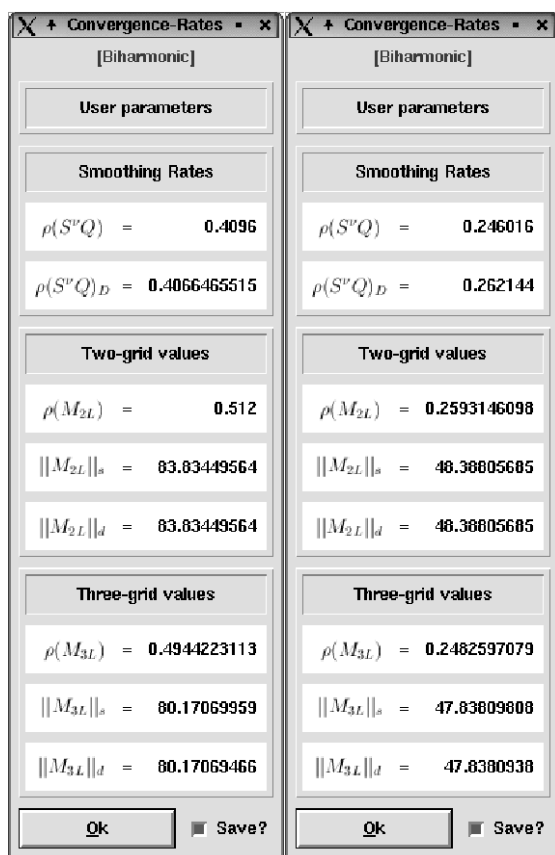


FIGURE 4.5: Influence of a multistage parameter for MG0(point RB-JAC) applied to the biharmonic equation.

TABLE 4.5: Fourier values and numerically obtained asymptotic convergence factors for MG0(point RB-JAC) involving 3 and 7 levels for the rotated anisotropic diffusion equation, $\beta = 45^\circ$, $\varepsilon = 10^{-5}$, $h = 1/128$

Cycle	$\rho(S^2Q)$	$\rho(M_{2L})$	$\rho(M_{3L})$	$\rho_n(3L)$	$\rho_n(7L)$
V(1,1)	0.35	0.76	0.94	0.92	0.95
W(1,1)	0.35	0.76	0.90	0.87	0.89

Comparing Tables 4.5 and 4.6, we see a remarkable improvement of the two-grid convergence factor. For this example, however, Fourier one- and two-grid analysis yield to some extent misleading results. At first sight, the coarse-grid correction problem seems to be solved since the two-grid value nearly recovers the smoothing factor. But if we look at the increasing three-grid values, it has to be expected that the multigrid convergence will further deteriorate. This is validated by the numerical reference values for the related 7-grid iterations.

TABLE 4.6: Fourier values and numerically obtained asymptotic convergence factors for MG6 involving 3 and 7 levels for the rotated anisotropic diffusion equation, $\beta = 45^\circ$, $\varepsilon = 10^{-5}$, $h = 1/128$

Cycle	$\rho(S^2Q)$	$\rho(M_{2L})$	$\rho(M_{3L})$	$\rho_n(3L)$	$\rho_n(7L)$
V(1,1)	0.28	0.36	0.63	0.61	0.87
W(1,1)	0.28	0.36	0.47	0.48	0.61

If the prolongation and restriction from [77] are replaced by the transfer operators from the blackbox multigrid package for nonsymmetric problems by Dendy [18], we find very similar results. These transfer operators are investigated in the next subsection.

Finally, we would like to mention that there are other multigrid components like ILU-type smoothers or nonstandard coarsening to overcome the coarse-grid correction difficulty efficiently. De Zeeuw [77], for example, combines his matrix-dependent Galerkin coarsening with a powerful smoother—the incomplete line LU decomposition (ILLU)—and obtains very fast multigrid convergence. These improvements might be verified by a Fourier three-grid analysis. Another possibility to deal with the difficulty caused by the mixed derivative is to apply multigrid as a preconditioner for a Krylov subspace method like GMRES. The Fourier analysis for this combined solution method [69] is briefly addressed in Section 7.4. In [68] it is shown, that *three-grid* methods can be

nicely accelerated by GMRES(m) with a small Krylov subspace. Thus it can be expected that the corresponding *multigrid* iterations are appropriate preconditioners as well, which has been confirmed by numerical experiments.

4.1.7 Convection diffusion equation: first-order upwind discretization

The next example we discuss in some detail is the convection diffusion equation with dominant convection. Here, it is important to distinguish between *entering* flows with an inflow and outflow boundary condition and *recirculating* flows for which no real inflow and outflow boundaries exist and where the boundary information is diffusing into the domain. In principle, efficient multigrid iterations can be constructed for upstream discretizations if relaxation methods are used with a downstream ordering of grid points. Then the relaxation acts not only as a smoother but also partly as a solver and takes care of problematic “characteristic” low-frequency error components [11]. For entering flows such smoothers can be found among standard relaxation methods, whereas for recirculating flows it is very difficult to construct a smoother with the desired property.

For convection-dominated rotating flow problems like

$$-\varepsilon \Delta u(\mathbf{x}) + a_1(\mathbf{x})u_1(\mathbf{x}) + a_2(\mathbf{x})u_2(\mathbf{x}) = 1 \text{ on } \Omega = (0, 1)^2 \quad (4.7)$$

with $\varepsilon = 10^{-5}$, $a_1(\mathbf{x}) = -\sin(\pi x_1) \cos(\pi x_2)$, $a_2(\mathbf{x}) = \sin(\pi x_2) \cos(\pi x_1)$,

$$u(\mathbf{x}) = \sin(\pi x_1) + \sin(13\pi x_1) + \sin(\pi x_2) + \sin(13\pi x_2) \text{ on } \Gamma = [0, 1]^2 \setminus \Omega,$$

the situation changes if standard smoothers are used, and we observe a similar coarse-grid correction difficulty as in the previous subsection. If DCA is applied combined with standard coarsening, the two-grid convergence is limited by the factor

$$1 - 2^{-p}, \quad (4.8)$$

where p denotes the order of the discretization; compare with Section 7.2 and [12]. This results in a deterioration of the V-cycle multigrid convergence on k grids, quantified by $1 - 2^{-(k-1) \cdot p}$.

Dirichlet boundary effects are neglected by the local Fourier analysis on infinite grids. For entering flows, high-frequency boundary information may propagate far into the domain, and thus it should be taken into account by a reliable analysis. This is done by the so-called *half-space* full multigrid (FMG) analysis in [11]. For recirculating flows, however, the influence of the domain boundary is negligible in the limit of small mesh size, and the (*infinite-space*) local Fourier analysis is again useful [12].

Recall from Section 2.3 that a direct application of the Fourier analysis is not possible dealing with operators $L_h(\mathbf{x})$ that are characterized by variable coefficients. Instead, the analysis is applied to the locally frozen operator at a fixed grid point ξ leading to the following approximation of two- and

three-grid factors in case of variable coefficients:

$$\rho(M_{kL}) = \rho(M_{kL}(L_h(\mathbf{x}))) := \sup_{\boldsymbol{\xi} \in \Omega} \rho(M_{kL}(L_h(\boldsymbol{\xi}))) \quad \text{for } k = 2, 3. \quad (4.9)$$

The smoothing factor and the corresponding norm values are defined accordingly. Consequently, the whole range of convection angles that occur in problem (4.7) has to be investigated. For an explicit calculation, we approximate (4.9) by a repeated application of the Fourier analysis to discretizations of $-\varepsilon \Delta u + a_1 u_1 + a_2 u_2$ with fixed $a_1 = \cos \beta$ and $a_2 = \sin \beta$ for multiples of 3° until the range of possible convection angles $\beta \in [0^\circ, 360^\circ]$ is passed through, as proposed in [65]. Then, the maximal values for $\rho(S^\nu(\beta)Q)$, $\rho(M_{2L}(\beta))$, and $\rho(M_{3L}(\beta))$ are assumed to be upper bounds for the smoothing and k -grid factors referring to (4.7). The calculation of the related norm values is carried out analogously.

A first-order upwind discretization of (4.7) is given by the following stencil:

$$\frac{\varepsilon}{h^2} \begin{bmatrix} & -1 & \\ -1 & 4 & -1 \\ & -1 & \end{bmatrix}_h + \frac{1}{2h} [-a_1 - |a_1| \ 2|a_1| \ a_1 - |a_1|]_h + \frac{1}{2h} \begin{bmatrix} a_2 - |a_2| & \\ & 2|a_2| \\ -a_2 - |a_2| \end{bmatrix}_h. \quad (4.10)$$

This discretization is studied in [11, 48, 58, 75] where the coarse-grid correction difficulty for geometric multigrid with DCA is fixed by different remedies, like overweighting of residuals [11], additional Krylov acceleration [48], application of algebraic multigrid [58], or special higher-order coarse-grid discretizations [75]. Many of these approaches can be well analyzed by the Fourier three-grid analysis.

The above-reported deterioration of the two- (and multi)grid convergence (4.8) due to DCA suggests the use of different discretizations on different grids in order to overcome the difficulty induced by the characteristic Fourier components. The convergence factors of the multigrid method by de Zeeuw [77] based on matrix-dependent transfer operators and GCA get worse for discretization (4.10) with an increasing number of grids, even when using a powerful relaxation like ILLU. This can be confirmed by Fourier three-grid analysis. It is possible to improve the convergence properties with another Galerkin coarse-grid operator based on transfer operators from the nonsymmetric blackbox multigrid by Dendy [18]. It is stated in [75] that the application of this Galerkin coarsening should be useful. The symmetric prolongation, based on the symmetric part of the respective discretization operator $\frac{1}{2}(L_h + L_h^*)$, is similar to the well-known operator-dependent prolongations for jumping coefficients; see, for instance, [28, 65, 77]. The nonsymmetric restriction, however, is defined as the transpose of a prolongation operator that is based on L_h^* , leading to an upstream restriction. This is particularly useful because the coarse-grid operators remain upstream as well and tend to a second-order compact upstream discretization. This can be nicely observed in Fig. 4.6 showing the resulting stencil representations for L_h , L_{2h} , and L_{4h}

and convection angle $\beta = 45^\circ$. This agrees with the observation that the coarse-grid operators must become higher-order, at least in the cross-stream direction, to provide a good coarse-grid approximation, see [75]. The related

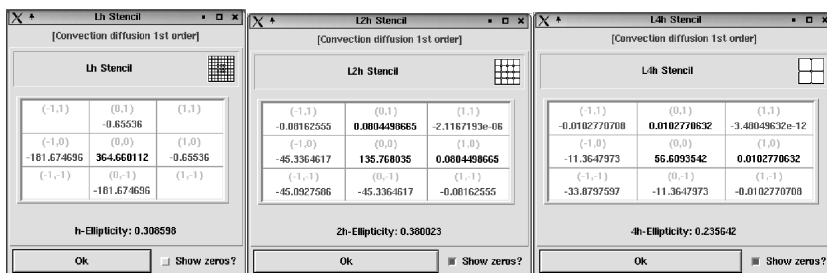


FIGURE 4.6: Sequence of stencils for the convection diffusion equation generated by Galerkin coarsening based on transfer operators due to Dendy [18]; $\varepsilon = 10^{-5}$, $\beta = 45^\circ$, $h = 1/256$.

measures of h -ellipticity indicate that for this particular angle (representing flow directions that are nonaligned with the computational grid) efficient *pointwise* smoothing is possible. However, considering the aligned case (i.e., $\beta = 0^\circ, 90^\circ, 180^\circ, 270^\circ$) one has to switch to an appropriate line relaxation (or a point smoother with a downstream numbering of grid points). More details concerning this matter are given in Section 5.8.2.

Table 4.7 shows the maximal one-, two-, and three-grid factors with the corresponding convection angles in brackets for MG7 which is composed of

-
- | | |
|------|---|
| MG7: | <ul style="list-style-type: none"> • standard coarsening and GCA • matrix-dependent prolongation and restriction, Dendy [18] • and damped ($\omega = 0.7$) AD-RB-JAC relaxation |
|------|---|
-

TABLE 4.7: Fourier values and numerically obtained asymptotic convergence factors for MG7 involving 8 levels for the rotating convection diffusion equation 4.7 discretized by a first-order upwind scheme, $\varepsilon = 10^{-5}$, $h = 1/256$

Cycle	$\rho(S^\nu(\beta)Q)$ ($\beta = 3^\circ$)	$\rho(M_{2L}(\beta))$ ($\beta = 6^\circ$)	$\rho(M_{3L}(\beta))$ ($\beta = 6^\circ$)	$\ M_{3L}(\beta)\ _d$ ($\beta = 6^\circ$)	$\rho_n(8L)$
$W(1,1)$	0.29	0.22	0.22	0.24	0.20
$W(2,1)$	0.15	0.17	0.18	0.20	0.17

As the two- and three-grid factors are very similar or even equal, it can be expected that the multigrid convergence for discretization (4.10) is close to these estimates, which is confirmed by the numerical reference for an 8-grid method. The maximal norm values $\|M_{3L}(\beta)\|_d$ differ only slightly from the corresponding $\rho(M_{3L}(\beta))$, which indicates that the convergence speed for a single iteration is not much larger than the asymptotic convergence factor. Thus, $\|M_{3L}(\beta)\|_d$ is a satisfactory prediction for the actual multigrid convergence. This is observed in the numerical convergence history. Furthermore, we see that the damped AD-RB-JAC relaxation, which is a robust smoother for the fine-grid discretization [65] keeps this property for the Galerkin coarse-grid discretizations resulting from the blackbox transfer operators. If Galerkin coarsening and matrix-dependent transfer operators are replaced by DCA and geometric transfer operators, the deterioration of the convergence factors predicted by (4.8) can be confirmed by **LFA** as well.

4.1.8 Convection diffusion equation: higher-order upwind discretization

We conclude the discussion of two-dimensional scalar equations with the higher-order upwind discretization of the convection diffusion equation (4.7) using van Leer's κ -scheme [39]. The diffusive part is discretized as in Section 4.1.7. If $a_1 u_1(\mathbf{x})$ is discretized by higher-order kappa upwinding, one obtains

$$\begin{aligned} & \frac{a_1}{2h_1} [-1 \ 0 \ 1]_{h_1} + \\ & \frac{1-\kappa}{8h_1} [|a_1| + a_1 \ -4|a_1| - 2a_1 \ 6|a_1| \ 2a_1 - 4|a_1| \ -a_1 + |a_1|]_{h_1}. \end{aligned} \quad (4.11)$$

The discretization of $a_2 u_2(\mathbf{x})$ can be found similarly. The convective part is discretized with second-order for $\kappa \in [-1, 1] \setminus \{1/3\}$ and with third-order for $\kappa = 1/3$.

Channel-type flows employing higher-order upwind discretizations are treated successfully in [49] by introducing so-called KAPPA-smoothers which are especially designed for the above κ -schemes. However, an efficient multigrid treatment for rotating flows (4.7) involving higher-order upwind discretization is a particular difficult task. A promising approach presented in [48] applies a multigrid method based on the KAPPA-smoother as a preconditioner for a Krylov subspace method.

4.2 CASE STUDIES FOR 3D SCALAR PROBLEMS

The next sequence of case studies consists of three-dimensional scalar equations. More precisely, the three-dimensional versions of the test cases from the previous section are presented. Hence, we can keep the discussion brief. We mainly focus on differences and similarities compared to the two-dimensional variants.

For three-dimensional applications, it is often convenient to decompose the complex stencil representations of the different discretizations into the central one-dimensional stencils from Examples 1.2 and 1.3.

4.2.1 Anisotropic diffusion equation: second-order discretization

A second-order accurate approximation of the anisotropic diffusion equation

$$Lu(\mathbf{x}) = -\varepsilon_1 u_{11}(\mathbf{x}) - \varepsilon_2 u_{22}(\mathbf{x}) - \varepsilon_3 u_{33}(\mathbf{x}) = f(\mathbf{x}) \quad (\varepsilon_1, \varepsilon_2, \varepsilon_3 > 0) \quad (4.12)$$

is represented by the stencil

$$\begin{aligned} L_{\mathbf{h}} &\triangleq -\varepsilon_1 [d_{\kappa_1}^{11}]_{h_1} - \varepsilon_2 [d_{\kappa_2}^{22}]_{h_2} - \varepsilon_3 [d_{\kappa_3}^{33}]_{h_3} \\ &= \left[\begin{bmatrix} -\frac{\varepsilon_3}{h_3^2} \end{bmatrix}_{\mathbf{h}} \left[-\frac{\varepsilon_1}{h_1^2} 2 \left(\frac{\varepsilon_1}{h_1^2} + \frac{\varepsilon_2}{h_2^2} + \frac{\varepsilon_3}{h_3^2} \right) - \frac{\varepsilon_1}{h_1^2} \right]_{\mathbf{h}} \begin{bmatrix} -\frac{\varepsilon_3}{h_3^2} \end{bmatrix}_{\mathbf{h}} \right]. \end{aligned} \quad (4.13)$$

By setting $\varepsilon_1 = \varepsilon_2 = \varepsilon_3 = 1$ we get the Poisson equation. A W(1,1)-cycle of MGO(point RB-JAC) applied to the three-dimensional Poisson equation with mesh size $h = 1/32$ yields an asymptotic three-grid convergence factor of $\rho(M_{3L}) = 0.192$; compare with the left part of Figure 4.7.

In [74] it was shown that an overrelaxation parameter ($\omega > 1$) improves the smoothing properties of RB-JAC (leading to ω -RB-JAC) and therefore improves the convergence of MGO(point ω -RB-JAC) compared to MGO(point RB-JAC) for d -dimensional Poisson-type equations. The extra computational work for performing the overrelaxation in the smoother is worthwhile if $d = 3$. The resulting excellent convergence factors for $\omega = 1.15$ are shown in the right part of Figure 4.7. The application of an optimal overrelaxation parameter leads to an orbital eigenvalue distribution of the three-grid error-reduction operator; compare with Section 4.1.5 where an optimal one-stage RB-JAC method applied to the biharmonic equation is discussed. The implications of optimal relaxation parameters to the eigenvalue distribution of the related multigrid method are investigated in [67, 69].

In general, the coefficients $\varepsilon_1, \varepsilon_2, \varepsilon_3$ referring to the different spatial directions may be different. As long as the coefficients are of the same magni-

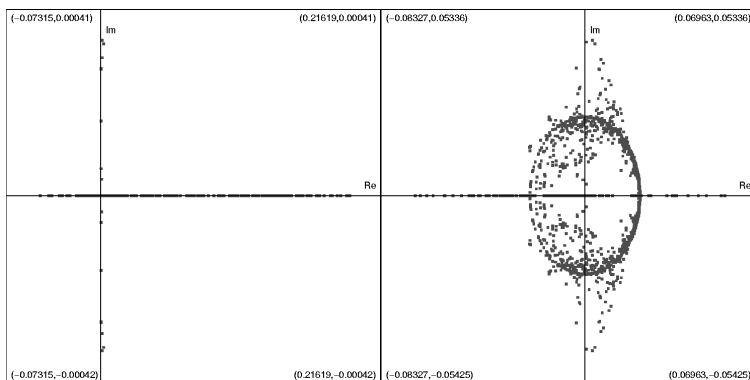
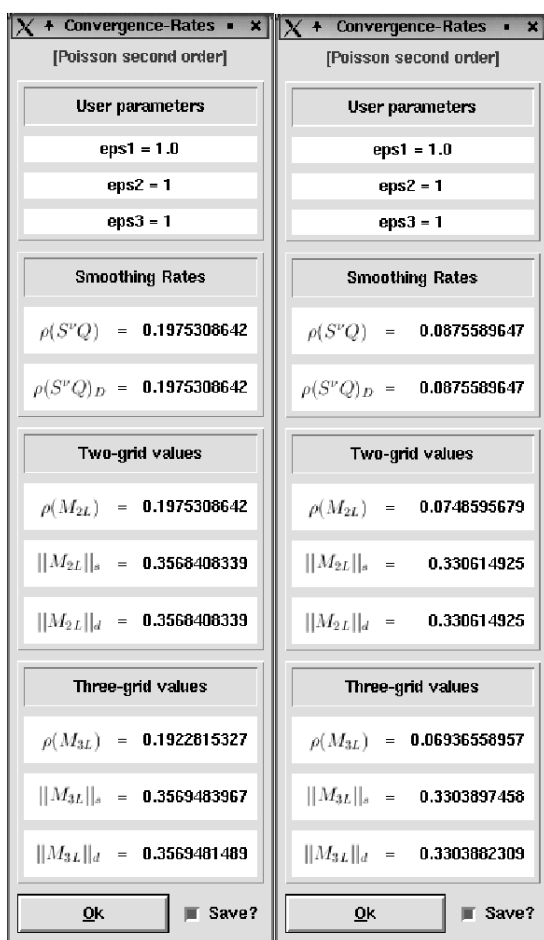


FIGURE 4.7: Influence of a relaxation parameter for MG0(point RB-JAC) and MG0(point ω -RB-JAC) applied to the three-dimensional Poisson equation.

tude (Poisson-like problem) multigrid algorithms based on standard coarsening and point relaxation are the methods of choice. If, however, the coefficients $\varepsilon_1, \varepsilon_2, \varepsilon_3$ differ considerably, more sophisticated methods have to be chosen. As in the two-dimensional case, either the coarsening strategy or the smoothing method have to be adapted. Considering three dimensions, we have an increased number of different coarsening strategies and relaxation methods; compare with the lower parts of [Figures 3.7](#) and [3.10](#). To smooth all strongly coupled unknowns simultaneously in a block, line, or even plane relaxation has to be selected. For the implementation of plane relaxations, appropriate two-dimensional multigrid methods can be applied.

In [61], concrete recommendations are given for how to choose the smoothing method or the coarsening strategy in case of considerably varying coefficients. Following [61], we discuss three representative situations. We use “ \sim ” in the sense of being similar and “ \gg ” for being much larger. In the first case, $\varepsilon_1 \gg \varepsilon_2 \sim \varepsilon_3$, x_1 -line relaxation is recommended if standard coarsening is selected. Alternatively, x_1 -semicoarsening might be used in combination with point GS-LEX smoothing. In [Figure 4.8](#), the effect of an adapted relaxation scheme (right, $\text{MGO}(x_1\text{-line GS-LEX})$) or an adapted coarsening yielding **MG8** (middle)

	<ul style="list-style-type: none"> • x_1-semicoarsening and DCA
MG8:	<ul style="list-style-type: none"> • bilinear interpolation and full weighting • point GS-LEX relaxation

compared to the naive approach (left, $\text{MGO}(\text{point GS-LEX})$) becomes obvious. The respective distributions of the eigenvalues, are shown from left to right, too.

For the second case, $\varepsilon_1 \sim \varepsilon_2 \gg \varepsilon_3$, an (x_1, x_2) -plane relaxation could be applied, where the corresponding two-dimensional multigrid method—used to perform the plane smoothing—might be based on point relaxation. Setting $\varepsilon_1 = \varepsilon_2 = 100$ and $\varepsilon_3 = 1$, we show in [Figure 4.9](#) that playing with either smoothing strategy ($\text{MGO}((x_1, x_2)\text{-plane GS-LEX})$, right) or coarsening strategy leading to **MG9** (middle)

	<ul style="list-style-type: none"> • (x_1, x_2)-semicoarsening and DCA
MG9:	<ul style="list-style-type: none"> • linear interpolation and full weighting • point GS-LEX relaxation

helps to overcome the problems which occur with the naive approach for the anisotropic case ($\text{MGO}(\text{point GS-LEX})$, left).

Regarding the third case, $\varepsilon_1 \gg \varepsilon_2 \gg \varepsilon_3$, an (x_1, x_2) -plane relaxation becomes necessary, where the related two-dimensional multigrid method to carry out the plane smoothing should use x_1 -line relaxation. A robust multigrid algorithm w.r.t. all possible choices for $\varepsilon_1, \varepsilon_2, \varepsilon_3$ is based on alternating plane

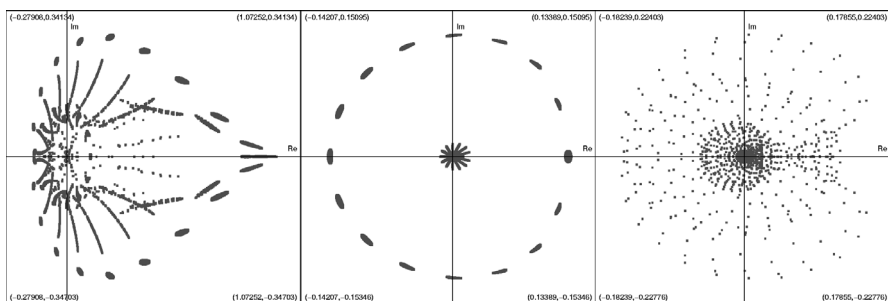
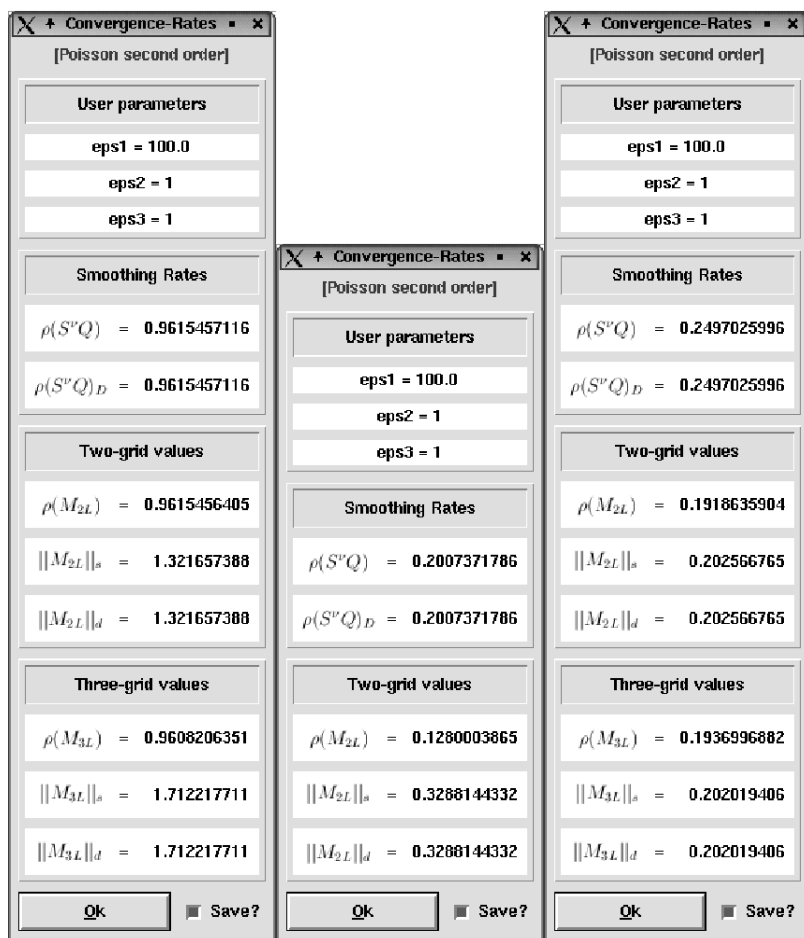


FIGURE 4.8: Local Fourier analysis estimates and eigenvalue distributions for MG0(point GS-LEX) (left), MG8 (middle), and MG0(x_1 -line GS-LEX) applied to the 3D anisotropic diffusion equation, W(1,1)-cycles; $\varepsilon_1 = 100$, $\varepsilon_2 = \varepsilon_3 = 1$.

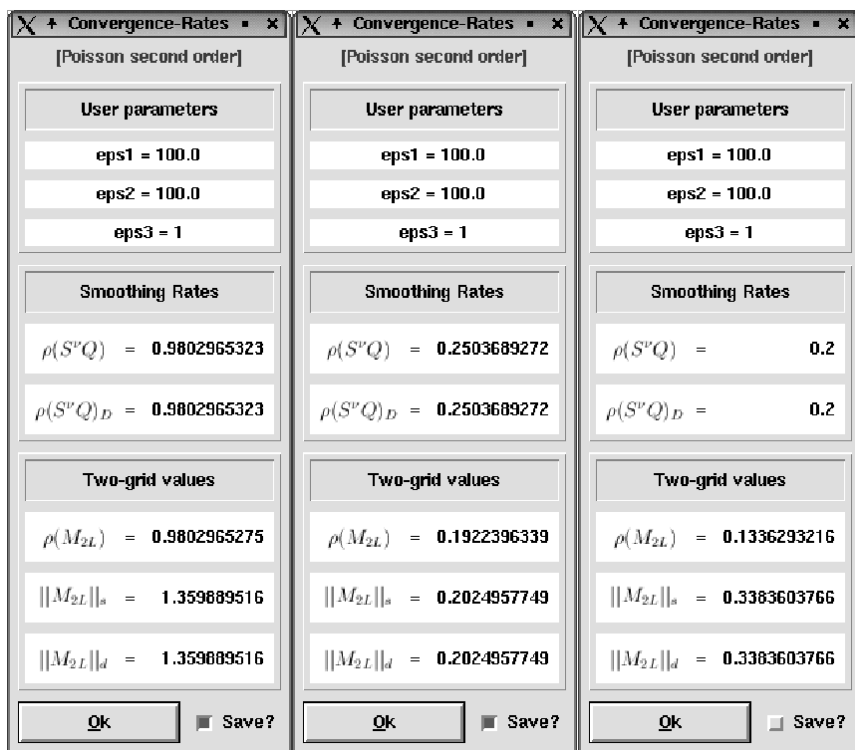


FIGURE 4.9: Local Fourier analysis estimates for MG0(point GS-LEX) (left), MG9 (middle), and MG0((x_1, x_2) -plane GS-LEX) applied to the 3D anisotropic diffusion equation, W(1,1)-cycles; $\varepsilon_1 = \varepsilon_2 = 100, \varepsilon_3 = 1$.

relaxation. The related two-dimensional multigrid methods to carry out the different plane relaxations have to apply alternating line relaxation.

Note that for moderate anisotropies it is possible to construct efficient *point* relaxation methods (in connection with standard coarsening) by a careful choice of relaxation or multistage parameters as it is shown in [67, 74]. This observation carries over to two-dimensional (moderately) anisotropic cases.

4.2.2 Anisotropic diffusion equation: fourth-order discretization

A fourth-order approximation of the anisotropic diffusion equation (4.12) is represented by the stencil

$$L_h \triangleq \left[[\ell_{\kappa}]_h^{(-2)} [\ell_{\kappa}]_h^{(-1)} [\ell_{\kappa}]_h^{(0)} [\ell_{\kappa}]_h^{(1)} [\ell_{\kappa}]_h^{(2)} \right]$$

with $[\ell_{\kappa}]_h^{(-2)} = [\ell_{\kappa}]_h^{(2)} = \frac{1}{12} \left[\frac{\varepsilon_3}{h_3^2} \right]_h$, $[\ell_{\kappa}]_h^{(-1)} = [\ell_{\kappa}]_h^{(1)} = \frac{1}{12} \left[\frac{-16\varepsilon_3}{h_3^2} \right]_h$,

$$\text{and } [\ell_{\kappa}]_h^{(0)} = \frac{1}{12} \left[\begin{array}{c} \frac{\varepsilon_2}{h_2^2} \\ \frac{-16\varepsilon_2}{h_2^2} \\ \frac{\varepsilon_1}{h_1^2} \frac{-16\varepsilon_1}{h_1^2} 30 \left(\frac{\varepsilon_1}{h_1^2} + \frac{\varepsilon_2}{h_2^2} + \frac{\varepsilon_3}{h_3^2} \right) \frac{-16\varepsilon_1}{h_1^2} \frac{\varepsilon_1}{h_1^2} \\ \frac{-16\varepsilon_2}{h_2^2} \\ \frac{\varepsilon_2}{h_2^2} \end{array} \right]_h.$$

Similar as in the two-dimensional variant it is useful to apply the second-order discretization (4.13) on coarse grids to obtain an improved V-cycle convergence. A three-dimensional variant of MG5 with finest mesh size $h = 1/32$ yields a V(1,1) three-grid convergence factor $\rho(M_{3L}) = 0.213$ compared to $\rho(M_{3L}) = 0.290$ obtained by the standard algorithm MG0(point RB-JAC).

4.2.3 Anisotropic diffusion equation: Mehrstellen discretization

The fourth-order Mehrstellen discretization

$$L_h u_h(x) = R_h f_h(x)$$

of the anisotropic diffusion equation (4.12) is described by the stencils

$$L_h \triangleq -\varepsilon_1 [d_{\kappa_1}^{11}]_{h_1} - \varepsilon_2 [d_{\kappa_2}^{22}]_{h_2} - \varepsilon_3 [d_{\kappa_3}^{33}]_{h_3} - \left(\varepsilon_1 \frac{h_2^2}{12} + \varepsilon_2 \frac{h_1^2}{12} \right) [d_{\kappa}^{1122}]_h$$

$$- \left(\varepsilon_1 \frac{h_3^2}{12} + \varepsilon_3 \frac{h_1^2}{12} \right) [d_{\kappa}^{1133}]_h - \left(\varepsilon_2 \frac{h_3^2}{12} + \varepsilon_3 \frac{h_2^2}{12} \right) [d_{\kappa}^{2233}]_h$$

$$R_h \triangleq [1]_h + \frac{h_1^2}{12} [d_{\kappa_1}^{11}]_{h_1} + \frac{h_2^2}{12} [d_{\kappa_2}^{22}]_{h_2} + \frac{h_3^2}{12} [d_{\kappa_3}^{33}]_{h_3}.$$

For $\varepsilon_1 = \varepsilon_2 = \varepsilon_3 = 1$ and $h_1 = h_2 = h_3 = h$, one obtains

$$L_h \triangleq \frac{1}{6h^2} \left[\left[\begin{array}{ccc} -1 & & \\ -1 & -2 & -1 \\ & -1 & \end{array} \right]_h \left[\begin{array}{ccc} -1 & -2 & -1 \\ -2 & 24 & -2 \\ -1 & -2 & -1 \end{array} \right]_h \left[\begin{array}{ccc} -1 & & \\ -1 & -2 & -1 \\ & -1 & \end{array} \right]_h \right] \quad (4.14)$$

$$R_h \triangleq \frac{1}{12} \left[[1]_h \left[\begin{array}{ccc} 1 & & \\ 1 & 6 & 1 \\ & 1 & \end{array} \right]_h [1]_h \right].$$

The optimal overrelaxation parameter for damped JAC relaxation applied to (4.14) is $\omega = 12/11$. The resulting asymptotic V(1,1) three-grid convergence for **MG0**(point ω -JAC) with finest mesh size $h = 1/32$ reads $\rho(M_{3L}) = 0.229$. It is interesting to note that the three-dimensional variant of **MG0**(point ω -JAC) exactly resembles the convergence behavior of the two-dimensional version discussed in Section 4.1.3. This is a particular property of optimal damped JAC relaxation applied to the Mehrstellen discretization of the Laplacian. Usually one has to expect a (slight) deterioration of the multigrid convergence with an increasing dimension. This can be observed, for example, by comparing the results for the two- and three-dimensional versions of the second-order approximation of the anisotropic diffusion equation; see Sections 4.1.1 and 4.2.1. For a detailed investigation of this phenomenon, we refer to Chapter 4 of [67].

4.2.4 Helmholtz equation

The three-dimensional variant of the Helmholtz equation reads

$$Lu(\mathbf{x}) = -u_{11}(\mathbf{x}) - u_{22}(\mathbf{x}) - u_{33}(\mathbf{x}) + cu(\mathbf{x}) = f(\mathbf{x}) \quad (c \in \mathbb{R}).$$

The corresponding central second-order approximation implemented in **LFA** is given by

$$L_h \triangleq \left[\left[-\frac{1}{h^2} \right]_h \left[-\frac{1}{h^2} \frac{6}{h^2} + c - \frac{1}{h^2} \right]_h \left[-\frac{1}{h^2} \right]_h \right].$$

4.2.5 Biharmonic equation

A second-order approximation of the biharmonic equation,

$$Lu(\mathbf{x}) = -(\partial_{11} + \partial_{22} + \partial_{33})(\partial_{11} + \partial_{22} + \partial_{33})u(\mathbf{x}) = f(\mathbf{x}),$$

is represented by the stencil

$$L_h \triangleq -[d_{\kappa_1}^{1111}]_{h_1} - [d_{\kappa_2}^{2222}]_{h_2} - [d_{\kappa_3}^{3333}]_{h_3} - 2[d_{\kappa}^{1122}]_h - 2[d_{\kappa}^{1133}]_h - 2[d_{\kappa}^{2233}]_h.$$

As in the two-dimensional case it is difficult to find an efficient smoothing method for the biharmonic equation. A promising approach is to apply a composite relaxation scheme, which is explained in detail in [42]. Alternatively, one might construct multistage RB-JAC relaxations; compare with Section 4.1.5.

4.2.6 Convection diffusion equation: first-order upwind discretization

Another well-known test problem is the convection diffusion equation

$$Lu(\mathbf{x}) = -\varepsilon(u_{11}(\mathbf{x}) + u_{22}(\mathbf{x}) + u_{33}(\mathbf{x})) + a_1 u_1(\mathbf{x}) + a_2 u_2(\mathbf{x}) + a_3 u_3(\mathbf{x}) = f(\mathbf{x}),$$

with coefficients $a_1 = \cos \alpha \cos \beta$, $a_2 = \sin \alpha \cos \beta$, and $a_3 = \sin \beta$. The operator is split into the diffusive part, represented by $L^d u(\mathbf{x}) = -\varepsilon(u_{11}(\mathbf{x}) + u_{22}(\mathbf{x}) + u_{33}(\mathbf{x}))$, and into the convective part $L^c u(\mathbf{x}) = a_1 u_1(\mathbf{x}) + a_2 u_2(\mathbf{x}) + a_3 u_3(\mathbf{x})$.

The Laplace term L^d is discretized by central differences as in Section 4.2.1. A general stencil formula with a first-order upwind scheme of $a_1 u_1(\mathbf{x})$ for $a_1 \geq 0$ as well as for $a_1 < 0$ was already given in Section 4.1.7. The discretization of $a_2 u_2(\mathbf{x})$ and $a_3 u_3(\mathbf{x})$ is carried out accordingly.

The difficulties for convection-dominated recirculating flow problems carry over from the two-dimensional case as well as the possible remedies discussed in Section 4.1.7.

4.3 CASE STUDIES FOR 2D SYSTEMS OF EQUATIONS

The last class of case studies is concerned with two-dimensional systems of equations. Local Fourier analysis is particularly valuable for systems of PDEs since it is often much more difficult to identify the correct multigrid components than in the scalar case. For example, the different behavior of collective (COL) relaxation and decoupled variants (DEC1, DEC2) can often not be predicted by simple intuition or an “extrapolation” from the scalar case.

For systems of equations implemented in the accompanying software, not only the corresponding discretization has to be specified but also the predefined distributor to be applied within a distributive relaxation scheme; compare with Section 3.5.2.

4.3.1 Biharmonic system

Recall from Section 4.1.5 that the biharmonic equation is sometimes transformed into a system of two Poisson-type equations in order to get a more convenient formulation w.r.t. a possible multigrid treatment. A second-order accurate approximation of the resulting biharmonic system

$$L\mathbf{u} = \begin{pmatrix} \Delta & 0 \\ -I & \Delta \end{pmatrix} \begin{pmatrix} u^1 \\ u^2 \end{pmatrix} = \begin{pmatrix} f^1 \\ 0 \end{pmatrix}$$

reads

$$L_h \mathbf{u}_h = \begin{pmatrix} \Delta_h & 0 \\ -I_h & \Delta_h \end{pmatrix} \begin{pmatrix} u_h^1 \\ u_h^2 \end{pmatrix} = \begin{pmatrix} f_h^1 \\ 0 \end{pmatrix}.$$

I_h denotes the discrete identity operator and the predefined distributor is given by $\mathbf{C}_h = L_h^*$. For this particular system one obtains the same smoothing factors for collective and decoupled relaxation schemes due to the vanishing

$L_h^{1,2}$ block and the simple structure of $L_h^{2,1}$. More precisely, the smoothing properties are simply governed by the diagonal blocks, i.e., the same smoothing factors as for the two-dimensional Poisson equation are found. Hence, applying a V(1,1)-cycle of MG0(point COL-RB-JAC) to the biharmonic system yields excellent multigrid convergence as demonstrated in Figure 4.10.

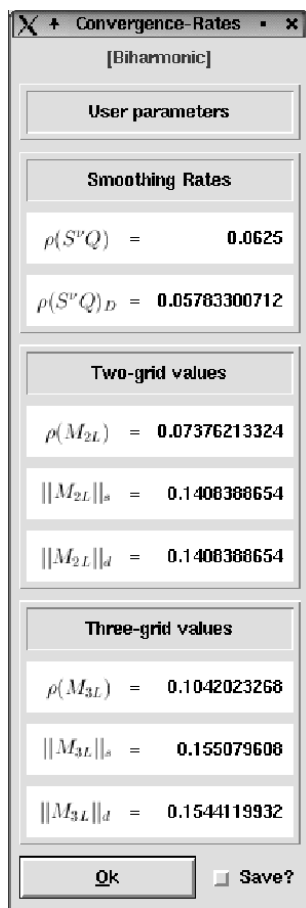


FIGURE 4.10: V(1,1)-cycle of MG0(point COL-RB-JAC) with finest mesh size $h = 1/32$ applied to the biharmonic system.

4.3.2 Stokes equations

Next, the Stokes equations are considered to point out the different behavior of collective and decoupled relaxation and the possible benefits of distributive smoothing regarding this matter. A central second-order discretization of the Stokes equations

$$\begin{aligned} L^{1,1}u + L^{1,3}p &= -(\partial_{11} + \partial_{22})u + \partial_1 p = 0, \\ L^{2,2}v + L^{2,3}p &= -(\partial_{11} + \partial_{22})v + \partial_2 p = 0, \\ L^{3,1}u + L^{3,2}v &= \partial_1 u + \partial_2 v = 0 \end{aligned}$$

is given by

$$L_h = \begin{pmatrix} -\Delta_h & 0 & D_h^1 \\ 0 & -\Delta_h & D_h^2 \\ D_h^1 & D_h^2 & -\mu h^2 \Delta_h \end{pmatrix} \quad \text{with } \mu \geq 0.$$

p , u , and v denote the pressure, the velocity component in x_1 -direction, and the velocity component in x_2 -direction, respectively. The artificial pressure term $L_h^{3,3}$ is applied to overcome the well-known checkerboard instability for nonstaggered discretizations of Navier-Stokes-type systems. A reasonable choice for the parameter μ is $\mu = 1/16$; see [62] for details.

Applying a collective AD-GS-LEX relaxation with relaxation parameter $\omega = 0.7$ yields satisfactory asymptotic convergence factors. A V(1,1)-cycle of MGO(ω -COL-AD-GS-LEX) with finest mesh size $h = 1/32$ and $\mu = 1/16$ has an asymptotic three-grid factor of $\rho(M_{3L}) = 0.268$. The corresponding decoupled variants lead to *diverging* algorithms. A possible cure for such a phenomenon concerning decoupled smoothing is the introduction of an appropriate distributor. The discrete distributor C_h , taken from [7], and the resulting transformed system read

$$C_h = \begin{pmatrix} I_h & 0 & -D_h^1 \\ 0 & I_h & -D_h^2 \\ 0 & 0 & -\Delta_h \end{pmatrix}, \quad L_h C_h = \begin{pmatrix} -\Delta_h & 0 & 0 \\ 0 & -\Delta_h & 0 \\ D_h^1 & D_h^2 & -\Delta_h + \mu h^2 \Delta_h^2 \end{pmatrix}.$$

The stencil representations of C_h and $L_h C_h$ are shown in Figure 4.11. Obviously, a triangular system results (i.e., the upper blocks of $(L_h C_h)^{i,j}$ with $j > i$ vanish) which is suited for decoupled smoothing. The distributive variant of ω -DEC2-AD-GS-LEX relaxation yields an excellent asymptotic three-grid factor of $\rho(M_{3L}) = 0.091$ for a V(1,1)-cycle of our standard multigrid algorithm MGO.

4.3.3 First-order discretization of the Oseen equations

In this subsection, we consider a *nonlinear* system with varying coefficients in order to demonstrate the applicability of local Fourier analysis even for such complicated situations. The driven cavity flow is a popular test case in

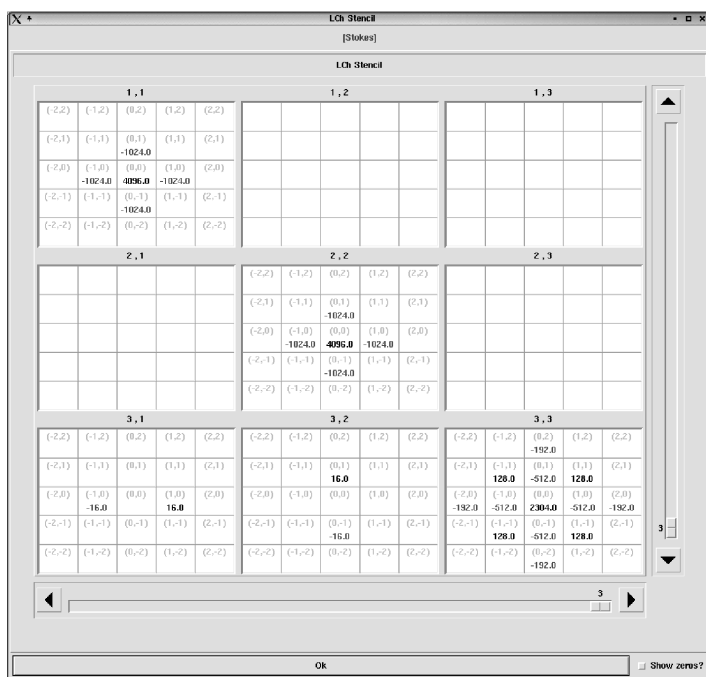
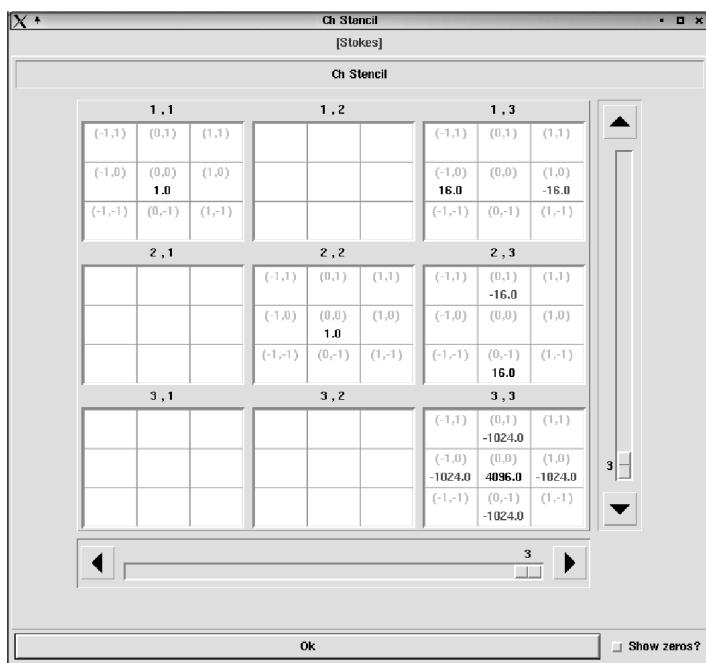


FIGURE 4.11: Stencil representations for the Stokes distributor C_h (upper part) and the transformed system $L_h C_h$ (lower part); $\mu = 1/16$, $h = 1/32$.

computational fluid dynamics. It is governed by the incompressible Navier-Stokes (NVI) equations on the unit square, i.e.,

$$-\frac{1}{Re} (u_{11}(\mathbf{x}) + u_{22}(\mathbf{x})) + (u^2(\mathbf{x}))_1 + (u(\mathbf{x})v(\mathbf{x}))_2 + p_1(\mathbf{x}) = 0, \quad (4.15)$$

$$-\frac{1}{Re} (v_{11}(\mathbf{x}) + v_{22}(\mathbf{x})) + (u(\mathbf{x})v(\mathbf{x}))_1 + (v^2(\mathbf{x}))_2 + p_2(\mathbf{x}) = 0, \quad (4.16)$$

$$u_1(\mathbf{x}) + v_2(\mathbf{x}) = 0 \quad (4.17)$$

with $\mathbf{x} \in (0,1)^2$. As boundary conditions we set $(u(\mathbf{x}), v(\mathbf{x})) = (1,0)$ at $x_2 = 1$ —representing a moving top wall—and homogeneous Dirichlet boundary conditions for $u(\mathbf{x})$ and $v(\mathbf{x})$ elsewhere at the boundary.

Re denotes the Reynolds number, p the pressure, and u and v the velocity components. (4.15) and (4.16) are called the momentum equations and (4.17) is named the continuity equation. These equations model the conservation of momentum and mass. The driven cavity problem is an example for *recirculating* flow; see Figure 4.12 showing the streamlines for the driven cavity problem with $Re = 5000$. We are interested in flow problems with a domi-

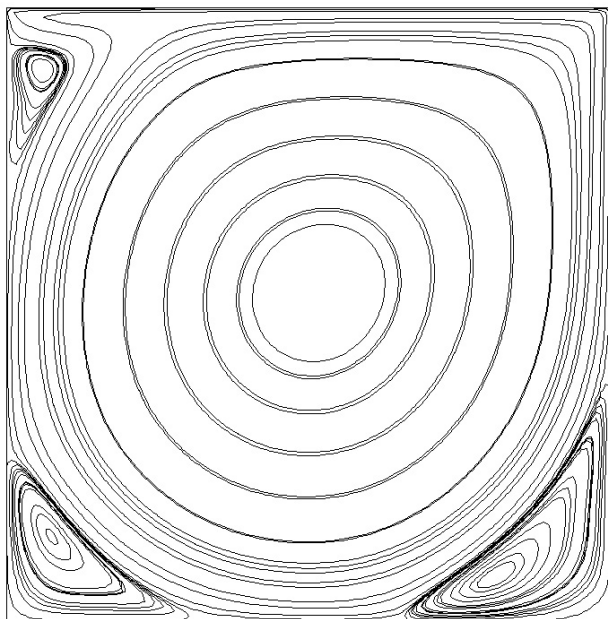


FIGURE 4.12: Streamlines for the driven cavity problem, $Re = 5000$.

nant convection, i.e., $Re \gg 1$ which are particularly difficult w.r.t. an efficient multigrid treatment as it has already been discussed in connection with the

convection diffusion equation in Section 4.1.7. Note that the above system requires only two boundary conditions. If, however, boundary conditions for u and v are specified, the pressure is only determined up to a constant. Therefore, a second-order extrapolation for the pressure at the boundary is applied. More details can be found, for example, in [62].

We use a nonstaggered discretization, i.e., all unknowns $u_h(\mathbf{x})$, $v_h(\mathbf{x})$ and $p_h(\mathbf{x})$ ($\mathbf{x} \in \Omega_h$) are located at the grid vertices of Ω_h . The diffusive parts of the momentum equations are discretized by second-order central differences. For the remaining parts of the three equations, we apply the first-order variant of Dick's flux difference splitting [19]. The momentum equations exhibit a singular perturbation similar as for the convection diffusion equation, if $1/Re$ tends to 0. The flux difference splitting discretization introduces artificial viscosity terms. Therefore, it is an appropriate choice for the calculation of flows involving high Reynolds numbers $Re \gg 1$ due to its stabilizing impact.

To allow an application of local Fourier analysis we consider a linearized form of the NVI called the Oseen equations:

$$\begin{aligned} L^{1,1}u + L^{1,3}p &= \left(-\frac{1}{Re} (\partial_{11} + \partial_{22}) + a\partial_1 + b\partial_2 \right) u + \partial_1 p = 0, \\ L^{2,2}v + L^{2,3}p &= \left(-\frac{1}{Re} (\partial_{11} + \partial_{22}) + a\partial_1 + b\partial_2 \right) v + \partial_2 p = 0, \\ L^{3,1}u + L^{3,2}v &= \partial_1 u + \partial_2 v = 0. \end{aligned}$$

The scalar operators $L^{1,2}$, $L^{2,1}$, and $L^{3,3}$ are zero. The frozen velocity coefficients can be varied with the help of a parameter β , i.e., $a = \cos(\beta)$ and $b = \sin(\beta)$.

The first-order accurate flux difference splitting discretization [19] of the Oseen equations with constant coefficients can be expressed by the following scalar discrete operators which constitute \mathbf{L}_h as calculated in [23]:

$$\begin{aligned} L_h^{1,1} &= -\frac{1}{Re} (D_h^{11} + D_h^{22}) + aD_h^1 + bD_h^2 - \frac{h}{2} \frac{a^2 + 2}{\sqrt{a^2 + 4}} D_h^{11} - \frac{h}{2} |b| D_h^{22}, \\ L_h^{1,2} &= L_h^{2,1} = 0, \\ L_h^{1,3} &= L_h^{3,1} = D_h^1 - \frac{h}{2} \frac{a}{\sqrt{a^2 + 4}} D_h^{11}, \\ L_h^{2,2} &= -\frac{1}{Re} (D_h^{11} + D_h^{22}) + aD_h^1 + bD_h^2 - \frac{h}{2} |a| D_h^{11} - \frac{h}{2} \frac{b^2 + 2}{\sqrt{b^2 + 4}} D_h^{22}, \\ L_h^{2,3} &= L_h^{3,2} = D_h^2 - \frac{h}{2} \frac{b}{\sqrt{b^2 + 4}} D_h^{22}, \\ L_h^{3,3} &= -\frac{h}{\sqrt{a^2 + 4}} D_h^{11} - \frac{h}{\sqrt{b^2 + 4}} D_h^{22}. \end{aligned}$$

It is interesting to note that $L_h^{3,3} \neq 0$ although the continuous operator $L^{3,3}$ vanishes. The flux difference splitting approach implicitly introduces an arti-

ficial pressure term which serves as a remedy for the checkerboard instability mentioned above.

The predefined distributor reads $\mathbf{C}_h = \mathbf{L}_h^*$.

To evaluate the driven cavity flow we proceed as for the convection diffusion equation: We fix the coefficients $a = \cos(\beta)$, $b = \sin(\beta)$ and repeat the Fourier analysis for the discretized Oseen equations with constant coefficients until the range of convection angles $\beta \in [0^\circ, 360^\circ]$ is processed. For convection-dominated flow problems, i.e., $Re \gg 1$, one observes the same coarse-grid correction difficulties as described in Section 4.1.7 in connection with the convection diffusion equation. An application of our standard multigrid algorithm **MG0** with collective symmetric alternating line GS-LEX (COL-SYM-AD-GS-LEX) with relaxation parameter $\omega = 0.9$ (that is, alternating line GS-LEX with a forward ordering of grid points followed by alternating line GS-LEX with a backward ordering) yields excellent smoothing factors as it is demonstrated in Table 4.8 for $Re = 5000$. However, due to a wrong

TABLE 4.8: Fourier values and numerically obtained asymptotic convergence factors for **MG0**(ω -COL-SYM-AD-GS-LEX) involving 7 levels for the driven cavity flow discretized by first-order flux difference splitting, $Re = 5000$, $h = 1/128$

Cycle	$\rho(S^\nu(\beta)Q)$ ($\beta = 0^\circ$)	$\rho(M_{2L}(\beta))$ ($\beta = 45^\circ$)	$\rho(M_{3L}(\beta))$ ($\beta = 45^\circ$)	$\ M_{3L}(\beta)\ _d$ ($\beta = 45^\circ$)	$\rho_n(7L)$
$W(1,1)$	0.021	0.449	0.509	0.525	0.500

scaling of the coarse-grid discretization operators (DCA) a severe deterioration of the two- and three-grid factors is observed. Note that the convergence prediction provided by local Fourier analysis is in good agreement with the numerical test calculation involving seven levels. This is a remarkable result for such a difficult application. In the limit of small mesh size, i.e., $h \rightarrow 0$, one obtains the same limiting factor $1/2$ for the asymptotic two-grid convergence as for the first-order upwind discretization of the convection diffusion equation; compare with (4.8). More details concerning this matter are given in Section 7.2 in connection with the simplified Fourier k -grid analysis.

In an analogy to the scalar case, it is possible to overcome this difficulty by applying GCA based on the operator-dependent transfer operators introduced by Dendy [18] for scalar equations. The generalization of this approach to systems of equations has been outlined in Section 3.4.6. The modified version of **MG7** for systems of equations based on ω -COL-AD-RB-JAC relaxation leads to very fast multigrid convergence. The worst three-grid factor found for $\beta = 96^\circ$ reads $\rho(M_{3L}(\beta)) = 0.241$. The stencil representations of \mathbf{L}_h and

L_{4h} are shown in Figure 4.13 for $Re = 5000$, $\beta = 45^\circ$, and $h = 1/128$. It can be observed that the discretizations of the convective parts tend to second-order compact upstream discretizations—similar as for the convection diffusion equation—in contrast to the first-order upwind discretization on the fine grid.

Other possibilities to deal with the coarse-grid correction difficulty—like an acceleration with a Krylov subspace method (compare with Section 7.4) or the application of higher-order upwind discretizations on coarser grids—are discussed in [67].

4.3.4 Higher-order discretization of the Oseen equations

Employing the κ -scheme [39], we obtain a higher-order flux difference splitting discretization of the Oseen equations with constant coefficients. The representation of L_h in terms of central differences (see Example 1.2) is calculated in [66]:

$$\begin{aligned}
 L_h^{1,1} &= -\frac{1}{Re} (D_h^{11} + D_h^{22}) + aD_h^1 + bD_h^2 - \frac{1-\kappa}{4} (h^2 a D_h^{111} + h^2 b D_h^{222}) \\
 &\quad + \frac{1-\kappa}{8} \left(h^3 \frac{a^2+2}{\sqrt{a^2+4}} D_h^{1111} + h^3 |b| D_h^{2222} \right), \\
 L_h^{1,2} &= L_h^{2,1} = 0, \\
 L_h^{1,3} &= L_h^{3,1} = D_h^1 - \frac{1-\kappa}{4} h^2 a D_h^{111} + \frac{1-\kappa}{8} h^3 \frac{a}{\sqrt{a^2+4}} D_h^{1111}, \\
 L_h^{2,2} &= -\frac{1}{Re} (D_h^{11} + D_h^{22}) + aD_h^1 + bD_h^2 - \frac{1-\kappa}{4} (h^2 a D_h^{111} + h^2 b D_h^{222}) \\
 &\quad + \frac{1-\kappa}{8} \left(h^3 |a| D_h^{1111} + h^3 \frac{b^2+2}{\sqrt{b^2+4}} D_h^{2222} \right), \\
 L_h^{2,3} &= L_h^{3,2} = D_h^2 - \frac{1-\kappa}{4} h^2 b D_h^{222} + \frac{1-\kappa}{8} h^3 \frac{b}{\sqrt{b^2+4}} D_h^{2222}, \\
 L_h^{3,3} &= \frac{1-\kappa}{8} \left(h^3 \frac{2}{\sqrt{a^2+4}} D_h^{1111} + h^3 \frac{2}{\sqrt{b^2+4}} D_h^{2222} \right).
 \end{aligned}$$

Again, the predefined distributor is given by $C_h = L_h^*$. The discretization of the convective part is of second-order for $\kappa \in [-1, 1] \setminus \{1/3\}$ and of third-order for $\kappa = 1/3$. The KAPPA-relaxation proposed in [49] is an excellent smoothing method for higher-order upwind discretization of Navier-Stokes-type systems involving van Leer's κ -scheme [39]. However, due to similar coarse-grid correction difficulties as discussed above it is very difficult to find an efficient multigrid method especially for recirculating flow problems. More details about possible multigrid approaches for this particular problem can be found, for example, in [48, 67].

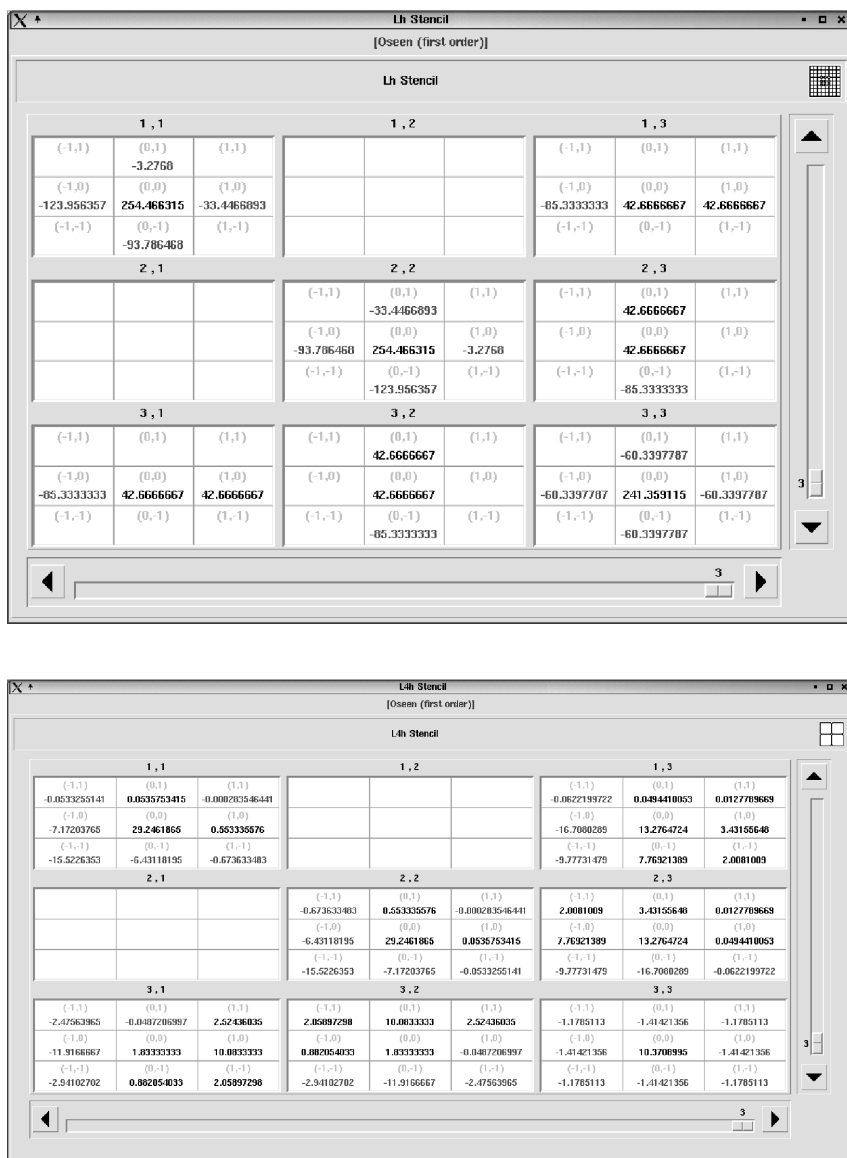


FIGURE 4.13: L_h and L_{4h} for the first-order flux difference splitting discretization of the Oseen equations; $Re = 5000$, $\beta = 45^\circ$, $h = 1/128$.

4.3.5 Elasticity system

A second-order accurate approximation of the linear elasticity problem

$$\begin{aligned} L^{1,1}u + L^{2,1}v &= -((2\mu + \lambda)\partial_{11} + \mu\partial_{22})u - (\mu + \lambda)\partial_{12}v = f \\ L^{2,1}u + L^{2,2}v &= -(\mu + \lambda)\partial_{12}u - (\mu\partial_{11} + (2\mu + \lambda)\partial_{22})v = g \end{aligned}$$

with Lamé coefficients μ and λ reads

$$\begin{aligned} L_{\mathbf{h}}^{1,1} &= -(2\mu + \lambda)D_{h_1}^{11} - \mu D_{h_2}^{22}, \\ L_{\mathbf{h}}^{1,2} &= L_{\mathbf{h}}^{2,1} = -(\mu + \lambda)D_{\mathbf{h}}^{12}, \\ L_{\mathbf{h}}^{2,2} &= -\mu D_{h_1}^{11} - (2\mu + \lambda)D_{h_2}^{22}. \end{aligned} \tag{4.18}$$

The implemented distributor is given by $\mathbf{C}_{\mathbf{h}} = \mathbf{L}_{\mathbf{h}}^*$.

The diagonal blocks $L_{\mathbf{h}}^{1,1}$ and $L_{\mathbf{h}}^{2,2}$ become anisotropic for $\lambda \gg \mu$. That is, the size of the coefficients referring to the different spatial directions (i.e., $-(2\mu + \lambda)$ and $-\mu$) varies considerably for such a choice of the Lamé coefficients. Hence, a robust smoothing method consists of alternating line relaxations; compare with Section 4.1.1 dedicated to anisotropic diffusion problems. Moreover, $\lambda \gg \mu$ leads to coarse-grid correction difficulties for standard multigrid components. This particular parameter selection refers to materials that are almost incompressible, i.e., the corresponding Poisson ratio

$$\bar{\nu} := \frac{\lambda}{2\mu + 2\lambda}$$

tends to $1/2$. It is well known that the efficiency of many solution methods for problems from linear elasticity deteriorates for $\bar{\nu} \rightarrow 1/2$. A fast converging but somewhat exotic multigrid algorithm (involving CGA, residual weighting and overrelaxation parameters) is presented in [50] which is based on insights from Fourier three-grid analysis.

4.3.6 A linear shell problem

A system of PDEs (see, for example, Section 8.5 from [62]) derived by linear shell theory, is

$$\begin{aligned} L^{1,1}f + L^{1,4}v &= (\partial_{11} + \partial_{22})f - v = 0, \\ L^{2,1}f + L^{2,2}u &= -\Lambda^2(z_{yy}\partial_{11} - 2z_{xy}\partial_{12} + z_{xx}\partial_{22})f + (\partial_{11} + \partial_{22})u = p, \\ L^{3,2}u + L^{3,3}w &= -u + (\partial_{11}\partial_{22})w = 0, \\ L^{4,3}w + L^{4,4}v &= \Lambda^2(z_{yy}\partial_{11} - 2z_{xy}\partial_{12} + z_{xx}\partial_{22})w + (\partial_{11} + \partial_{22})v = 0 \end{aligned}$$

for the four unknown functions f , u , w , and v . A central second-order accurate discretization is then given by

$$\begin{aligned} L_{\mathbf{h}}^{1,1} &= L_{\mathbf{h}}^{2,2} = L_{\mathbf{h}}^{3,3} = L_{\mathbf{h}}^{4,4} = D_{h_1}^{11} + D_{h_2}^{22}, \\ L_{\mathbf{h}}^{1,2} &= L_{\mathbf{h}}^{1,3} = L_{\mathbf{h}}^{2,3} = L_{\mathbf{h}}^{2,4} = L_{\mathbf{h}}^{3,1} = L_{\mathbf{h}}^{3,4} = L_{\mathbf{h}}^{4,1} = L_{\mathbf{h}}^{4,2} = 0, \\ L_{\mathbf{h}}^{1,4} &= L_{\mathbf{h}}^{3,2} = -I_{\mathbf{h}}, \\ L_{\mathbf{h}}^{2,1} &= -\Lambda^2 (z_{yy} D_{h_1}^{11} - 2z_{xy} D_{\mathbf{h}}^{12} + z_{xx} D_{h_2}^{22}), \\ L_{\mathbf{h}}^{4,3} &= \Lambda^2 (z_{yy} D_{h_1}^{11} - 2z_{xy} D_{\mathbf{h}}^{12} + z_{xx} D_{h_2}^{22}). \end{aligned}$$

The predefined distributor reads $\mathbf{C}_{\mathbf{h}} = \mathbf{L}_{\mathbf{h}}^*$.

Possibilities for an efficient multigrid treatment of this linear shell problem are investigated in [62] with a special emphasis on the distinction between collective and decoupled smoothing. The presented examples might be examined with the help of **LFA**. Note, that we resorted the equations compared to [62].

4.4 CREATING NEW APPLICATIONS

The collection of problems provided by the distributed software can be expanded by the user. To do so, the functions of the main-window shown in Figure 1.5 should be used.

There is a sample directory for a new application to be specified by the user. Of course one could also create a new directory and copy the files of an existing application into the new problem directory. The first step is to implement the stencil of a new discretization in the Fortran subroutine **star.f** by specifying the stencil array

- $\text{Lh}(-\text{lhmax}:\text{lhmax}, -\text{lhmax}:\text{lhmax})$ (2D_scalar)
- $\text{Lh}(-\text{lhmax}:\text{lhmax}, -\text{lhmax}:\text{lhmax}, -\text{lhmax}:\text{lhmax})$ (3D_scalar)
- $\text{Lh}(\text{Nq}, \text{Nq}, -\text{lhmax}:\text{lhmax}, -\text{lhmax}:\text{lhmax})$ (2D_systems).

It is necessary to adjust the length **lhmax** of the stencil. The default value of **lhmax** is 1 for compact 9- or 27-point stencils in two or three dimensions, respectively. **Nq** refers to the number of equations constituting the system under consideration. For example, for the biharmonic system we have **Nq** = 2 and for the Stokes equations **Nq** = 3 holds.

As an example we consider the two-dimensional anisotropic diffusion equation with possibly different mesh sizes **h**(1) and **h**(2) into x_1 - and x_2 -direction, respectively. The related source code appears as follows:

```
Lh( 0, 1) = -1.0d0/(h(2)*h(2))
Lh( 1, 0) = -1.0d0/(h(1)*h(1))*eps
Lh( 0, 0) =  2.0d0/(h(1)*h(1))*eps + 2.0d0/(h(2)*h(2))
Lh(-1, 0) = -1.0d0/(h(1)*h(1))*eps
Lh( 0,-1) = -1.0d0/(h(2)*h(2))
```

The strength of the anisotropy is governed by the user parameter **eps**. Often, there are problem specific parameters which have to be transferred to the Fourier analysis program. A typical example is the coefficient **eps** of the anisotropic diffusion equation shown above. It is possible to rename these user-defined symbolic parameters or to create more of them. Creating new discretization stencils may be simplified by using these symbolic names. They are stored for each application in a file called **user.pro**. A corresponding **Problem** menu-item allows an easy editing of the user parameters.

Considering systems of equations characterized by $q \cdot q$ scalar stencils $[\ell_{\kappa}^{i,j}]_h$ ($i, j = 1, \dots, q$) (compare with Section 1.1.4), $[\ell_{\kappa}^{i,j}]_h$ has to be specified in **Lh(i,j,-lhmax:lhmax,-lhmax:lhmax)**. For systems of equations it is furthermore possible to define an appropriate distributor. This can be done with the help of the Fortran subroutine **dist.f** by specifying the corresponding array **Ch(Nq,Nq,-lhmax:lhmax,-lhmax:lhmax)**. By default, the discrete distributor is simply given by $C_h = L_h^*$ yielding Kaczmarz relaxation schemes.

If there is some reason to apply a particular coarse-grid discretization (different from that given in **star.f**) the coarse-grid stencil might be specified in the subroutine **starH.f**. This allows the study of different discretizations on the coarse grids as it was done in Section 4.1.2.

As soon as the stencil files **star.f** and **starH.f** (and the distributor file **dist.f** considering systems of equations) have been changed they have to be compiled to create the new executable of **LFA** within the problem path. It is recommended to apply the **Problem** menu and its **systems**-option to store files in the proper directory and to adapt the makefiles for compilation of newly created files. A compilation is always required if a new application has been incorporated, if existing stencil or distributor files have been modified, or if the names of user parameters have been changed.

Part II

The Theory behind LFA

Chapter 5

FOURIER ONE-GRID OR SMOOTHING ANALYSIS

In this chapter, we concentrate on the smoothing procedure S_h in a multigrid cycle. The influence of the coarse-grid correction is neglected or, more precisely, an “ideal” coarse-grid correction operator (see below) is assumed. This smoothing analysis might also be named one-grid analysis as it only takes the fine-grid operators into account.

The main aim of this chapter is to provide the technical framework of Fourier smoothing analysis for all kinds of relaxation methods that are implemented in the accompanying software; compare with [Figure 3.10](#). According to this intention, we choose a rather general and elaborate formulation for d -dimensional problems. In particular, a detailed description of the smoothing analysis for *three-dimensional* pattern relaxations (Section 5.4) and *systems* of equations (Section 5.5) is given which is lacking in textbooks. Although this can be considered as a straightforward generalization of the two-dimensional scalar case, it is technically very complicated. Therefore it seems worthwhile to close this gap which is a key feature of this chapter.

Many examples are given to make the mathematical formulation more transparent. We would like to point out that we mainly apply *simple* examples—like the Laplacian—which can often be dealt with analytically, because they are particularly illustrative. However, for slightly more difficult problems an analytical treatment is usually not possible or at least very cumbersome. For such situations it is convenient to apply the accompanying software to perform the smoothing analysis, as it has been demonstrated in [Chapter 4](#) by a large number of test cases.

First of all, some basic facts about local Fourier analysis are provided in Section 5.1. They will be extensively used in the following. Moreover, they serve as a preparation for the more complicated Fourier k -grid analysis presented in [Chapter 6](#). We concentrate on three classes of smoothing methods:

- JAC: Jacobi-type relaxations (Section 5.3)
- GS-LEX: lexicographical Gauss-Seidel relaxations (Section 5.3)
- RB-JAC, RB-GS: pattern relaxations of red-black Jacobi- and red-black Gauss-Seidel-type (Section 5.4)

including point- and blockwise (for example, lines or planes) variants. Note that RB-JAC line or plane relaxation is often called ZEBRA relaxation. Each method may be used with a damping parameter or with multistage parameters. The damped versions are sometimes denoted by ω -JAC, ω -GS-LEX, and ω -RB-JAC referring to the damping or (over)relaxation parameter ω ; compare with Section 1.2. The analysis and development of optimal multistage relaxations is described in Section 5.6.

Section 5.8 is dedicated to the concept of h -ellipticity [6, 7, 10]. In this context, the “measure of h -ellipticity” is computed for several discrete operators from Chapter 4 giving a first indication whether a multigrid treatment is supposed to be successful.

Other systematic introductions into Fourier smoothing analysis are presented in [60, 62, 65].

5.1 ELEMENTS OF LOCAL FOURIER ANALYSIS

Starting from a linear boundary value problem

$$L_h u_h(\mathbf{x}) = f_h(\mathbf{x}) \text{ on } \Omega_h, \quad B_h u_h(\mathbf{x}) = g_h(\mathbf{x}) \text{ at } \partial\Omega_h,$$

local Fourier analysis for multigrid is based on the following simplifications:

- L_h has constant coefficients.
- Neglect boundary conditions and extend the discrete problem to an infinite grid, i.e.,

$$L_h u_h(\mathbf{x}) = f_h(\mathbf{x}) \text{ on } G_h. \quad (5.1)$$

5.1.1 Basic definitions

For a fixed grid point $\mathbf{x} \in G_h$, (5.1) reads in stencil notation (see Section 1.1.3) as

$$L_h u_h(\mathbf{x}) = \sum_{\boldsymbol{\kappa} \in J} \ell_{\boldsymbol{\kappa}} u_h(\mathbf{x} + \boldsymbol{\kappa}h) = f_h(\mathbf{x}) \quad (5.2)$$

with constant stencil coefficients $\ell_{\boldsymbol{\kappa}} \in \mathbb{R}$ (i.e., $\ell_{\boldsymbol{\kappa}}$ does not depend on \mathbf{x}) and a certain finite subset $J \subset \mathbb{Z}^d$ containing $(0, \dots, 0)$.

Definition 5.1 (Fourier component, Fourier frequency, Fourier symbol). From (5.2) it can be easily seen that the eigenfunctions of a constant coefficient infinite grid operator are given by

$$\varphi_h(\boldsymbol{\theta}, \mathbf{x}) := \prod_{j=1}^d \exp(i\theta_j x_j / h_j) \quad \text{with } \boldsymbol{\theta} \in \mathbb{R}^d, \mathbf{x} \in G_h,$$

and complex unit $i = \sqrt{-1}$. These eigenfunctions are called the *Fourier components* associated with a *Fourier frequency* θ . The corresponding eigenvalues or *Fourier symbols* of L_h read

$$\tilde{L}_h(\theta) := \sum_{\kappa \in J} \ell_{\kappa} \exp(i \theta \kappa).$$

▲

Example 5.1 (Fourier symbol of the Laplacian). The Fourier symbol of the two-dimensional Laplacian

$$L_h = -\Delta_h \triangleq \frac{1}{h^2} \begin{bmatrix} & -1 & \\ -1 & 4 & -1 \\ & -1 & \end{bmatrix}_h$$

is given by

$$\begin{aligned} \tilde{L}_h(\theta) &= \frac{1}{h^2} (4 - \exp(i\theta_1) - \exp(-i\theta_1) - \exp(i\theta_2) - \exp(-i\theta_2)) \\ &= \frac{1}{h^2} (4 - 2\cos(\theta_1) - 2\cos(\theta_2)). \end{aligned}$$

Similarly, we obtain for the d -dimensional Laplacian

$$\tilde{L}_h(\theta) = \frac{1}{h^2} \left(2d - \sum_{j=1}^d 2\cos(\theta_j) \right).$$

◀

Example 5.2 (Fourier symbols for central discretizations). The Fourier symbols for the central discretizations from Example 1.2 and Example 1.3 are given by

$$\begin{aligned} \tilde{D}_h^j(\theta) &= \frac{i}{h_j} \sin(\theta_j), \quad \tilde{D}_h^{jjj}(\theta) = \frac{i}{h_j^3} (-2\sin(\theta_j) + \sin(2\theta_j)), \\ \tilde{D}_h^{jj}(\theta) &= \frac{2}{h_j^2} (\cos(\theta_j) - 1), \quad \tilde{D}_h^{jjjj}(\theta) = \frac{2}{h_j^4} (3 - 4\cos(\theta_j) + \cos(2\theta_j)) \\ \text{and } \tilde{D}_h^{12}(\theta) &= -\frac{1}{h_1 h_2} \sin(\theta_1) \sin(\theta_2). \end{aligned}$$

◀

On G_h , we introduce the scaled Euclidean inner product

$$\langle v_h, w_h \rangle_{G_h} := \lim_{m \rightarrow \infty} \frac{1}{m^d} \sum_{|\kappa| \leq m} v_h(\kappa h) \overline{w_h(\kappa h)} \quad (5.3)$$

with $|\kappa| := \max\{|\kappa_1|, \dots, |\kappa_d|\}$ and $v_h, w_h : G_h \longrightarrow \mathbb{C}$, leading to a norm $\|v_h\|_{G_h} := \sqrt{\langle v_h, v_h \rangle_{G_h}}$. Note that the Fourier components are unitary with respect to this inner product. We define the space of bounded infinite grid functions by

$$\mathcal{F}(G_h) := \{v_h \mid v_h(\cdot) : G_h \longrightarrow \mathbb{C} \text{ with } \|v_h\|_{G_h} < \infty\}.$$

For each $v_h \in \mathcal{F}(G_h)$, there exists a Fourier transformation, that is, each v_h can be written as a linear combination of Fourier components [9, 29, 65]. In particular an error or residual function can be expanded into a series with respect to the Fourier components.

Definition 5.2 (Fourier space). Fourier components with

$$|\hat{\theta}| := \max\{|\hat{\theta}_1|, \dots, |\hat{\theta}_d|\} \geq \pi$$

are not visible on G_h , since they coincide with components $\varphi_h(\theta, \cdot)$, where $\theta = \hat{\theta} \pmod{2\pi}$, due to the periodicity of the exponential function. Therefore, the *Fourier space*

$$\mathcal{F} := \text{span} \{ \varphi_h(\theta, \cdot) : \theta \in \Theta := (-\pi, \pi]^d \} \quad (5.4)$$

contains any bounded infinite grid function. ▲

5.1.2 Generalization to systems of PDEs

The generalization to systems of q equations (1.11) is straightforward. Here, we are dealing with *vector-valued* Fourier components

$$\varphi_h(\theta, \cdot) := \varphi(\theta, \cdot) \cdot \mathbf{I} \quad \mathbf{I} = (1, \dots, 1)^T \in \mathbb{R}^q$$

and the Fourier symbols

$$\tilde{L}_h(\theta) = \begin{pmatrix} \tilde{L}_h^{1,1}(\theta) & \dots & \tilde{L}_h^{1,q}(\theta) \\ \vdots & \dots & \vdots \\ \tilde{L}_h^{q,1}(\theta) & \dots & \tilde{L}_h^{q,q}(\theta) \end{pmatrix}$$

are given by $(q \times q)$ -matrices consisting of scalar Fourier symbols.

Example 5.3 (Fourier symbol for the biharmonic system). The Fourier symbol for the second-order approximation of the d -dimensional biharmonic system

$$L_h = \begin{pmatrix} \Delta_h & 0 \\ -I_h & \Delta_h \end{pmatrix}$$

reads

$$\tilde{L}_h(\theta) = \begin{pmatrix} \frac{1}{h^2} \left(\sum_{j=1}^d 2 \cos(\theta_j) - 2d \right) & 0 \\ -1 & \frac{1}{h^2} \left(\sum_{j=1}^d 2 \cos(\theta_j) - 2d \right) \end{pmatrix}.$$

◀

The Euclidean inner product for vector-valued infinite grid functions $\mathbf{v}_h = (v_h^1, \dots, v_h^q)^T$, $\mathbf{w}_h = (w_h^1, \dots, w_h^q)^T$ has to be adapted accordingly:

$$\langle \mathbf{v}_h, \mathbf{w}_h \rangle_{G_h}^q := \sum_{j=1}^q \langle v_h^j, w_h^j \rangle_{G_h},$$

compare with (5.3).

5.2 HIGH AND LOW FOURIER FREQUENCIES

Fourier smoothing analysis is based on a distinction between “high” and “low” Fourier frequencies governed by the coarsening strategy under consideration.

5.2.1 Standard and semicoarsening

If standard or full coarsening ($H = 2h$) is selected, the Fourier components $\varphi_h(\boldsymbol{\theta}, \cdot)$ with $|\boldsymbol{\theta}| := \max\{|\theta_1|, \dots, |\theta_d|\} \leq \pi/2$ are also visible on the coarse grid G_H whereas components with $|\boldsymbol{\theta}| > \pi/2$ coincide with certain $\varphi_h(\hat{\boldsymbol{\theta}}, \cdot)$ where $|\hat{\boldsymbol{\theta}}| \leq \pi/2$. This observation leads to the following.

Definition 5.3 (High and low frequencies, standard coarsening). An element θ_j ($j \in \{1, \dots, d\}$) of a Fourier frequency $\boldsymbol{\theta}$ is called *low* (or *smooth*) if

$$-\pi/2 < \theta_j \leq \pi/2 \quad \text{for} \quad \boldsymbol{\theta} \in \Theta = (-\pi, \pi]^d.$$

Otherwise it is called *high* (or *rough*). We speak of a low (or smooth) Fourier frequency $\boldsymbol{\theta} \in \Theta$ if all its elements are low (or smooth). Otherwise it is named a high (or rough) frequency. We collect the low and high frequencies in the subsets $\Theta_{\text{low}} = (-\pi/2, \pi/2]^d$ and $\Theta_{\text{high}} = \Theta \setminus \Theta_{\text{low}}$, respectively. \blacktriangle

A two-dimensional example of low and high frequencies for standard coarsening is given in the left picture of [Figure 5.1](#).

The distinction obviously depends on the coarsening as for different coarsening strategies different sets of Fourier frequencies are visible on the coarse grid. We define the set of coordinate indices by $\mathcal{I} := \{1, \dots, d\}$. If the grid is coarsened only in a subset $\{x_j \mid j \in \mathcal{I}^c \subset \mathcal{I}\}$ of the coordinate directions and remains fixed in the other coordinates x_j with $j \in \mathcal{I}^f = \mathcal{I} \setminus \mathcal{I}^c$, we speak of semi- or partial coarsening. Several examples of semicoarsening in two and three dimensions are illustrated in [Figure 3.7](#). Considering semicoarsening, we have for the coarse-grid mesh size \mathbf{H} that $H_j = 2h_j$ for $j \in \mathcal{I}^c$ and $H_j = h_j$

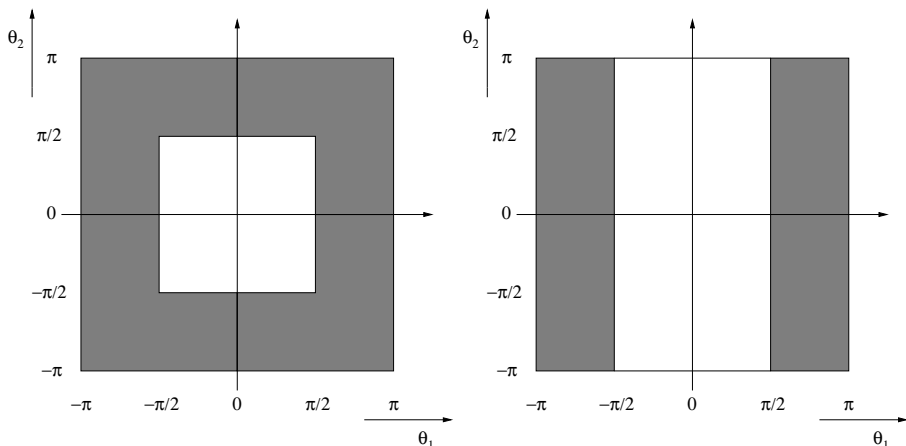


FIGURE 5.1: Low (white region) and high (shaded region) frequencies for standard (left) and x_1 -semicoarsening (right).

for $j \in \mathcal{I}^f$. The superscripts “c” and “f” refer to “coarse” and “fine,” indicating in which space directions the grid is getting *coarser* and where it remains *fine* during the coarsening procedure. Then, the definition of high and low frequencies has to be adapted accordingly.

Definition 5.4 (High and low frequencies, semicoarsening). A Fourier frequency $\theta \in \Theta$ is called *low* (or *smooth*) if all elements $\theta_j \in \mathcal{I}^c$ are low (or smooth), compare with Definition 5.3. Otherwise it is called a *high* (or *rough*) frequency. Again, the low and high frequencies constitute the subsets Θ_{low} and Θ_{high} , respectively. \blacktriangle

Figure 5.1 (right picture) illustrates the distinction of low and high frequencies in the case of x_1 -semicoarsening for a two-dimensional problem, i.e., $\mathcal{I}^c = \{1\}$ and $\mathcal{I}^f = \{2\}$.

5.2.2 Red-black coarsening and quadrupling

Another famous coarsening strategy is red-black coarsening. Here, the coarse grid is constructed by leaving out every other fine-grid point. More precisely, the fine grid is subdivided into red (R) and black (B) points in a checkerboard manner,

$$G_h^R = \{\mathbf{x} = \kappa h \in G_h \mid \sum_{j=1}^d \kappa_j \text{ even}\}, \quad G_h^B = \{\mathbf{x} = \kappa h \in G_h \mid \sum_{j=1}^d \kappa_j \text{ odd}\},$$

and the coarse grid is build by the black points only, i.e., $G_H = G_h^B$. G_H can be considered as a rotated uniform grid with mesh size $H = \sqrt{2}h$. The

corresponding set of low-frequency components for $d = 2$ reads

$$\Theta_{\text{low}} = \{\boldsymbol{\theta} \in \Theta \mid |\boldsymbol{\theta}| \leq \pi\}.$$

Quadrupling means that the coarse grid is given by $G_H = G_{4h}$. Then, the low-frequency components are simply defined by

$$\Theta_{\text{low}} = (-\pi/4, \pi/4]^d$$

in an analogy to standard coarsening ($G_H = G_{2h}$). In practice, quadrupling is rarely applied because it is difficult to find efficient smoothing methods for this coarsening strategy. This is due to the fact that the relaxation method has to take care of too many high-frequency error components.

Fourier smoothing analysis for both coarsening strategies is implemented in the accompanying software for two-dimensional applications. The related distinction in high and low Fourier frequencies is shown in Figure 5.2.

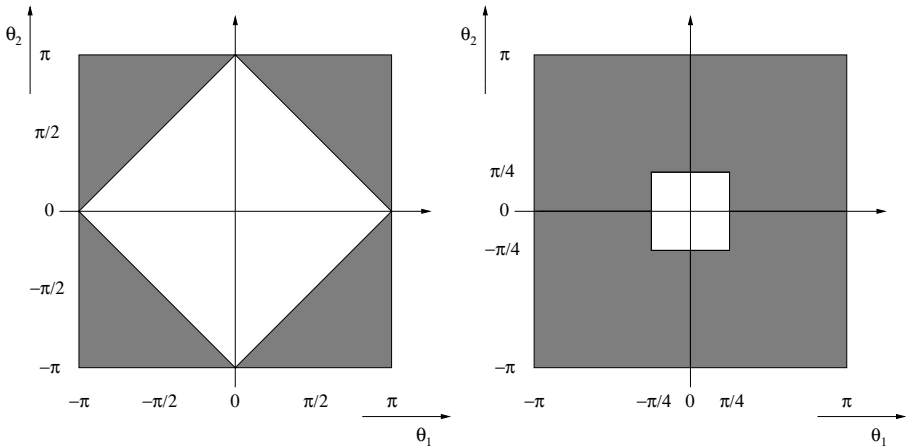


FIGURE 5.2: Low (white region) and high (shaded region) frequencies for red-black coarsening (left) and quadrupling (right).

5.3 SIMPLE RELAXATION METHODS

Classical relaxation methods can be easily characterized by adopting the following notation from [60]. According to the **LFA** we consider the discrete

problem on an infinite grid

$$L_h u_h(\mathbf{x}) = f_h(\mathbf{x}) \quad (\mathbf{x} \in G_h, u_h \in \mathcal{F}(G_h)).$$

The discrete operator L_h is divided into a sum

$$L_h = L_h^+ + L_h^0 + L_h^-. \quad (5.5)$$

For a specification of this splitting, recall the stencil notation for L_h given by

$$L_h u_h(\mathbf{x}) = \sum_{\kappa \in J} \ell_{\kappa} u_h(\mathbf{x} + \kappa h).$$

Then (5.5) is characterized by a corresponding splitting of the index set J into three disjoint subsets: J^+ , J^0 , and J^- . J^0 characterizes those grid points where the unknowns are smoothed simultaneously, for example, in a line or a plane. At grid points related to J^- the relaxation employs old approximations whereas at grid points defined by J^+ new values are already available. Using this convention we consider relaxation methods with a relaxation parameter ω defined by

$$\begin{aligned} (L_h^0 + \omega L_h^+) \bar{e}_h(\mathbf{x}) &= ((1 - \omega)L_h^0 - \omega L_h^-) e_h(\mathbf{x}) \quad \text{for } \mathbf{x} \in \tilde{G}_h \\ \bar{e}_h(\mathbf{x}) &= e_h(\mathbf{x}) \quad \text{for } \mathbf{x} \in G_h \setminus \tilde{G}_h. \end{aligned} \quad (5.6)$$

e_h and \bar{e}_h denote the error before and after one complete or one partial relaxation step. Considering a complete relaxation step, we have $\tilde{G}_h = G_h$ whereas in the case of a partial relaxation step \tilde{G}_h denotes a certain subspace of G_h . In this section, we focus on relaxation methods consisting of *one complete* sweep over all grid points, like Jacobi or lexicographic Gauss-Seidel relaxation. Smoothing analysis is particularly *simple* for these type of relaxations because the Fourier components are eigenfunctions of such a relaxation method.

The Fourier symbols of L_h^+ , L_h^0 , and L_h^- are simply given by

$$\begin{aligned} \tilde{L}_h^+(\boldsymbol{\theta}) &= \sum_{\kappa \in J^+} \ell_{\kappa} \exp(i\boldsymbol{\theta}\kappa), \quad \tilde{L}_h^0(\boldsymbol{\theta}) = \sum_{\kappa \in J^0} \ell_{\kappa} \exp(i\boldsymbol{\theta}\kappa), \\ \text{and } \tilde{L}_h^-(\boldsymbol{\theta}) &= \sum_{\kappa \in J^-} \ell_{\kappa} \exp(i\boldsymbol{\theta}\kappa) \quad (\boldsymbol{\theta} \in \Theta); \end{aligned}$$

see Definition 5.1. Assuming a complete relaxation step (i.e., $\tilde{G}_h = G_h$), it is then possible to calculate the Fourier symbol which corresponds to the relaxation operator $S_h(\omega)$ defined by (5.6):

$$\begin{aligned} S_h(\omega) \varphi_h(\boldsymbol{\theta}, \mathbf{x}) &= A(\boldsymbol{\theta}, \omega) \varphi_h(\boldsymbol{\theta}, \mathbf{x}) \quad \text{for } \boldsymbol{\theta} \in \Theta \quad \text{and } \mathbf{x} \in G_h \\ \text{with } A(\boldsymbol{\theta}, \omega) &= \frac{(1 - \omega)\tilde{L}_h^0(\boldsymbol{\theta}) - \omega\tilde{L}_h^-(\boldsymbol{\theta})}{\tilde{L}_h^0(\boldsymbol{\theta}) + \omega\tilde{L}_h^+(\boldsymbol{\theta})}. \end{aligned} \quad (5.7)$$

(5.7) only makes sense for $\tilde{L}_h^0(\boldsymbol{\theta}) + \omega \tilde{L}_h^+(\boldsymbol{\theta}) \neq 0$. If this condition does not hold for certain $\boldsymbol{\theta}^*$, they have to be excluded from the analysis. The corresponding results of the Fourier analysis for this slightly shrunk space of frequencies have to be interpreted carefully. For an example, see the discussion of the measure of h -ellipticity for the convection diffusion equation in Section 5.8.3.

In the following, we simply have to specify J^0 , J^- , and \tilde{G}_h in order to define the relaxation method under consideration. Of course, J^+ is then given by $J^+ = J \setminus (J^0 \cup J^-)$.

5.3.1 Jacobi relaxation

For Jacobi-type relaxations we have $\tilde{G}_h = G_h$. An ω -JAC point smoother is characterized by

$$J^0 = \{(0, \dots, 0)\} \quad \text{and} \quad J^- = J \setminus J^0. \quad (5.8)$$

To define general block relaxation methods for d -dimensional problems, we apply the following definition; see [73].

Definition 5.5 (Block relaxation). We split the index set $\mathcal{I} = \{1, \dots, d\}$ of coordinate indices into two disjoint subsets \mathcal{I}^b and \mathcal{I}^p . Unknowns at different grid points \mathbf{x} and $\bar{\mathbf{x}}$ are relaxed simultaneously in a block (for example, a line or a plane) if $x_j = \bar{x}_j$ holds for at least one $j \in \mathcal{I}^b$. The superscripts “b” and “p” refer to “blockwise” and “pointwise”, respectively. \blacktriangle

Using this definition, it is possible to specify ω -JAC block relaxation methods by

$$J^0 = \{\boldsymbol{\kappa} \in J \mid \kappa_j = 0 \text{ for } j \in \mathcal{I}^p\} \quad \text{and} \quad J^- = J \setminus J^0. \quad (5.9)$$

Example 5.4 (ω -JAC (x_1, x_2) -plane relaxation, $d = 3$). To illustrate the above definition of general block relaxations we consider ω -JAC (x_1, x_2) -plane relaxation applied to a three-dimensional operator which is described by a 7-point stencil like the central discretization of the Laplacian. Here, we have $\mathcal{I}^b = \{1, 2\}$, $\mathcal{I}^p = \{3\}$ and consequently

$$J^0 = \{\boldsymbol{\kappa} \in J \mid \kappa_3 = 0\} = \{(0, 0, 0), (1, 0, 0), (-1, 0, 0), (0, 1, 0), (0, -1, 0)\}.$$

Unknowns at \mathbf{x} and $\bar{\mathbf{x}}$ are smoothed simultaneously in a plane, if $x_1 = \bar{x}_1$ or $x_2 = \bar{x}_2$ holds. \blacktriangleleft

Due to $\tilde{G}_h = G_h$ and $J^+ = \emptyset$ in the case of ω -JAC relaxations, the general description (5.6) reduces to

$$L_h^0 \bar{e}_h(\mathbf{x}) = ((1 - \omega)L_h^0 - \omega L_h^-) e_h(\mathbf{x}) \quad \text{for } \mathbf{x} \in G_h.$$

This yields the following Fourier symbol for ω -JAC relaxations

$$A(\boldsymbol{\theta}, \omega) = 1 - \omega \left(1 + \frac{\tilde{L}_h^-(\boldsymbol{\theta})}{\tilde{L}_h^0(\boldsymbol{\theta})} \right) = 1 - \omega \frac{\tilde{L}_h(\boldsymbol{\theta})}{\tilde{L}_h^0(\boldsymbol{\theta})} \quad (\boldsymbol{\theta} \in \Theta). \quad (5.10)$$

Example 5.5 (Fourier symbol of point JAC for the Laplacian). For the d -dimensional Laplacian and ω -JAC point relaxation, we have $\tilde{L}_h(\boldsymbol{\theta}) = \frac{1}{h^2}(2d - \sum_{j=1}^d 2\cos(\theta_j))$ and $\tilde{L}_h^0(\boldsymbol{\theta}) = \frac{2d}{h^2}$ leading to

$$A(\boldsymbol{\theta}, \omega) = 1 - \omega + \frac{\omega}{d} \left(\sum_{j=1}^d \cos(\theta_j) \right).$$

◀

Example 5.6 (Fourier symbol of (x_1, x_2) -plane JAC for the Laplacian, $d = 3$). Applying ω -JAC (x_1, x_2) -plane relaxation to the three-dimensional Laplacian gives $\tilde{L}_h^0(\boldsymbol{\theta}) = \frac{1}{h^2}(6 - 2\cos(\theta_1) - 2\cos(\theta_2))$ (see Example 5.4 for J^0) yielding

$$A(\boldsymbol{\theta}, \omega) = 1 - \omega + \omega \frac{\cos(\theta_3)}{3 - \cos(\theta_1) - \cos(\theta_2)}.$$

◀

5.3.2 Lexicographic Gauss-Seidel relaxation

First of all, note that for all GS-LEX-type relaxations $\tilde{G}_h = G_h$ holds. General expressions for J^+ , J^0 , and J^- are rather intricate for point and block GS-LEX-type relaxations in d dimensions. Therefore, it is more convenient to give several examples of GS-LEX smoothing in two and three dimensions which are most relevant (and, in particular, which are implemented in the accompanying software) .

In two dimensions we distinguish four different point smoothers characterized by the order in which the grid points are visited. If the relaxation marches rowwise from left to right and from bottom to top, we speak of forward GS-LEX defined by

$$J^0 = \{(0, 0)\} \quad \text{and} \quad J^- = \{\boldsymbol{\kappa} \in J \mid \kappa_2 > 0\} \cup \{\boldsymbol{\kappa} \in J \mid \kappa_2 = 0 \text{ and } \kappa_1 > 0\}.$$

If the order of the grid points is of minor importance, we simply speak of GS-LEX as already done above.

Example 5.7 (Fourier symbol of point GS-LEX for the Laplacian). The Fourier symbol of forward GS-LEX point relaxation applied to the two-dimensional Laplacian reads

$$A(\boldsymbol{\theta}, \omega) = \frac{(1 - \omega)4 + \omega(\exp(i\theta_1) + \exp(i\theta_2))}{4 - \omega(\exp(-i\theta_1) + \exp(-i\theta_2))}.$$

◀

A rowwise marching from right to left and from top to bottom is called backward lexicographic Gauss-Seidel relaxation:

$$J^0 = \{(0,0)\} \quad \text{and} \quad J^- = \{\kappa \in J \mid \kappa_2 < 0\} \cup \{\kappa \in J \mid \kappa_2 = 0 \text{ and } \kappa_1 < 0\}.$$

Note that there is no natural starting point for lexicographical Gauss-Seidel relaxations on an infinite grid. However, in practice GS-LEX is applied on a finite domain Ω_h . Figure 5.3 shows the numbering of grid points on a square domain for forward (first picture) point GS-LEX and backward (second picture) point GS-LEX, respectively. In forward point GS-LEX, the relaxation starts in the southwestern corner whereas in backward point GS-LEX it starts in the northeastern corner, leading to the abbreviations GS-LEX-SW and GS-LEX-NE, respectively.

21 22 23 24 25	5 4 3 2 1	25 24 23 22 21	1 2 3 4 5
16 17 18 19 20	10 9 8 7 6	20 19 18 17 16	6 7 8 9 10
11 12 13 14 15	15 14 13 12 11	15 14 13 12 11	11 12 13 14 15
6 7 8 9 10	20 19 18 17 16	10 9 8 7 6	16 17 18 19 20
1 2 3 4 5	25 24 23 22 21	5 4 3 2 1	21 22 23 24 25
GS-LEX-SW	GS-LEX-NE	GS-LEX-SE	GS-LEX-NW

FIGURE 5.3: Different numberings of grid points for point GS-LEX.

Regarding Figure 5.3 the definition and meaning of GS-LEX-SE and GS-LEX-NW should be obvious.

Remark 5.1 (Four direction GS-LEX). A “robust” relaxation method for the first-order upwind discretization of the convection diffusion equation, denoted by four direction (4DIR) GS-LEX, consists of four consecutive sweeps of GS-LEX with different ordering of grid points, i.e., GS-LEX-SW followed by GS-LEX-NE, GS-LEX-SE, and GS-LEX-NW. The corresponding operator reads

$$S_h^{4\text{DIR}}(\omega) := S_h^{\text{NW}}(\omega) \cdot S_h^{\text{SE}}(\omega) \cdot S_h^{\text{NE}}(\omega) \cdot S_h^{\text{SW}}(\omega).$$

Further details are given in Section 5.8.3. ►

A similar distinction between forward and backward relaxation can be made for two-dimensional line smoothers. Forward and backward x_1 -line GS-LEX are defined by

$$\begin{aligned} J^0 &= \{\kappa \in J \mid \kappa_2 = 0\}, & J^- &= \{\kappa \in J \mid \kappa_2 > 0\} \\ \text{and } J^0 &= \{\kappa \in J \mid \kappa_2 = 0\}, & J^- &= \{\kappa \in J \mid \kappa_2 < 0\}, \end{aligned}$$

respectively. For x_2 -line GS-LEX an analog definition applies.

The generalization to three-dimensional GS-LEX-type relaxations is straightforward. In analogy to the two-dimensional case (see Figure 5.3), we have eight different point smoothers. For example, forward GS-LEX is given by

$$J^0 = \{(0, 0, 0)\} \quad \text{and} \quad J^- = \{\kappa \in J \mid \kappa_3 > 0\} \cup \{\kappa \in J \mid \kappa_3 = 0 \text{ and } \kappa_2 > 0\} \\ \cup \{\kappa \in J \mid \kappa_3 = \kappa_2 = 0 \text{ and } \kappa_1 > 0\}.$$

Moreover, we distinguish six different line relaxations (x_1 -, x_2 -, x_3 -line in a forward and backward manner) and six different plane smoothers ((x_1, x_2) -, (x_1, x_3) -, (x_2, x_3) -plane in forward and backward manner).

Example 5.8 (Forward GS-LEX block relaxations). Forward x_1 -line and forward (x_1, x_2) -plane GS-LEX relaxation are defined by

$$J^0 = \{\kappa \in J \mid \kappa_2 = \kappa_3 = 0\}, \quad J^- = \{\kappa \in J \mid \kappa_3 > 0\} \\ \cup \{\kappa \in J \mid \kappa_3 = 0 \text{ and } \kappa_2 > 0\} \\ J^0 = \{\kappa \in J \mid \kappa_3 = 0\}, \quad J^- = \{\kappa \in J \mid \kappa_3 > 0\},$$

respectively. The Fourier symbol of x_1 -line GS-LEX applied to the three-dimensional Laplacian reads

$$A(\theta, \omega) = \frac{(1 - \omega)(6 - 2 \cos(\theta_1)) + \omega(\exp(i\theta_2) + \exp(i\theta_3))}{6 - 2 \cos(\theta_1) - \omega(\exp(-i\theta_2) + \exp(-i\theta_3))}.$$

◀

5.3.3 A first definition of the smoothing factor

After the description of JAC- and GS-LEX-type relaxation methods, we continue with the definition of their *smoothing factor*. Considering Fourier smoothing analysis, one investigates the influence of a smoothing operator $S_h(\omega)$ to the high-frequency error components as the multigrid idea is based on the assumption that high-frequency components are smoothed out by the relaxation whereas the low-frequency components are reduced by the coarse-grid correction; see Section 1.5.

Definition 5.6 (Smoothing Factor for ω -JAC and ω -GS-LEX): Let $A(\theta, \omega)$ denote the Fourier symbol of ω -JAC or ω -GS-LEX. Then we define the corresponding smoothing factors by

$$\rho_1(\omega) := \sup_{\theta \in \Theta_{\text{high}}} |A(\theta, \omega)|.$$

In analogy to the two- and three-grid factors to be defined in Chapter 6 it could also be named a *one-grid* factor as it only takes the fine-grid operators into account. The subscript “1” refers to “one-grid.” If $\omega = 1$, we often use the shorter expression $\rho_1 := \rho_1(1)$. ▲

Example 5.9 (Smoothing factors for the Laplacian). The smoothing factor for damped ($\omega = 4/5$) JAC point relaxation applied to the two-dimensional Laplacian reads

$$\rho_1(4/5) = \sup_{\boldsymbol{\theta} \in \Theta_{\text{high}}} \left| \frac{1}{5} + \frac{2}{5}(\cos(\theta_1) + \cos(\theta_2)) \right|.$$

The supremum in case of standard coarsening is attained at $\boldsymbol{\theta}_{\text{sup}} = (\pm\pi, \pm\pi)$, $(\pi/2, \pm\pi)$, $(\pm\pi, \pi/2)$ yielding $\rho_1(4/5) = 3/5$; compare with Section 1.3.3. For the derivation of the optimal damping parameter $\omega = 4/5$, we refer to Example 5.17.

An application of (undamped) point GS-LEX to the two-dimensional Laplacian in connection with standard coarsening leads to

$$\rho_1 = \sup_{\boldsymbol{\theta} \in \Theta_{\text{high}}} \left| \frac{\exp(i\theta_1) + \exp(i\theta_2)}{4 - \exp(-i\theta_1) + \exp(-i\theta_2)} \right| = 1/2$$

with $\boldsymbol{\theta}_{\text{sup}} = (\pi/2, \cos^{-1}(4/5))$, $(\cos^{-1}(4/5), \pi/2)$ [65]. ◀

Remark 5.2 (Finite dimensional Fourier space). Note that the Fourier space (5.4) has a nondenumerable basis as $\boldsymbol{\theta}$ varies continuously in $(\pi, \pi]^d$. The use of infinite dimensional spaces and operators leads to some technical simplifications in the analysis, see [60]. However, in general the supremum in the definition of the smoothing factor cannot be calculated analytically as in the above example. The practical computation performed by the Fourier analysis software is based on a *finite dimensional* Fourier space. This finite dimensional Fourier space $\mathcal{F}_{h\boldsymbol{\theta}}$ is related to the mesh size h under consideration if the same values are selected for the parameters fixing the physical “gridsize” h and the “Fourier gridsizes” $h_{\boldsymbol{\theta}}$ (compare with the **Discretization** dialogue shown in Figure 3.8). Then $\mathcal{F}_{h\boldsymbol{\theta}}$ is given by

$$\mathcal{F}_{h\boldsymbol{\theta}} := \text{span} \left\{ \varphi_h(\boldsymbol{\theta}, \cdot) : \boldsymbol{\theta} \in \Theta^{\text{P}} := (-\pi, \pi]^d \cap G_{h\boldsymbol{\theta}} \right\} \quad (5.11)$$

with $G_{h\boldsymbol{\theta}} := \{\boldsymbol{\theta} = h_{\boldsymbol{\theta}}\boldsymbol{\kappa} \text{ with } h_{\boldsymbol{\theta}} = 2\pi h, \boldsymbol{\kappa} \in \mathbb{Z}^d\}$.

The definitions of Θ_{low} and Θ_{high} have to be changed accordingly, leading to $\Theta_{\text{low}}^{\text{P}} := \Theta_{\text{low}} \cap G_{h\boldsymbol{\theta}}$ and $\Theta_{\text{high}}^{\text{P}} := \Theta_{\text{high}} \cap G_{h\boldsymbol{\theta}}$. Hence, the supremum is replaced by a maximum, i.e.,

$$\rho_1^{\text{P}}(\omega) := \max_{\boldsymbol{\theta} \in \Theta_{\text{high}}^{\text{P}}} |A(\boldsymbol{\theta}, \omega)|, \quad (5.12)$$

which can easily be calculated numerically. The superscript “P” refers to “periodic.” The reason for this choice is explained in the following remark. In principle, the supremum can be approximated by increasing the parameter which determines the Fourier gridsizes while the physical mesh size remains fixed. However, in light of Remark 5.4 given below we do not recommend such an approach. ▶

Remark 5.3 (Exact analysis for periodic boundary conditions). Using this finite dimensional Fourier space, the local Fourier analysis becomes an exact analysis for certain model problems on rectangular domains with periodic boundary conditions. A famous exception is GS-LEX relaxation. As discussed in the previous subsection, there is no natural starting point for GS-LEX on an infinite grid in contrast to a finite domain, see Figure 5.3. As a consequence, the infinite grid operator does not coincide with its variant on a finite domain. This observation is described in detail in [67] using a matrix notation. However, the difference between these operators is solely induced by the boundary treatment. Away from the boundary we can expect that the smoothing properties of the infinite grid operator give a reliable approximation for the relaxation operator on the finite grid as smoothing is a purely local process; see, for example, [4, 9, 62]. ►

Such an exact analysis that explicitly takes into account the underlying domain and the corresponding boundary conditions is known as model problem analysis or rigorous Fourier analysis [60, 62]. An example of the model problem analysis has already been given in Section 1.3 investigating the two-dimensional Laplacian on the unit square with Dirichlet (D) boundary conditions. Here, the appropriate eigenfunctions or Fourier components were given by certain sine functions $\varphi_h^D(\boldsymbol{\theta}, \cdot)$ (1.17). The generalization of these eigenfunctions to d -dimensional applications is clearly given by

$$\begin{aligned}\varphi_h^D(\boldsymbol{\theta}, \mathbf{x}) &:= \prod_{j=1}^d \sin(\theta_j x_j / h_j) \quad \text{with } \mathbf{x} \in [0, 1]^d \cap G_h \quad \text{and} \\ \boldsymbol{\theta} \in \Theta^D &:= \{ \boldsymbol{\theta} \mid \boldsymbol{\theta} = (\theta_1, \dots, \theta_d) \text{ with } \theta_j = \pi m_j / n_j, \\ &\quad 1 \leq m_j \leq n_j - 1, n_j = 1/h_j (j = 1, \dots, d) \}.\end{aligned}$$

For $\varphi_h^D(\boldsymbol{\theta}, \cdot)$, Fourier frequencies $\boldsymbol{\theta}$ with $\theta_j = 0$ ($j \in \{1, \dots, d\}$) do not occur in Θ^D , but they are contained in Θ for the exponential components $\varphi_h(\boldsymbol{\theta}, \cdot)$. This indicates that a more realistic prediction of the smoothing factor is obtained by leaving out the Fourier frequencies with $\theta_j = 0$, as it has been observed in [16, 65, 72]. Based on the heuristics, one might mimic the effect of Dirichlet boundary conditions by removing the “zero frequencies” leading to the following modified definition of the smoothing factor.

Definition 5.7 (Smoothing Factor for ω -JAC and ω -GS-LEX; Dirichlet boundary conditions). To study the effect of Dirichlet boundary conditions we apply the following slightly modified smoothing factor:

$$\begin{aligned}\rho_1^D(\omega) &:= \max_{\boldsymbol{\theta} \in \Theta_{\text{high}}^D} |A(\boldsymbol{\theta}, \omega)| \quad \text{with} \\ \Theta_{\text{high}}^D &= (\Theta_{\text{high}} \setminus \{ \boldsymbol{\theta} \mid \theta_j = 0, j \in \{1, \dots, d\} \}) \cap G_{h_\theta}.\end{aligned}\tag{5.13}$$

As in the previous definition we will use $\rho_1^D := \rho_1^D(1)$. ▲

Definition 5.7 only makes sense if it is based on the finite dimensional Fourier space \mathcal{F}_{h_θ} yielding the calculation of a certain maximum (5.13). Otherwise the maximum would have to be replaced by a supremum. As a consequence, the exclusion of the zero frequencies would not make any difference compared to $\rho_1(\omega)$ as long as $\rho_1^D(\omega)$ depends continuously on θ . A particularly illustrative example—demonstrating the benefits of the above modification of the smoothing factor in case of Dirichlet boundary conditions—is given in Section 5.8.3 for the first-order upwind discretization of the convection diffusion equation.

Remark 5.4 (*h*-dependency of smoothing analysis). Equation (5.12) and Definition 5.7 lead to *h*-dependent results since Θ_{high}^P and Θ_{high}^D are related to the underlying computational grid under consideration. Obviously, $\rho_1^P(\omega) \leq \rho_1(\omega)$ holds, since $\Theta^P \subset \Theta$. If $A(\theta, \omega)$ is a smooth function in θ , then $\rho_1(\omega) - \rho_1^P(\omega) = O(h_j^m)$ follows for some $m \geq 1$ [65]. This means that $\rho_1^P(\omega)$ quickly approaches the limit $\rho_1(\omega)$ for “usual” elliptic problems [28].

LFA based on Θ yields *h*-independent results for discretization operators involving partial derivatives of the same order only, since the Fourier frequencies are not related to a certain grid in contrast to Θ^P and Θ^D . More precisely, the continuously defined set $\Theta (= \lim_{h_1, \dots, h_d \rightarrow 0} \Theta^P)$ implies that $h_1, \dots, h_d \rightarrow 0$. These limits will, of course, never be attained in a real numerical calculation where it is reasonable to assume a lower bound $h_0 \approx 10^{-3}$ for h_1, \dots, h_d [72].

It may happen that a set of problems is controlled by a parameter ε such that $\rho_1(\omega) - \rho_1^P(\omega) = O(h_j^m/\varepsilon)$ [65]. Considering only practical values $h_j \geq h_0$ ($j = 1, \dots, d$), there might be a big difference between $\rho_1^P(\omega)$, $\rho_1^D(\omega)$ and $\rho_1(\omega)$ if $\varepsilon \ll 1$ as for the anisotropic diffusion or the convection diffusion equation [43, 65, 72]. In this case, $\rho_1(\omega)$ can be too pessimistic and the *h*-dependent values $\rho_1^P(\omega)$ and $\rho_1^D(\omega)$ yield more reliable predictions.

From the above discussion it is obvious that $\rho_1^P(\omega)$ and $\rho_1^D(\omega)$ —provided by the accompanying software—can be considered as the relevant values for practical applications. ►

5.4 PATTERN RELAXATIONS

If the relaxation method under consideration consists of several partial relaxation steps—i.e., $\widehat{G}_h \neq G_h$, see (5.6)—we speak of pattern relaxations. Here, the situation is somewhat more difficult as the Fourier components are no longer eigenfunctions of the relaxation operator. However, it still leaves certain low-dimensional subspaces of Fourier components invariant yielding a block-diagonal Fourier representation of the smoothing operator consisting

of small blocks only. This representation allows for an easy calculation of smoothing factors (to be defined below for pattern relaxations).

5.4.1 Red-black Jacobi (RB-JAC) relaxations

Among the most popular pattern relaxations are red-black Jacobi smoothers (RB-JAC) applying two partial relaxation steps. For red-black relaxations, the grid $G_h = \{\mathbf{x} = \boldsymbol{\kappa}h \mid \boldsymbol{\kappa} \in \mathbb{Z}^d\}$ is divided into two disjoint subsets G_h^R and G_h^B referring to “red” and “black” points, respectively. One complete red-black relaxation $S_h^{RB}(\omega)$ consists of two steps. In the first half-step ($S_h^R(\omega)$) only the unknowns located at red points are smoothed whereas the unknowns at black points remain unchanged. Then, in the second half-step ($S_h^B(\omega)$) the unknowns at the black points are changed using the new values at the red points calculated in the first half-step. The iteration operator of one complete red-black iteration is given by

$$S_h^{RB}(\omega) = S_h^B(\omega) \cdot S_h^R(\omega).$$

With

$$\tilde{G}_h = G_h^R, \quad G_h \setminus \tilde{G}_h = G_h^B \quad \text{and} \quad \tilde{G}_h = G_h^B, \quad G_h \setminus \tilde{G}_h = G_h^R$$

for the first and for the second half-step, respectively, notation (5.6) applies.

For point ω -RB-JAC, the computational grid is subdivided into red and black points in a checkerboard manner as it has already been described in Section 5.2.2 in the context of red-black coarsening, i.e.,

$$G_h^R = \{\mathbf{x} = \boldsymbol{\kappa}h \in G_h \mid \sum_{j=1}^d \kappa_j \text{ even}\}, \quad G_h^B = \{\mathbf{x} = \boldsymbol{\kappa}h \in G_h \mid \sum_{j=1}^d \kappa_j \text{ odd}\}; \quad (5.14)$$

compare with Figure 5.4 for the two-dimensional case. In both half-steps we use the same splitting of the index set J as for point ω -JAC; see (5.8).

For general block ω -RB-JAC relaxations, recall the splitting of the coordinate indices into \mathcal{I}^b and \mathcal{I}^p from Definition 5.5. Here, it is more appropriate to speak of red blocks and black blocks that are collected in

$$G_h^R = \{\mathbf{x} = \boldsymbol{\kappa}h \in G_h \mid \sum_{j \in \mathcal{I}^p} \kappa_j \text{ even}\}, \quad G_h^B = \{\mathbf{x} = \boldsymbol{\kappa}h \in G_h \mid \sum_{j \in \mathcal{I}^p} \kappa_j \text{ odd}\}, \quad (5.15)$$

respectively. The splitting of J is adopted from the block ω -JAC relaxations; see (5.9). Line or plane RB-JAC relaxations are also known as line or plane ZEBRA relaxations. The corresponding distinction between red and black points is specified in Tables 5.1 and 5.2. x_1 -line relaxation for $d = 2$ is illustrated in Figure 5.4.

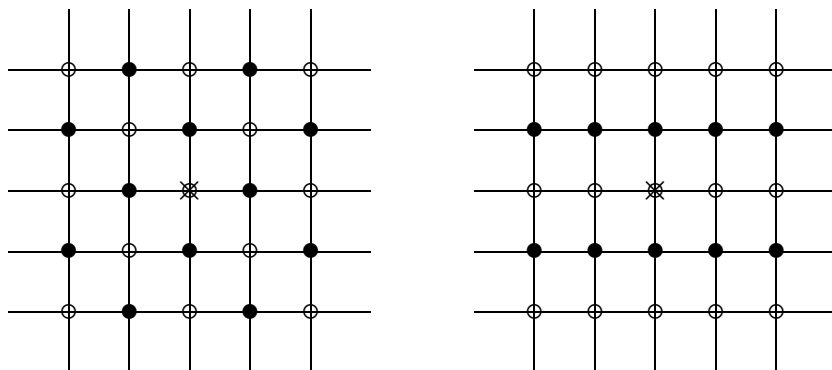


FIGURE 5.4: Red (○) and black (●) points for point (left) and x_1 -line (right) RB-JAC relaxation; $d = 2$. \times denotes the origin of G_h .

TABLE 5.1: Characterization of G_h^R and G_h^B for two-dimensional RB-JAC relaxations; compare with (5.14) and (5.15) for the general definitions.

relaxation	G_h^R	G_h^B
point	$\kappa_1 + \kappa_2$ even	$\kappa_1 + \kappa_2$ odd
x_1 -line	κ_2 even	κ_2 odd
x_2 -line	κ_1 even	κ_1 odd

5.4.2 Spaces of $2h$ -harmonics

Pattern relaxations involving partial step operators do not have simple scalar Fourier symbols like (5.7) or (5.10) because they couple certain Fourier components. In this respect it is convenient to consider the following low-dimensional spaces of Fourier components which turn out to be invariant under an application of RB-JAC relaxations.

Definition 5.8 ($2h$ -harmonics, d -dimensional case): The spaces of $2h$ -harmonics for $\theta \in \Theta_{2h} := (-\pi/2, \pi/2]^d$ are given by

$$\mathcal{F}_{2h}(\theta) := \text{span} \{ \varphi_h(\theta^\alpha, \cdot) \mid \alpha = (\alpha_1, \dots, \alpha_d), \alpha_j \in \{0, 1\}, j = 1, \dots, d \} \text{ with } \theta = \theta^{(0, \dots, 0)} \in \Theta_{2h} \text{ and } \theta^\alpha := \theta^{(0, \dots, 0)} - (\alpha_1 \text{sign}(\theta_1), \dots, \alpha_d \text{sign}(\theta_d)) \pi.$$

▲

TABLE 5.2: Characterization of G_h^R and G_h^B for three-dimensional RB-JAC relaxations; compare with (5.14) and (5.15) for the general definitions.

relaxation	G_h^R	G_h^B
point	$\kappa_1 + \kappa_2 + \kappa_3$ even	$\kappa_1 + \kappa_2 + \kappa_3$ odd
x_1 -line	$\kappa_2 + \kappa_3$ even	$\kappa_2 + \kappa_3$ odd
x_2 -line	$\kappa_1 + \kappa_3$ even	$\kappa_1 + \kappa_3$ odd
x_3 -line	$\kappa_1 + \kappa_2$ even	$\kappa_1 + \kappa_2$ odd
(x_1, x_2) -plane	κ_3 even	κ_3 odd
(x_1, x_3) -plane	κ_2 even	κ_2 odd
(x_2, x_3) -plane	κ_1 even	κ_1 odd

In the following, $\alpha = (\alpha_1, \dots, \alpha_d)$ is often abbreviated by $\alpha_1 \dots \alpha_d$. For example, we use 000 instead of $(0, 0, 0)$; see below.

Obviously, the $2h$ -harmonics generate the whole Fourier space. More precisely, we have

$$\mathcal{F} = \bigoplus_{\theta \in \Theta_{2h}} \mathcal{F}_{2h}(\theta);$$

see Definition 5.2. The motivation for such a splitting of \mathcal{F} and the meaning of the name “ $2h$ -harmonics” will become more obvious in the context of the k -grid analysis; see Section 6.2. The subscript “ $2h$ ” stands for standard coarsening ($H = 2h$). (Note that $\Theta_{2h} = \Theta_{\text{low}}$, where *low* refers to standard coarsening.) A set of Fourier frequencies generating one space of $2h$ -harmonics is shown in Figure 5.5.

In the context of Fourier smoothing analysis it is crucial that the pattern relaxation methods ω -RB-JAC leave these spaces of $2h$ -harmonics invariant. This means that pattern relaxations $S_h(\omega)$ are unitarily equivalent to a block matrix consisting of 2^d -blocks, as the $2h$ -harmonics build 2^d -dimensional subspaces. Hence, the Fourier symbols of these pattern relaxations based on the $2h$ -harmonics are given by $(2^d \times 2^d)$ complex matrices $\tilde{S}_h(\theta, \omega)$, i.e.,

$$S_h(\omega)|_{\mathcal{F}_{2h}(\theta)} = \tilde{S}_h(\theta, \omega) \in \mathbb{C}^{2^d \times 2^d}.$$

Example 5.10 (Fourier representation of point RB-JAC relaxation, $d = 2$). The Fourier representations for several pattern relaxations in the two-dimensional case are derived, for example, in [59, 60] assuming the following order for the multiindex α :

$$(0, 0), \quad (1, 1), \quad (1, 0), \quad (0, 1). \tag{5.16}$$

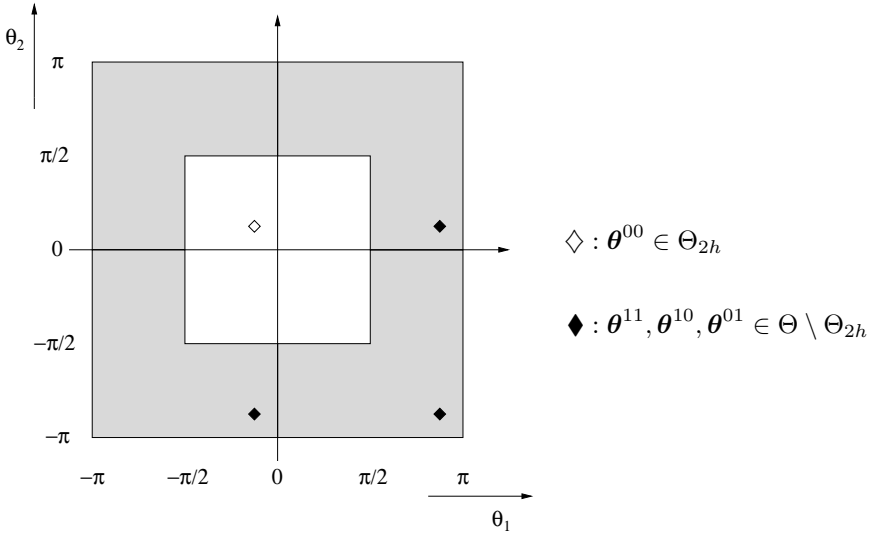


FIGURE 5.5: Sample set of Fourier frequencies generating a space of $2h$ -harmonics. Θ_{2h} : white region, $\Theta \setminus \Theta_{2h}$: shaded region.

For point RB-JAC relaxation one obtains

$$\begin{aligned} \tilde{S}_h^{\text{RB}}(\boldsymbol{\theta}, \omega) &= \tilde{S}_h^{\text{B}}(\boldsymbol{\theta}, \omega) \cdot \tilde{S}_h^{\text{R}}(\boldsymbol{\theta}, \omega) \quad \text{with} \\ \tilde{S}_h^{\text{R}}(\boldsymbol{\theta}, \omega) &= \frac{1}{2} \begin{pmatrix} A^{00} + 1 & A^{11} - 1 & 0 & 0 \\ A^{00} - 1 & A^{11} + 1 & 0 & 0 \\ 0 & 0 & A^{10} + 1 & A^{01} - 1 \\ 0 & 0 & A^{10} - 1 & A^{01} + 1 \end{pmatrix}, \\ \tilde{S}_h^{\text{B}}(\boldsymbol{\theta}, \omega) &= \frac{1}{2} \begin{pmatrix} A^{00} + 1 & -A^{11} + 1 & 0 & 0 \\ -A^{00} + 1 & A^{11} + 1 & 0 & 0 \\ 0 & 0 & A^{10} + 1 & -A^{01} + 1 \\ 0 & 0 & -A^{10} + 1 & A^{01} + 1 \end{pmatrix}, \end{aligned}$$

and $A^\alpha := A(\boldsymbol{\theta}^\alpha, \omega)$ from (5.10). The Fourier representations for x_1 -, x_2 -line RB-JAC relaxation ($d = 2$) are given in Appendix A.1. ◀

The three-dimensional case is tackled in [61], but the explicit derivation and definition of the (8×8) -blocks is lacking. To close this gap, we calculate the Fourier symbols for various *three-dimensional* RB-JAC relaxations. The following considerations are based on a straightforward modification of the two-dimensional case which is described in [59, 60]. The point relaxation is discussed in detail whereas the Fourier representations for the block relaxations are listed in Appendix A.2.

5.4.3 Auxiliary definitions and relations

We start with some basic definitions and relations which are crucial for the calculation of the Fourier symbols w.r.t. the $2h$ -harmonics. First of all, one has to prescribe an order of the basis elements $\varphi_h(\theta^\alpha, \cdot)$ of the spaces of $2h$ -harmonics. Here, we take the order

$$(0, 0, 0), (1, 1, 1), (1, 0, 0), (0, 1, 1), (0, 1, 0), (1, 0, 1), (0, 0, 1), (1, 1, 0)$$

for the multiindex α .

We distinguish eight different types of grid points within the infinite grid G_h , namely

$$G_h^\alpha := \{x = \kappa h \mid \kappa \in \mathbb{Z}^3, \kappa = \alpha \pmod{2}\}. \quad (5.17)$$

This compact but somewhat complex definition is broken down in Table 5.3.

TABLE 5.3: Characterization of G_h^α ($d = 3$); compare with (5.17).

G_h^{000}	$\kappa_1, \kappa_2, \kappa_3$ even	G_h^{111}	$\kappa_1, \kappa_2, \kappa_3$ odd
G_h^{100}	κ_1 odd; κ_2, κ_3 even	G_h^{011}	κ_1 even; κ_2, κ_3 odd
G_h^{010}	κ_1, κ_3 even; κ_2 odd	G_h^{101}	κ_1, κ_3 odd; κ_2 even
G_h^{001}	κ_1 even; κ_2, κ_3 odd	G_h^{110}	κ_1, κ_2 odd; κ_3 even

Remark 5.5 (Generalization to d dimensions). The infinite grid has to be subdivided into 2^d different kinds of grid points for a general treatment of pattern relaxations in d dimensions. Using α from Definition 5.8, (5.17) has to be adapted as follows

$$G_h^\alpha := \{x = \kappa h \mid \kappa \in \mathbb{Z}^d, \kappa = \alpha \pmod{2}\}. \quad (5.18)$$

The two-dimensional case is specified in Table 5.4. ►

TABLE 5.4: Characterization of G_h^α ($d = 2$); compare with (5.18).

G_h^{00}	κ_1, κ_2 even	G_h^{11}	κ_1, κ_2 , odd
G_h^{10}	κ_1 odd; κ_2 even	G_h^{01}	κ_1 even; κ_2 odd

For the $2h$ -harmonics, it can be easily established that the following relations are valid:

$$\begin{aligned}
 \varphi_h(\theta^{111}, x) &= \begin{cases} \varphi_h(\theta^{000}, x) & \text{for } x \in G_h^{000} \cup G_h^{011} \cup G_h^{101} \cup G_h^{110}, \\ -\varphi_h(\theta^{000}, x) & \text{for } x \in G_h^{111} \cup G_h^{100} \cup G_h^{010} \cup G_h^{001}. \end{cases} \\
 \varphi_h(\theta^{100}, x) &= \begin{cases} \varphi_h(\theta^{000}, x) & \text{for } x \in G_h^{000} \cup G_h^{011} \cup G_h^{010} \cup G_h^{001}, \\ -\varphi_h(\theta^{000}, x) & \text{for } x \in G_h^{111} \cup G_h^{100} \cup G_h^{110} \cup G_h^{101}. \end{cases} \\
 \varphi_h(\theta^{011}, x) &= \begin{cases} \varphi_h(\theta^{000}, x) & \text{for } x \in G_h^{000} \cup G_h^{111} \cup G_h^{011} \cup G_h^{100}, \\ -\varphi_h(\theta^{000}, x) & \text{for } x \in G_h^{110} \cup G_h^{101} \cup G_h^{010} \cup G_h^{001}. \end{cases} \\
 \varphi_h(\theta^{010}, x) &= \begin{cases} \varphi_h(\theta^{000}, x) & \text{for } x \in G_h^{000} \cup G_h^{101} \cup G_h^{100} \cup G_h^{001}, \\ -\varphi_h(\theta^{000}, x) & \text{for } x \in G_h^{111} \cup G_h^{110} \cup G_h^{011} \cup G_h^{010}. \end{cases} \\
 \varphi_h(\theta^{101}, x) &= \begin{cases} \varphi_h(\theta^{000}, x) & \text{for } x \in G_h^{000} \cup G_h^{111} \cup G_h^{101} \cup G_h^{010}, \\ -\varphi_h(\theta^{000}, x) & \text{for } x \in G_h^{110} \cup G_h^{011} \cup G_h^{100} \cup G_h^{001}. \end{cases} \\
 \varphi_h(\theta^{001}, x) &= \begin{cases} \varphi_h(\theta^{000}, x) & \text{for } x \in G_h^{000} \cup G_h^{110} \cup G_h^{100} \cup G_h^{010}, \\ -\varphi_h(\theta^{000}, x) & \text{for } x \in G_h^{111} \cup G_h^{101} \cup G_h^{011} \cup G_h^{001}. \end{cases} \\
 \varphi_h(\theta^{110}, x) &= \begin{cases} \varphi_h(\theta^{000}, x) & \text{for } x \in G_h^{000} \cup G_h^{111} \cup G_h^{110} \cup G_h^{001}, \\ -\varphi_h(\theta^{000}, x) & \text{for } x \in G_h^{101} \cup G_h^{011} \cup G_h^{100} \cup G_h^{010}. \end{cases}
 \end{aligned} \tag{5.19}$$

Furthermore, we introduce eight functions ψ_1, \dots, ψ_8 defined by:

$$\begin{aligned}
 \psi_1(x) &:= (\varphi_h(\theta^{000}, x) + \varphi_h(\theta^{111}, x) + \varphi_h(\theta^{100}, x) + \varphi_h(\theta^{011}, x) \\
 &\quad + \varphi_h(\theta^{010}, x) + \varphi_h(\theta^{101}, x) + \varphi_h(\theta^{001}, x) + \varphi_h(\theta^{110}, x))/8 \\
 \psi_2(x) &:= (\varphi_h(\theta^{000}, x) - \varphi_h(\theta^{111}, x) - \varphi_h(\theta^{100}, x) + \varphi_h(\theta^{011}, x) \\
 &\quad - \varphi_h(\theta^{010}, x) + \varphi_h(\theta^{101}, x) - \varphi_h(\theta^{001}, x) + \varphi_h(\theta^{110}, x))/8 \\
 \psi_3(x) &:= (\varphi_h(\theta^{000}, x) - \varphi_h(\theta^{111}, x) - \varphi_h(\theta^{100}, x) + \varphi_h(\theta^{011}, x) \\
 &\quad + \varphi_h(\theta^{010}, x) - \varphi_h(\theta^{101}, x) + \varphi_h(\theta^{001}, x) - \varphi_h(\theta^{110}, x))/8 \\
 \psi_4(x) &:= (\varphi_h(\theta^{000}, x) + \varphi_h(\theta^{111}, x) + \varphi_h(\theta^{100}, x) + \varphi_h(\theta^{011}, x) \\
 &\quad - \varphi_h(\theta^{010}, x) - \varphi_h(\theta^{101}, x) - \varphi_h(\theta^{001}, x) - \varphi_h(\theta^{110}, x))/8 \\
 \psi_5(x) &:= (\varphi_h(\theta^{000}, x) - \varphi_h(\theta^{111}, x) + \varphi_h(\theta^{100}, x) - \varphi_h(\theta^{011}, x) \\
 &\quad - \varphi_h(\theta^{010}, x) + \varphi_h(\theta^{101}, x) + \varphi_h(\theta^{001}, x) - \varphi_h(\theta^{110}, x))/8 \\
 \psi_6(x) &:= (\varphi_h(\theta^{000}, x) + \varphi_h(\theta^{111}, x) - \varphi_h(\theta^{100}, x) - \varphi_h(\theta^{011}, x) \\
 &\quad + \varphi_h(\theta^{010}, x) + \varphi_h(\theta^{101}, x) - \varphi_h(\theta^{001}, x) - \varphi_h(\theta^{110}, x))/8 \\
 \psi_7(x) &:= (\varphi_h(\theta^{000}, x) - \varphi_h(\theta^{111}, x) + \varphi_h(\theta^{100}, x) - \varphi_h(\theta^{011}, x) \\
 &\quad + \varphi_h(\theta^{010}, x) - \varphi_h(\theta^{101}, x) - \varphi_h(\theta^{001}, x) + \varphi_h(\theta^{110}, x))/8 \\
 \psi_8(x) &:= (\varphi_h(\theta^{000}, x) + \varphi_h(\theta^{111}, x) - \varphi_h(\theta^{100}, x) - \varphi_h(\theta^{011}, x) \\
 &\quad - \varphi_h(\theta^{010}, x) - \varphi_h(\theta^{101}, x) + \varphi_h(\theta^{001}, x) + \varphi_h(\theta^{110}, x))/8
 \end{aligned}$$

Using (5.19) it can be verified that

$$\begin{aligned}
 \psi_1(\mathbf{x}) &= \begin{cases} \varphi_h(\boldsymbol{\theta}^{000}, \mathbf{x}) & \text{for } \mathbf{x} \in G_h^{000}, \\ 0 & \text{for } \mathbf{x} \notin G_h^{000}. \end{cases} & \psi_2(\mathbf{x}) &= \begin{cases} \varphi_h(\boldsymbol{\theta}^{000}, \mathbf{x}) & \text{for } \mathbf{x} \in G_h^{111}, \\ 0 & \text{for } \mathbf{x} \notin G_h^{111}. \end{cases} \\
 \psi_3(\mathbf{x}) &= \begin{cases} \varphi_h(\boldsymbol{\theta}^{000}, \mathbf{x}) & \text{for } \mathbf{x} \in G_h^{100}, \\ 0 & \text{for } \mathbf{x} \notin G_h^{100}. \end{cases} & \psi_4(\mathbf{x}) &= \begin{cases} \varphi_h(\boldsymbol{\theta}^{000}, \mathbf{x}) & \text{for } \mathbf{x} \in G_h^{011}, \\ 0 & \text{for } \mathbf{x} \notin G_h^{011}. \end{cases} \\
 \psi_5(\mathbf{x}) &= \begin{cases} \varphi_h(\boldsymbol{\theta}^{000}, \mathbf{x}) & \text{for } \mathbf{x} \in G_h^{010}, \\ 0 & \text{for } \mathbf{x} \notin G_h^{010}. \end{cases} & \psi_6(\mathbf{x}) &= \begin{cases} \varphi_h(\boldsymbol{\theta}^{000}, \mathbf{x}) & \text{for } \mathbf{x} \in G_h^{101}, \\ 0 & \text{for } \mathbf{x} \notin G_h^{101}. \end{cases} \\
 \psi_7(\mathbf{x}) &= \begin{cases} \varphi_h(\boldsymbol{\theta}^{000}, \mathbf{x}) & \text{for } \mathbf{x} \in G_h^{001}, \\ 0 & \text{for } \mathbf{x} \notin G_h^{001}. \end{cases} & \psi_8(\mathbf{x}) &= \begin{cases} \varphi_h(\boldsymbol{\theta}^{000}, \mathbf{x}) & \text{for } \mathbf{x} \in G_h^{110}, \\ 0 & \text{for } \mathbf{x} \notin G_h^{110}. \end{cases}
 \end{aligned}$$

Altogether one obtains for arbitrary $c^\alpha \in \mathbb{C}$:

$$\begin{aligned}
 & c^{000} \psi_1(\mathbf{x}) + c^{111} \psi_2(\mathbf{x}) + c^{100} \psi_3(\mathbf{x}) + c^{011} \psi_4(\mathbf{x}) \\
 & + c^{010} \psi_5(\mathbf{x}) + c^{101} \psi_6(\mathbf{x}) + c^{001} \psi_7(\mathbf{x}) + c^{110} \psi_8(\mathbf{x}) = c^\alpha \varphi_h(\boldsymbol{\theta}^{000}, \mathbf{x}) \quad \text{for } \mathbf{x} \in G_h^\alpha. \\
 & c^{000} \psi_1(\mathbf{x}) - c^{111} \psi_2(\mathbf{x}) - c^{100} \psi_3(\mathbf{x}) + c^{011} \psi_4(\mathbf{x}) \\
 & - c^{010} \psi_5(\mathbf{x}) + c^{101} \psi_6(\mathbf{x}) - c^{001} \psi_7(\mathbf{x}) + c^{110} \psi_8(\mathbf{x}) = c^\alpha \varphi_h(\boldsymbol{\theta}^{111}, \mathbf{x}) \quad \text{for } \mathbf{x} \in G_h^\alpha. \\
 & c^{000} \psi_1(\mathbf{x}) - c^{111} \psi_2(\mathbf{x}) - c^{100} \psi_3(\mathbf{x}) + c^{011} \psi_4(\mathbf{x}) \\
 & + c^{010} \psi_5(\mathbf{x}) - c^{101} \psi_6(\mathbf{x}) + c^{001} \psi_7(\mathbf{x}) - c^{110} \psi_8(\mathbf{x}) = c^\alpha \varphi_h(\boldsymbol{\theta}^{100}, \mathbf{x}) \quad \text{for } \mathbf{x} \in G_h^\alpha. \\
 & c^{000} \psi_1(\mathbf{x}) + c^{111} \psi_2(\mathbf{x}) + c^{100} \psi_3(\mathbf{x}) + c^{011} \psi_4(\mathbf{x}) \\
 & - c^{010} \psi_5(\mathbf{x}) - c^{101} \psi_6(\mathbf{x}) - c^{001} \psi_7(\mathbf{x}) - c^{110} \psi_8(\mathbf{x}) = c^\alpha \varphi_h(\boldsymbol{\theta}^{011}, \mathbf{x}) \quad \text{for } \mathbf{x} \in G_h^\alpha. \\
 & c^{000} \psi_1(\mathbf{x}) - c^{111} \psi_2(\mathbf{x}) + c^{100} \psi_3(\mathbf{x}) - c^{011} \psi_4(\mathbf{x}) \\
 & - c^{010} \psi_5(\mathbf{x}) + c^{101} \psi_6(\mathbf{x}) + c^{001} \psi_7(\mathbf{x}) - c^{110} \psi_8(\mathbf{x}) = c^\alpha \varphi_h(\boldsymbol{\theta}^{010}, \mathbf{x}) \quad \text{for } \mathbf{x} \in G_h^\alpha. \\
 & c^{000} \psi_1(\mathbf{x}) + c^{111} \psi_2(\mathbf{x}) - c^{100} \psi_3(\mathbf{x}) - c^{011} \psi_4(\mathbf{x}) \\
 & + c^{010} \psi_5(\mathbf{x}) + c^{101} \psi_6(\mathbf{x}) - c^{001} \psi_7(\mathbf{x}) - c^{110} \psi_8(\mathbf{x}) = c^\alpha \varphi_h(\boldsymbol{\theta}^{101}, \mathbf{x}) \quad \text{for } \mathbf{x} \in G_h^\alpha. \\
 & c^{000} \psi_1(\mathbf{x}) - c^{111} \psi_2(\mathbf{x}) + c^{100} \psi_3(\mathbf{x}) - c^{011} \psi_4(\mathbf{x}) \\
 & + c^{010} \psi_5(\mathbf{x}) - c^{101} \psi_6(\mathbf{x}) - c^{001} \psi_7(\mathbf{x}) + c^{110} \psi_8(\mathbf{x}) = c^\alpha \varphi_h(\boldsymbol{\theta}^{001}, \mathbf{x}) \quad \text{for } \mathbf{x} \in G_h^\alpha. \\
 & c^{000} \psi_1(\mathbf{x}) + c^{111} \psi_2(\mathbf{x}) - c^{100} \psi_3(\mathbf{x}) - c^{011} \psi_4(\mathbf{x}) \\
 & - c^{010} \psi_5(\mathbf{x}) - c^{101} \psi_6(\mathbf{x}) + c^{001} \psi_7(\mathbf{x}) + c^{110} \psi_8(\mathbf{x}) = c^\alpha \varphi_h(\boldsymbol{\theta}^{110}, \mathbf{x}) \quad \text{for } \mathbf{x} \in G_h^\alpha.
 \end{aligned} \tag{5.20}$$

5.4.4 Fourier representation for RB-JAC point relaxation

Next we derive the Fourier representation $\tilde{S}_h^{\text{RB}}(\boldsymbol{\theta}, \omega)$ of RB-JAC point relaxation based on the $2h$ -harmonics by applying the preliminary considerations from the previous subsection. The general description (5.6) of a relaxation

method immediately yields

$$S_h^R(\omega) \varphi_h(\boldsymbol{\theta}, \mathbf{x}) = \begin{cases} A(\boldsymbol{\theta}, \omega) \varphi_h(\boldsymbol{\theta}, \mathbf{x}) & \text{for } \mathbf{x} \in G_h^R \\ \varphi_h(\boldsymbol{\theta}, \mathbf{x}) & \text{for } \mathbf{x} \in G_h^B \end{cases} \quad (5.21)$$

for the first Jacobi sweep over the red points with $A(\boldsymbol{\theta}, \omega)$ from (5.10).

Red and black points for point RB-JAC relaxation can be expressed by G_h^α :

$$G_h^R = G_h^{000} \cup G_h^{011} \cup G_h^{101} \cup G_h^{110} \quad \text{and} \quad G_h^B = G_h^{111} \cup G_h^{100} \cup G_h^{010} \cup G_h^{001}; \quad (5.22)$$

compare with [Tables 5.2](#) and [5.3](#). Combining (5.21), (5.22), and (5.20) gives

$$S_h^R(\omega) \varphi_h(\boldsymbol{\theta}^\alpha, \mathbf{x}) = \begin{cases} A(\boldsymbol{\theta}^{000}, \omega) (\psi_1(\mathbf{x}) + \psi_4(\mathbf{x}) + \psi_6(\mathbf{x}) + \psi_8(\mathbf{x}) \\ \quad + (\psi_2(\mathbf{x}) + \psi_3(\mathbf{x}) + \psi_5(\mathbf{x}) + \psi_7(\mathbf{x})) & (\alpha = (0, 0, 0)) \\ A(\boldsymbol{\theta}^{111}, \omega) (\psi_1(\mathbf{x}) + \psi_4(\mathbf{x}) + \psi_6(\mathbf{x}) + \psi_8(\mathbf{x}) \\ \quad + (-\psi_2(\mathbf{x}) - \psi_3(\mathbf{x}) - \psi_5(\mathbf{x}) - \psi_7(\mathbf{x})) & (\alpha = (1, 1, 1)) \\ A(\boldsymbol{\theta}^{100}, \omega) (\psi_1(\mathbf{x}) + \psi_4(\mathbf{x}) - \psi_6(\mathbf{x}) - \psi_8(\mathbf{x}) \\ \quad + (-\psi_2(\mathbf{x}) - \psi_3(\mathbf{x}) + \psi_5(\mathbf{x}) + \psi_7(\mathbf{x})) & (\alpha = (1, 0, 0)) \\ A(\boldsymbol{\theta}^{011}, \omega) (\psi_1(\mathbf{x}) + \psi_4(\mathbf{x}) - \psi_6(\mathbf{x}) - \psi_8(\mathbf{x}) \\ \quad + (\psi_2(\mathbf{x}) + \psi_3(\mathbf{x}) - \psi_5(\mathbf{x}) - \psi_7(\mathbf{x})) & (\alpha = (0, 1, 1)) \\ A(\boldsymbol{\theta}^{010}, \omega) (\psi_1(\mathbf{x}) - \psi_4(\mathbf{x}) + \psi_6(\mathbf{x}) - \psi_8(\mathbf{x}) \\ \quad + (-\psi_2(\mathbf{x}) + \psi_3(\mathbf{x}) - \psi_5(\mathbf{x}) + \psi_7(\mathbf{x})) & (\alpha = (0, 1, 0)) \\ A(\boldsymbol{\theta}^{101}, \omega) (\psi_1(\mathbf{x}) - \psi_4(\mathbf{x}) + \psi_6(\mathbf{x}) - \psi_8(\mathbf{x}) \\ \quad + (\psi_2(\mathbf{x}) - \psi_3(\mathbf{x}) + \psi_5(\mathbf{x}) - \psi_7(\mathbf{x})) & (\alpha = (1, 0, 1)) \\ A(\boldsymbol{\theta}^{001}, \omega) (\psi_1(\mathbf{x}) - \psi_4(\mathbf{x}) - \psi_6(\mathbf{x}) + \psi_8(\mathbf{x}) \\ \quad + (-\psi_2(\mathbf{x}) + \psi_3(\mathbf{x}) + \psi_5(\mathbf{x}) - \psi_7(\mathbf{x})) & (\alpha = (0, 0, 1)) \\ A(\boldsymbol{\theta}^{110}, \omega) (\psi_1(\mathbf{x}) - \psi_4(\mathbf{x}) - \psi_6(\mathbf{x}) + \psi_8(\mathbf{x}) \\ \quad + (\psi_2(\mathbf{x}) - \psi_3(\mathbf{x}) - \psi_5(\mathbf{x}) + \psi_7(\mathbf{x})) & (\alpha = (1, 1, 0)) \end{cases} \quad (5.23)$$

for $\boldsymbol{\theta} = \boldsymbol{\theta}^{000} \in \Theta_{\text{low}} = (-\pi/2, \pi/2]^3$. Due to

$$\begin{aligned}
 \psi_1(\mathbf{x}) + \psi_4(\mathbf{x}) + \psi_6(\mathbf{x}) + \psi_8(\mathbf{x}) &= \frac{1}{2} (\varphi_h(\boldsymbol{\theta}^{000}, \mathbf{x}) + \varphi_h(\boldsymbol{\theta}^{111}, \mathbf{x})) \\
 \psi_1(\mathbf{x}) + \psi_4(\mathbf{x}) - \psi_6(\mathbf{x}) - \psi_8(\mathbf{x}) &= \frac{1}{2} (\varphi_h(\boldsymbol{\theta}^{100}, \mathbf{x}) + \varphi_h(\boldsymbol{\theta}^{011}, \mathbf{x})) \\
 \psi_1(\mathbf{x}) - \psi_4(\mathbf{x}) + \psi_6(\mathbf{x}) - \psi_8(\mathbf{x}) &= \frac{1}{2} (\varphi_h(\boldsymbol{\theta}^{010}, \mathbf{x}) + \varphi_h(\boldsymbol{\theta}^{101}, \mathbf{x})) \\
 \psi_1(\mathbf{x}) - \psi_4(\mathbf{x}) - \psi_6(\mathbf{x}) + \psi_8(\mathbf{x}) &= \frac{1}{2} (\varphi_h(\boldsymbol{\theta}^{001}, \mathbf{x}) + \varphi_h(\boldsymbol{\theta}^{110}, \mathbf{x})) \\
 \psi_2(\mathbf{x}) + \psi_3(\mathbf{x}) + \psi_5(\mathbf{x}) + \psi_7(\mathbf{x}) &= \frac{1}{2} (\varphi_h(\boldsymbol{\theta}^{000}, \mathbf{x}) - \varphi_h(\boldsymbol{\theta}^{111}, \mathbf{x})) \\
 -\psi_2(\mathbf{x}) - \psi_3(\mathbf{x}) + \psi_5(\mathbf{x}) + \psi_7(\mathbf{x}) &= \frac{1}{2} (\varphi_h(\boldsymbol{\theta}^{100}, \mathbf{x}) - \varphi_h(\boldsymbol{\theta}^{011}, \mathbf{x})) \\
 -\psi_2(\mathbf{x}) + \psi_3(\mathbf{x}) - \psi_5(\mathbf{x}) + \psi_7(\mathbf{x}) &= \frac{1}{2} (\varphi_h(\boldsymbol{\theta}^{010}, \mathbf{x}) - \varphi_h(\boldsymbol{\theta}^{101}, \mathbf{x})) \\
 -\psi_2(\mathbf{x}) + \psi_3(\mathbf{x}) + \psi_5(\mathbf{x}) - \psi_7(\mathbf{x}) &= \frac{1}{2} (\varphi_h(\boldsymbol{\theta}^{001}, \mathbf{x}) - \varphi_h(\boldsymbol{\theta}^{110}, \mathbf{x}))
 \end{aligned}$$

(5.23) can be simplified, yielding

$$S_h^R(\omega) \varphi(\boldsymbol{\theta}^\alpha, \mathbf{x}) = \frac{1}{2} \left\{ \begin{array}{ll} A(\boldsymbol{\theta}^{000}, \omega) (\varphi_h(\boldsymbol{\theta}^{000}, \mathbf{x}) + \varphi_h(\boldsymbol{\theta}^{111}, \mathbf{x})) \\ \quad + (\varphi_h(\boldsymbol{\theta}^{000}, \mathbf{x}) + \varphi_h(\boldsymbol{\theta}^{111}, \mathbf{x})) & \text{for } \boldsymbol{\alpha} = (0, 0, 0) \\ A(\boldsymbol{\theta}^{111}, \omega) (\varphi_h(\boldsymbol{\theta}^{000}, \mathbf{x}) + \varphi_h(\boldsymbol{\theta}^{111}, \mathbf{x})) \\ \quad - (\varphi_h(\boldsymbol{\theta}^{000}, \mathbf{x}) - \varphi_h(\boldsymbol{\theta}^{111}, \mathbf{x})) & \text{for } \boldsymbol{\alpha} = (1, 1, 1) \\ A(\boldsymbol{\theta}^{100}, \omega) (\varphi_h(\boldsymbol{\theta}^{100}, \mathbf{x}) + \varphi_h(\boldsymbol{\theta}^{011}, \mathbf{x})) \\ \quad + (\varphi_h(\boldsymbol{\theta}^{100}, \mathbf{x}) - \varphi_h(\boldsymbol{\theta}^{011}, \mathbf{x})) & \text{for } \boldsymbol{\alpha} = (1, 0, 0) \\ A(\boldsymbol{\theta}^{011}, \omega) (\varphi_h(\boldsymbol{\theta}^{100}, \mathbf{x}) + \varphi_h(\boldsymbol{\theta}^{011}, \mathbf{x})) \\ \quad - (\varphi_h(\boldsymbol{\theta}^{100}, \mathbf{x}) - \varphi_h(\boldsymbol{\theta}^{011}, \mathbf{x})) & \text{for } \boldsymbol{\alpha} = (0, 1, 1) \\ A(\boldsymbol{\theta}^{010}, \omega) (\varphi_h(\boldsymbol{\theta}^{010}, \mathbf{x}) + \varphi_h(\boldsymbol{\theta}^{101}, \mathbf{x})) \\ \quad + (\varphi_h(\boldsymbol{\theta}^{010}, \mathbf{x}) - \varphi_h(\boldsymbol{\theta}^{101}, \mathbf{x})) & \text{for } \boldsymbol{\alpha} = (0, 1, 0) \\ A(\boldsymbol{\theta}^{101}, \omega) (\varphi_h(\boldsymbol{\theta}^{010}, \mathbf{x}) + \varphi_h(\boldsymbol{\theta}^{101}, \mathbf{x})) \\ \quad - (\varphi_h(\boldsymbol{\theta}^{010}, \mathbf{x}) - \varphi_h(\boldsymbol{\theta}^{101}, \mathbf{x})) & \text{for } \boldsymbol{\alpha} = (1, 0, 1) \\ A(\boldsymbol{\theta}^{001}, \omega) (\varphi_h(\boldsymbol{\theta}^{001}, \mathbf{x}) + \varphi_h(\boldsymbol{\theta}^{110}, \mathbf{x})) \\ \quad + (\varphi_h(\boldsymbol{\theta}^{001}, \mathbf{x}) + \varphi_h(\boldsymbol{\theta}^{110}, \mathbf{x})) & \text{for } \boldsymbol{\alpha} = (0, 0, 1) \\ A(\boldsymbol{\theta}^{110}, \omega) (\varphi_h(\boldsymbol{\theta}^{001}, \mathbf{x}) + \varphi_h(\boldsymbol{\theta}^{110}, \mathbf{x})) \\ \quad - (\varphi_h(\boldsymbol{\theta}^{001}, \mathbf{x}) - \varphi_h(\boldsymbol{\theta}^{110}, \mathbf{x})) & \text{for } \boldsymbol{\alpha} = (1, 1, 0). \end{array} \right. \quad (5.24)$$

Thus, $S_h^R(\omega) : \mathcal{F}_{2h}(\boldsymbol{\theta}) \rightarrow \mathcal{F}_{2h}(\boldsymbol{\theta})$ for $\boldsymbol{\theta} \in \Theta_{\text{low}} = (-\pi/2, \pi/2]^3$ holds and

(5.24) yields $S_h^R(\omega)|_{\mathcal{F}_{2h}(\boldsymbol{\theta})} = \tilde{S}_h^R(\boldsymbol{\theta}, \omega) \in \mathbb{C}^{8 \times 8}$ with

$$\tilde{S}_h^R(\boldsymbol{\theta}, \omega) = \frac{1}{2} \begin{pmatrix} A^{000} + 1 & A^{111} - 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ A^{000} - 1 & A^{111} + 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & A^{100} + 1 & A^{011} - 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & A^{100} - 1 & A^{011} + 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & A^{010} + 1 & A^{101} - 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & A^{010} - 1 & A^{101} + 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & A^{001} + 1 & A^{110} - 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & A^{001} - 1 & A^{110} + 1 \end{pmatrix}.$$

Due to space limitations we use the abbreviation $A^\alpha := A(\boldsymbol{\theta}^\alpha, \omega)$ as in the two-dimensional case discussed in Example 5.10. Similarly, it can be shown that $S_h^B(\omega) : \mathcal{F}_{2h}(\boldsymbol{\theta}) \rightarrow \mathcal{F}_{2h}(\boldsymbol{\theta})$ for $\boldsymbol{\theta} \in \Theta_{\text{low}}$ holds and that $S_h^B(\omega)|_{\mathcal{F}_{2h}(\boldsymbol{\theta})} = \tilde{S}_h^B(\boldsymbol{\theta}, \omega) \in \mathbb{C}^{8 \times 8}$ is given by

$$\frac{1}{2} \begin{pmatrix} A^{000} + 1 & -A^{111} + 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -A^{000} + 1 & A^{111} + 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & A^{100} + 1 & -A^{011} + 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -A^{100} + 1 & A^{011} + 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & A^{010} + 1 & -A^{101} + 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -A^{010} + 1 & A^{101} + 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & A^{001} + 1 & -A^{110} + 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -A^{001} + 1 & A^{110} + 1 \end{pmatrix}.$$

By $\tilde{S}_h^{\text{RB}}(\boldsymbol{\theta}, \omega) = \tilde{S}_h^B(\boldsymbol{\theta}, \omega) \cdot \tilde{S}_h^R(\boldsymbol{\theta}, \omega) \in \mathbb{C}^{8 \times 8}$ one obtains the Fourier representation of RB-JAC point relaxation w.r.t. the $2h$ -harmonics. The Fourier representation of *line* and *plane* RB-JAC relaxations can be computed analogously; see Appendix A.2.

Remark 5.6 (Minimal invariant subspaces for pattern relaxations).

The minimal invariant subspaces of Fourier components for red-black type relaxations are two-dimensional, see [36, 60, 62, 73]. In the case of point RB-JAC, it is already indicated by the distribution of zero entries in $\tilde{S}_h^R(\boldsymbol{\theta}, \omega)$ and $\tilde{S}_h^B(\boldsymbol{\theta}, \omega)$ that they can be decomposed further. However, as the minimal invariant subspaces differ for different variants of point/block RB-JAC relaxation it is convenient to stay with the $2h$ -harmonics for a uniform treatment, especially in connection with Fourier k -grid analysis from Chapter 6.

Smoothing factors for point/block RB-JAC relaxations applied to the d -dimensional anisotropic diffusion equation (see Sections 4.1.1 and 4.2.1 for the two- and three-dimensional case, respectively) are derived analytically in [73]. This analysis is based on Fourier representations w.r.t. the minimal invariant subspaces. ►

5.4.5 General definition of the smoothing factor

In order to measure the smoothing properties of pattern relaxation methods we adopt the general definition of the smoothing factor from [60]. Here, the *real* coarse-grid correction for a two-grid method (see Section 3.1) is replaced by an *ideal* coarse-grid correction operator Q_h^H which annihilates the

low-frequency error components and leaves the high-frequency components unchanged [60]. Q_h^H is a projection operator onto the space of high-frequency components

$$\mathcal{F}_{\text{high}} := \{\varphi_h(\boldsymbol{\theta}, \cdot) \mid \boldsymbol{\theta} \in \Theta_{\text{high}}\}.$$

The space of low-frequency components is then evidently given by $\mathcal{F}_{\text{low}} := \{\varphi_h(\boldsymbol{\theta}, \cdot) \mid \boldsymbol{\theta} \in \Theta_{\text{low}}\}$. Thus, for a general coarsening strategy Q_h^H is defined by

$$Q_h^H \varphi_h(\boldsymbol{\theta}, \cdot) := \begin{cases} \varphi_h(\boldsymbol{\theta}, \cdot) & \text{for } \varphi_h(\boldsymbol{\theta}, \cdot) \in \mathcal{F}_{\text{high}} \\ 0 & \text{for } \varphi_h(\boldsymbol{\theta}, \cdot) \in \mathcal{F}_{\text{low}} \end{cases}. \quad (5.25)$$

The Fourier representation of Q_h^H using the $2h$ -harmonics is particularly simple. It is given by $(2^d \times 2^d)$ diagonal matrices, since the Fourier components are eigenfunctions of Q_h^H .

Example 5.11 (\tilde{Q}_h^H for two-dimensional problems). Standard, x_1 -, and x_2 -semicoarsening in two dimensions lead to the following Fourier representations of the related ideal coarse-grid correction operator assuming order (5.16) for the multiindex α :

$$\tilde{Q}_h^{2h} = \text{diag}\{0, 1, 1, 1\}, \quad \tilde{Q}_h^{2h,h} = \text{diag}\{0, 1, 1, 0\}, \quad \tilde{Q}_h^{h,2h} = \text{diag}\{0, 1, 0, 1\}.$$

That is,

$$\begin{aligned} \tilde{Q}_h^{2h} : \mathcal{F}_{2h}(\boldsymbol{\theta}) &\longrightarrow \text{span} \{ \varphi_h(\boldsymbol{\theta}^{11}, \cdot), \varphi_h(\boldsymbol{\theta}^{10}, \cdot), \varphi_h(\boldsymbol{\theta}^{01}, \cdot) \}, \\ \tilde{Q}_h^{2h,h} : \mathcal{F}_{2h}(\boldsymbol{\theta}) &\longrightarrow \text{span} \{ \varphi_h(\boldsymbol{\theta}^{11}, \cdot), \varphi_h(\boldsymbol{\theta}^{10}, \cdot) \}, \\ \tilde{Q}_h^{h,2h} : \mathcal{F}_{2h}(\boldsymbol{\theta}) &\longrightarrow \text{span} \{ \varphi_h(\boldsymbol{\theta}^{11}, \cdot), \varphi_h(\boldsymbol{\theta}^{01}, \cdot) \} \end{aligned}$$

for $\boldsymbol{\theta} \in \Theta_{2h}$. ◀

Some representations in the three-dimensional case are given in the next example.

Example 5.12 (\tilde{Q}_h^H for three-dimensional problems). Using the same order for the multiindex α as in Section 5.4.3, we obtain the following Fourier representations for Q_h^H based on the $2h$ -harmonics, shown in Table 5.5. ◀

The representation of ω -JAC and ω -GS-LEX w.r.t. the $2h$ -harmonics is evidently given by $(2^d \times 2^d)$ diagonal matrices. Using the Fourier symbols from (5.7), this reads in two and three dimensions as

$$\begin{aligned} S_h(\omega)|_{\mathcal{F}_{2h}(\boldsymbol{\theta})} &=: \tilde{S}_h(\boldsymbol{\theta}, \omega) = \text{diag}\{A(\boldsymbol{\theta}^{00}, \omega), A(\boldsymbol{\theta}^{11}, \omega), A(\boldsymbol{\theta}^{10}, \omega), A(\boldsymbol{\theta}^{01}, \omega)\}, \\ S_h(\omega)|_{\mathcal{F}_{2h}(\boldsymbol{\theta})} &=: \tilde{S}_h(\boldsymbol{\theta}, \omega) = \text{diag}\{A(\boldsymbol{\theta}^{000}, \omega), A(\boldsymbol{\theta}^{111}, \omega), A(\boldsymbol{\theta}^{100}, \omega), A(\boldsymbol{\theta}^{011}, \omega), \\ &\quad A(\boldsymbol{\theta}^{010}, \omega), A(\boldsymbol{\theta}^{101}, \omega), A(\boldsymbol{\theta}^{001}, \omega), A(\boldsymbol{\theta}^{110}, \omega)\}, \end{aligned} \quad (5.26)$$

TABLE 5.5: Fourier representation of Q_h^H .

coarsening strategy	Fourier symbol of Q_h^H
standard coarsening	$\tilde{Q}_h^{2h} = \text{diag}\{0, 1, 1, 1, 1, 1, 1, 1\}$
(x_1-x_2) -coarsening	$\tilde{Q}_h^H = \text{diag}\{0, 1, 1, 1, 1, 1, 0, 1\}$
(x_1-x_3) -coarsening	$\tilde{Q}_h^H = \text{diag}\{0, 1, 1, 1, 0, 1, 1, 1\}$
(x_2-x_3) -coarsening	$\tilde{Q}_h^H = \text{diag}\{0, 1, 0, 1, 1, 1, 1, 1\}$
x_1 -coarsening	$\tilde{Q}_h^H = \text{diag}\{0, 1, 1, 0, 0, 1, 0, 1\}$
x_2 -coarsening	$\tilde{Q}_h^H = \text{diag}\{0, 1, 0, 1, 1, 0, 0, 1\}$
x_3 -coarsening	$\tilde{Q}_h^H = \text{diag}\{0, 1, 0, 1, 0, 1, 1, 0\}$

respectively. Note, that in connection with Fourier two-grid ($2g$) analysis presented in the following chapter, we prefer the notation $S_h(\omega)|_{\mathcal{F}_{2h}(\boldsymbol{\theta})} =: S^{2g}(\boldsymbol{\theta})$.

Now, we are able to give a general definition for the smoothing factor applying the Fourier representations of $S_h(\omega)$ and Q_h^H with respect to the $2h$ -harmonics.

Definition 5.9 (General definition of the smoothing factor): Definition 5.6 of $\rho_1(\omega)$ is generalized to relaxation methods $S_h(\omega)$ which leave the spaces of $2h$ -harmonics invariant in the following way [5, 60]:

$$\rho_1(\nu, \omega) := \sup_{\boldsymbol{\theta} \in \Theta_{\text{low}}} \left\{ \sqrt[\nu]{\rho \left(\tilde{Q}_h^H \tilde{S}_h^\nu(\boldsymbol{\theta}, \omega) \right)} \right\} \quad (5.27)$$

where $\nu = \nu_1 + \nu_2$ denotes the sum of pre- and postsmoothing steps.

Definition 5.7 referring to Dirichlet boundary conditions has to be adapted accordingly leading to $\rho_1^D(\nu, \omega)$. For example, for two-dimensional applications we have

$$\rho_1^D(\nu, \omega) := \max_{\boldsymbol{\theta} \in \Theta_{\text{low}}^D} \left\{ \sqrt[\nu]{\rho \left(\tilde{D}(\boldsymbol{\theta}) \tilde{Q}_h^H \tilde{S}_h^\nu(\boldsymbol{\theta}, \omega) \right)} \right\}$$

with $\tilde{D}(\boldsymbol{\theta}) = \text{diag}\{d(\boldsymbol{\theta}^{00}), 1, d(\boldsymbol{\theta}^{10}), d(\boldsymbol{\theta}^{01})\}$ where $d(\boldsymbol{\theta}^{00}) = 0$ if $\theta_1^{00} = 0$ and/or $\theta_2^{00} = 0$, and $d(\boldsymbol{\theta}^{00}) = 1$ otherwise; $d(\boldsymbol{\theta}^{10}) = 0$ if $\theta_2^{00} = 0$, and $d(\boldsymbol{\theta}^{10}) = 1$ otherwise; finally $d(\boldsymbol{\theta}^{01}) = 0$ if $\theta_1^{00} = 0$, and $d(\boldsymbol{\theta}^{01}) = 1$ otherwise [65]. The generalization to higher dimensions is straightforward. ω and ν will be skipped if they equal one. \blacktriangle

It should be pointed out that the explicit calculation of $\rho_1(\nu, \omega)$ by the Fourier analysis software is again based on the finite dimensional Fourier space

yielding $\rho_1^P(\nu, \omega)$; compare with Remarks 5.2 and 5.4. The main difference between Definition 5.6/Definition 5.7 and Definition 5.9 is that the smoothing factor from Definition 5.9 depends on ν . This is due to the fact that low- and high-frequency components are intermixed by general relaxation methods. However, if the Fourier components are eigenfunctions of the relaxation (as it is the case for JAC and GS-LEX) then the two definitions of the smoothing factor coincide due to the diagonal structure of $\tilde{S}_h(\theta, \omega)$ (5.26).

In the accompanying software,

$$\rho(S^\nu Q) := \left(\rho_1^P(\nu, \omega)\right)^\nu \quad \text{and} \quad \rho(S^\nu Q)_D := \left(\rho_1^D(\nu, \omega)\right)^\nu \quad (5.28)$$

are displayed. The smoothing factors for RB-JAC point relaxation applied to the three-dimensional Laplacian are shown in Figure 5.6.

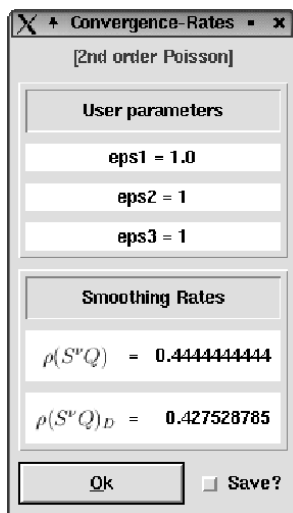


FIGURE 5.6: Smoothing factors for point RB-JAC relaxation applied to the three-dimensional Poisson equation, $h = 1/32$, $\nu = 1$.

Remark 5.7 (Ideal versus real coarse-grid correction). The smoothing factor should be understood as an indication of which two-grid or multigrid convergence factors can be expected if a “proper” treatment of the coarse-grid correction is performed. Then, $(\rho_1(\nu, \omega))^\nu$ yields a reasonable approximation for the convergence factor of the related multigrid cycle involving $\nu = \nu_1 + \nu_2$ pre- and postrelaxation steps. In this respect, Q_h^H can be seen as an ideal coarse-grid correction. However, the expression “ideal” is somewhat misleading. It should not imply that the smoothing factor is a lower

bound for the two- or multigrid convergence factor which is not true in general. A famous counterexample involves point GS-LEX relaxation. The application of a multigrid method—consisting of one point GS-LEX relaxation, full weighting of residuals, bilinear interpolation, and coarse-grid analogs of the fine-grid discretization—to the second-order central discretization of the two-dimensional Laplacian yields smoothing factor $\rho_1 = 0.5$ and a two-grid convergence factor of 0.4. ►

Remark 5.8 (Variable coefficients). Dealing with operators $L_h(\mathbf{x})$ that are characterized by variable coefficients prevents a direct application of the Fourier analysis. However, local Fourier analysis can be applied to the locally frozen operator at a fixed grid point ξ . Replacing the variable \mathbf{x} by a constant ξ , one obtains an operator $L_h(\xi)$ with *constant* frozen coefficients.

Properties of the frozen operators carry over to $L_h(\mathbf{x})$ if certain conditions (for instance sufficiently smooth coefficients [8]) are fulfilled. For example, Hackbusch's smoothing property can be retained by $L_h(\mathbf{x})$ if it holds for the locally frozen operators [28]. In [7, 8, 10] it is motivated that the smoothing factor for $L_h(\mathbf{x})$ can be bounded by the supremum over the smoothing factors for the locally frozen operators, i.e.,

$$\rho_1(\nu, \omega; L_h(\mathbf{x})) := \sup_{\xi \in \Omega} \rho_1(\nu, \omega; L_h(\xi)). \quad (5.29)$$

In the case of Dirichlet boundary conditions one may define $\rho_1^D(\nu, \omega; L_h(\mathbf{x})) := \sup_{\xi \in \Omega} \rho_1^D(\nu, \omega; L_h(\xi))$ following Definition 5.7. For an explicit calculation, one usually approximates the smoothing factor by $\max_{\xi \in \Omega_h} \rho_1(\nu, \omega; L_h(\xi))$ or $\max_{\xi \in \Omega_h} \rho_1^D(\nu, \omega; L_h(\xi))$. Examples are given in Section 4.1.7. ►

Remark 5.9 (Analytical formulas for smoothing factors). For certain operators like the Laplacian [43, 67, 73, 74, 76] it is possible to derive analytical formulas for the related smoothing factors even for RB-JAC relaxations. However, for most applications one has to rely on an appropriate software to perform Fourier smoothing analysis. ►

5.4.6 Red-black Gauss-Seidel (RB-GS) relaxations

Another type of pattern relaxation is red-black Gauss-Seidel (RB-GS) relaxation. It consists of two partial sweeps governed by a distinction between red and black points, in the same way as it has been discussed for RB-JAC relaxations. As a consequence, the Fourier representations $S_h(\omega)|_{\mathcal{F}(\theta)}$ for RB-GS relaxations have the same structure as the related RB-JAC variant. That is, they share the same distribution of $A(\theta^\alpha, \omega)$ -entries. However for RB-GS, each half-step is processed in a Gauss-Seidel-type manner, i.e., the unknowns within each half-step are numbered lexicographically and they are updated

dynamically. Hence, the calculation of $A(\theta^\alpha, \omega)$ is usually based on a different splitting of the discretization operator L_h in case of RB-GS smoothers compared to their RB-JAC counterpart.

Example 5.13 (RB-GS point relaxation, $d = 2$). RB-GS point relaxation is characterized by the same choice of G_h^R and G_h^B as for RB-JAC point relaxation (see Table 5.1) and the following splitting of the index set J within each half-sweep:

$$\begin{aligned} J^+ &= (\{\kappa \in J \mid \kappa_2 < 0\} \cup \{\kappa \in J \mid \kappa_2 = 0 \text{ and } \kappa_1 < 0\}) \\ &\quad \cap \{\kappa \in J \mid \kappa_1 + \kappa_2 \text{ even}\} \\ J^0 &= \{(0, 0)\}, \quad J^- = J \setminus (J^+ \cup J^0). \end{aligned} \quad (5.30)$$

Obviously, $J^+ \neq \emptyset$ for general applications in contrast to RB-JAC-type methods. An exception is given in the following remark. ◀

Remark 5.10 (RB-GS versus RB-JAC point relaxation). In two dimensions RB-GS point relaxation coincides with RB-JAC point relaxation for 5-point discretizations like Δ_h for the Laplacian. This is due to the fact that $J^+ = \emptyset$ for such a stencil, compare with (5.30). However, this equivalence is no longer valid for discrete operators based on “larger” stencils. As a consequence, RB-JAC and RB-GS behave differently in general. For example, consider the compact 9-point Mehrstellen discretization of the Laplacian from Section 4.1.3 with $h = 1/64$. The corresponding smoothing factors of one step of RB-JAC and RB-GS point relaxation are $\rho_1^P = 0.160$ and $\rho_1^P = 0.246$, respectively. ▶

5.4.7 Multicolor relaxations

Analogous to RB-JAC or RB-GS relaxation which can be considered as *two-color* relaxations one might define a *multi-* or *m-color* (MC) relaxation. Here, the computational grid is subdivided into m types of grid points G_h^1, \dots, G_h^m where each type is associated with a certain color. One complete step of m -color relaxation consists of m partial sweeps: In the j -th partial sweep ($j = 1, \dots, m$) only the unknowns located at $\mathbf{x} \in G_h^j$ are changed using the already updated values from the previous $(j - 1)$ partial sweeps. A lexicographic numbering of grid points and a dynamic update of unknowns within each color yield multicolor Gauss-Seidel (MC-GS) relaxation. If the partial sweeps are performed in a Jacobi-type manner (that means that there is *no* dynamic update of unknowns within each color) MC-JAC relaxation results.

Remark 5.11 (4C-GS versus 4C-JAC point relaxation). A proper choice of subgrids for four-color point relaxation in two dimensions reads

$$G_h^1 = G_h^{00}, \quad G_h^2 = G_h^{11}, \quad G_h^3 = G_h^{10}, \quad G_h^4 = G_h^{01};$$

compare with Table 5.4. The related 4C-JAC and 4C-GS point relaxations are equivalent for two-dimensional operators represented by compact 9-point stencils, see also Remark 5.10, where the relation between RB-JAC and RB-GS for 5-point stencils is discussed. This observation is an important issue when thinking about a parallelization of 4C-GS: all unknowns located at grid points with the same color can be updated simultaneously and independently of the unknowns associated with another color. The Fourier representation of the above 4C-JAC point relaxation is given in [59].

Note that for the three-dimensional variant of the Mehrstellen discretization of the Laplacian—represented by a 19-point stencil, see Section 4.2.3—it is also possible to construct a parallel 4C-GS relaxation by an application of the following subgrids:

$$G_h^1 = G_h^{000} \cup G_h^{001}, \quad G_h^2 = G_h^{111} \cup G_h^{110}, \quad G_h^3 = G_h^{100} \cup G_h^{101}, \quad G_h^4 = G_h^{010} \cup G_h^{011};$$

see Table 5.3 for the definition of G_h^α . ▶

5.5 SMOOTHING ANALYSIS FOR SYSTEMS

Fourier smoothing analysis can be generalized to systems of q PDEs in a straightforward manner by applying the vector-valued Fourier components $\varphi_h(\theta, \cdot)$ from Section 5.1.2. The related spaces of $2h$ -harmonics, $\mathcal{F}_{2h}(\theta)$ with $\theta \in \Theta_{2h}$, are defined analogously to Definition 5.8, replacing the scalar Fourier components by its vector-valued counterparts.

Standard relaxation methods of JAC-, GS-LEX, and RB-JAC-type are covered in Section 5.5.1. Section 5.5.2 is dedicated to another important class of smoothing methods which are particularly designed for systems of equations, the distributive relaxations.

5.5.1 Collective versus decoupled smoothing

The scalar operators $L_h^{i,j}$ ($i, j = 1, \dots, q$) which constitute the system \mathbf{L}_h (1.11) are divided as in (5.5),

$$L_h^{i,j} = (L_h^+)^{i,j} + (L_h^0)^{i,j} + (L_h^-)^{i,j}, \quad (5.31)$$

governed by the particular relaxation at hand, for example, by Jacobi point relaxation. Recall the distinction between “collective” (or “coupled”) relaxation and “decoupled” relaxation from Section 3.5.2. Using expression (5.31), it is possible to define a splitting of the operator \mathbf{L}_h into $\mathbf{L}_h = \mathbf{L}_h^+ + \mathbf{L}_h^0 + \mathbf{L}_h^-$ for both types.

We start with collective (COL) relaxations, i.e., all q difference equations are updated simultaneously at each grid point \mathbf{x} . Then, the Fourier symbols

of $\mathbf{L}_h^{+/0/-}$ are simply given by

$$\tilde{\mathbf{L}}_h^{+/0/-}(\boldsymbol{\theta}) := \begin{pmatrix} \left(\tilde{L}_h^{+/0/-}(\boldsymbol{\theta})\right)^{1,1} & \dots & \left(\tilde{L}_h^{+/0/-}(\boldsymbol{\theta})\right)^{1,q} \\ \vdots & & \vdots \\ \left(\tilde{L}_h^{+/0/-}(\boldsymbol{\theta})\right)^{q,1} & \dots & \left(\tilde{L}_h^{+/0/-}(\boldsymbol{\theta})\right)^{q,q} \end{pmatrix}.$$

In this respect, collective relaxation can be considered as a straightforward generalization of the scalar case.

The situation is somewhat more complex for decoupled smoothing, where we distinguish two different variants consisting of an inner and an outer sweep. The first variant (DEC1) consists of one outer sweep over the grid points, where at a fixed grid point \mathbf{x}^* the unknowns $u_h^1(\mathbf{x}^*), \dots, u_h^q(\mathbf{x}^*)$ are updated one after the other (inner sweep). For this variant, $\tilde{\mathbf{L}}_h^{+/0/-}(\boldsymbol{\theta}) = (\ell_{i,j}^{+/0/-})_{i,j=1,\dots,q}$ reads

$$\begin{aligned} \ell_{i,j}^+ &= \begin{cases} \left(\tilde{L}_h^+(\boldsymbol{\theta})\right)^{i,j} & \text{for } i \leq j \\ \left(\tilde{L}_h^+(\boldsymbol{\theta})\right)^{i,j} + \left(\tilde{L}_h^0(\boldsymbol{\theta})\right)^{i,j} & \text{for } i > j \end{cases} \\ \ell_{i,j}^0 &= \begin{cases} \left(\tilde{L}_h^0(\boldsymbol{\theta})\right)^{i,j} & \text{for } i = j \\ 0 & \text{for } i \neq j \end{cases} \\ \ell_{i,j}^- &= \begin{cases} \left(\tilde{L}_h^-(\boldsymbol{\theta})\right)^{i,j} & \text{for } i \geq j \\ \left(\tilde{L}_h^-(\boldsymbol{\theta})\right)^{i,j} + \left(\tilde{L}_h^0(\boldsymbol{\theta})\right)^{i,j} & \text{for } i < j. \end{cases} \end{aligned}$$

In the second variant (DEC2), the discrete unknown functions $u_h^1(\mathbf{x}), \dots, u_h^q(\mathbf{x})$ ($\mathbf{x} \in G_h$) are updated one after the other (outer sweep) involving q sweeps over the grid points (inner sweep). Here, we have

$$\begin{aligned} \ell_{i,j}^+ &= \begin{cases} \left(\tilde{L}_h(\boldsymbol{\theta})\right)^{i,j} & \text{for } i > j \\ \left(\tilde{L}_h^+(\boldsymbol{\theta})\right)^{i,j} & \text{for } i = j \\ 0 & \text{for } i < j \end{cases} \\ \ell_{i,j}^0 &= \begin{cases} \left(\tilde{L}_h^0(\boldsymbol{\theta})\right)^{i,j} & \text{for } i = j \\ 0 & \text{for } i \neq j \end{cases} \\ \ell_{i,j}^- &= \begin{cases} \left(\tilde{L}_h(\boldsymbol{\theta})\right)^{i,j} & \text{for } i < j \\ \left(\tilde{L}_h^-(\boldsymbol{\theta})\right)^{i,j} & \text{for } i = j \\ 0 & \text{for } i > j. \end{cases} \end{aligned} \tag{5.32}$$

Example 5.14 (Decoupled smoothing for a system of two PDEs). DEC1 and DEC2 applied to a system of two PDEs are governed by the following splittings:

$$\begin{aligned}
 \text{DEC1:} \quad \tilde{\mathbf{L}}_h^+(\boldsymbol{\theta}) &= \begin{pmatrix} (\tilde{\mathbf{L}}_h^+(\boldsymbol{\theta}))^{1,1} & (\tilde{\mathbf{L}}_h^+(\boldsymbol{\theta}))^{1,2} \\ (\tilde{\mathbf{L}}_h^+(\boldsymbol{\theta}))^{2,1} + (\tilde{\mathbf{L}}_h^0(\boldsymbol{\theta}))^{2,1} & (\tilde{\mathbf{L}}_h^+(\boldsymbol{\theta}))^{2,2} \end{pmatrix}, \\
 \tilde{\mathbf{L}}_h^0(\boldsymbol{\theta}) &= \begin{pmatrix} (\tilde{\mathbf{L}}_h^0(\boldsymbol{\theta}))^{1,1} & 0 \\ 0 & (\tilde{\mathbf{L}}_h^0(\boldsymbol{\theta}))^{2,2} \end{pmatrix}, \\
 \tilde{\mathbf{L}}_h^-(\boldsymbol{\theta}) &= \begin{pmatrix} (\tilde{\mathbf{L}}_h^-(\boldsymbol{\theta}))^{1,1} & (\tilde{\mathbf{L}}_h^-(\boldsymbol{\theta}))^{1,2} + (\tilde{\mathbf{L}}_h^0(\boldsymbol{\theta}))^{1,2} \\ (\tilde{\mathbf{L}}_h^-(\boldsymbol{\theta}))^{2,1} & (\tilde{\mathbf{L}}_h^-(\boldsymbol{\theta}))^{2,2} \end{pmatrix}. \\
 \text{DEC2:} \quad \tilde{\mathbf{L}}_h^+(\boldsymbol{\theta}) &= \begin{pmatrix} (\tilde{\mathbf{L}}_h^+(\boldsymbol{\theta}))^{1,1} & 0 \\ (\tilde{\mathbf{L}}_h^+(\boldsymbol{\theta}))^{2,1} & (\tilde{\mathbf{L}}_h^+(\boldsymbol{\theta}))^{2,2} \end{pmatrix}, \\
 \tilde{\mathbf{L}}_h^0(\boldsymbol{\theta}) &= \begin{pmatrix} (\tilde{\mathbf{L}}_h^0(\boldsymbol{\theta}))^{1,1} & 0 \\ 0 & (\tilde{\mathbf{L}}_h^0(\boldsymbol{\theta}))^{2,2} \end{pmatrix}, \quad \tilde{\mathbf{L}}_h^-(\boldsymbol{\theta}) = \begin{pmatrix} (\tilde{\mathbf{L}}_h^-(\boldsymbol{\theta}))^{1,1} & (\tilde{\mathbf{L}}_h^-(\boldsymbol{\theta}))^{1,2} \\ 0 & (\tilde{\mathbf{L}}_h^-(\boldsymbol{\theta}))^{2,2} \end{pmatrix}.
 \end{aligned}$$

◀

An application of the above splittings of $\tilde{\mathbf{L}}_h(\boldsymbol{\theta})$ for collective and decoupled smoothing immediately yields the generalization of the Fourier symbol (5.7) for ω -JAC and ω -GS-LEX relaxations:

$$\mathbf{A}(\boldsymbol{\theta}, \omega) = \left(\tilde{\mathbf{L}}_h^0(\boldsymbol{\theta}) + \omega \tilde{\mathbf{L}}_h^+(\boldsymbol{\theta}) \right)^{-1} \left((1 - \omega) \tilde{\mathbf{L}}_h^0(\boldsymbol{\theta}) - \tilde{\mathbf{L}}_h^-(\boldsymbol{\theta}) \right) \in \mathbb{C}^{q \times q}. \quad (5.33)$$

Hence, the Fourier representation of these relaxations w.r.t. $2h$ -harmonics is given by $(2^d q \times 2^d q)$ block-diagonal (bdiag) matrices consisting of 2^d ($q \times q$)-blocks. For example, considering three-dimensional applications one obtains

$$\begin{aligned}
 \tilde{\mathbf{S}}_h(\boldsymbol{\theta}, \omega) &= \text{bdiag} \{ \mathbf{A}(\boldsymbol{\theta}^{000}, \omega), \mathbf{A}(\boldsymbol{\theta}^{111}, \omega), \mathbf{A}(\boldsymbol{\theta}^{100}, \omega), \mathbf{A}(\boldsymbol{\theta}^{011}, \omega), \\
 &\quad \mathbf{A}(\boldsymbol{\theta}^{010}, \omega), \mathbf{A}(\boldsymbol{\theta}^{101}, \omega), \mathbf{A}(\boldsymbol{\theta}^{001}, \omega), \mathbf{A}(\boldsymbol{\theta}^{110}, \omega) \} \in \mathbb{C}^{8q \times 8q}.
 \end{aligned}$$

Recall that the Fourier representation of pattern relaxations for scalar problems is given by *nondiagonal* $(2^d \times 2^d)$ -matrices. These matrices have two kinds of entries: scalar Fourier symbols $A^\alpha = A(\boldsymbol{\theta}^\alpha, \omega)$ and certain constants c , see Section 5.4.4. In the case of systems of PDEs, they have to be replaced by the appropriate $\mathbf{A}^\alpha = \mathbf{A}(\boldsymbol{\theta}^\alpha, \omega)$ from (5.33) and by $c I_q$, respectively. Here, $I_q = \text{diag}\{1, \dots, 1\} \in \mathbb{C}^{q \times q}$ denotes the matrix identity. For instance, the first entry of $\tilde{\mathbf{S}}_h^R(\boldsymbol{\theta})$ from Section 5.4.4 has to be adapted as follows:

$$\frac{1}{2} (\mathbf{A}^{000} + I_q) \in \mathbb{C}^{q \times q}.$$

The resulting Fourier representations $\tilde{\mathbf{S}}_h(\boldsymbol{\theta}, \omega)$ are $(2^d q \times 2^d q)$ -matrices again, but in general they do not have a block-diagonal structure as ω -JAC and ω -GS-LEX relaxations, see above. As in the scalar case, we write $\mathbf{S}_h(\omega)|_{\mathcal{F}_{2h}(\boldsymbol{\theta})} =: \mathbf{S}^{2g}(\boldsymbol{\theta})$ in connection with Fourier two-grid analysis.

The generalization of the ideal coarse-grid correction operator to systems of PDEs is clearly defined by

$$\mathbf{Q}_h^H \varphi_h(\boldsymbol{\theta}, \cdot) := \begin{cases} \varphi_h(\boldsymbol{\theta}, \cdot) & \text{for } \boldsymbol{\theta} \in \Theta_{\text{high}} \\ 0 \cdot \mathbf{I} & \text{for } \boldsymbol{\theta} \in \Theta_{\text{low}} \end{cases} \quad (5.34)$$

with $\varphi_h(\boldsymbol{\theta}, \cdot) = \varphi_h(\boldsymbol{\theta}, \cdot) \cdot \mathbf{I}$ from Section 5.1.2. The related Fourier representation w.r.t. the $2h$ -harmonics is then given by $(2^d q \times 2^d q)$ block-diagonal matrices. For example, in the case of standard coarsening in three dimensions we have

$$\tilde{\mathbf{Q}}_h^{2h} = \text{bdiag} \{ \text{diag}\{0, \dots, 0\}, I_q, I_q, I_q, I_q, I_q, I_q, I_q \}$$

assuming the same order of the multiindex $\boldsymbol{\alpha}$ as in Section 5.4.3.

Now, Definition 5.9 of the smoothing factor can be easily adapted to systems of PDEs by inserting $\tilde{\mathbf{Q}}_h^H$ and $\tilde{\mathbf{S}}_h(\boldsymbol{\theta}, \omega)$ into (5.27) instead of \tilde{Q}_h^H and $\tilde{S}_h(\boldsymbol{\theta}, \omega)$, respectively. The definition of $\rho_1^P(\omega)$ and $\rho_1^D(\omega)$ can be generalized to systems of equations in the same way.

Example 5.15 (Smoothing factors for collective and decoupled relaxation). We consider the first-order flux difference-splitting discretization of the Oseen equations discussed in Section 4.3.3. We set Reynolds number $Re = 1000$, convection angle $\beta = 45^\circ$, and mesh size $h = 1/32$. Applying alternating line (i.e., forward x_1 -line followed by forward x_2 -line) GS-LEX relaxation yields

$$\rho_1^P = 0.264, \quad 0.691, \quad \text{and} \quad 0.782$$

for COL, DEC1, and DEC2, respectively. The collective variant shows the best smoothing properties. On the other hand, it is more expensive than the decoupled relaxations as it involves the solution of a $(q \times q)$ linear system at each grid point due to the simultaneous update of the q equations. The application of alternating line RB-JAC relaxation gives

$$\rho_1^P = 0.279, \quad 0.437, \quad \text{and} \quad 0.267$$

for COL, DEC1, and DEC2. Hence, it is also possible to find a *decoupled* relaxation with satisfactory smoothing properties for the system under consideration. ◀

5.5.2 Distributive relaxation

Distributive relaxations have already been briefly addressed in Section 3.5.2. Originally, they were designed for applications from computational fluid dynamics [10, 71] but recently they have been applied to problems from poroelasticity as well [24, 70]. A distributive relaxation for the discrete system \mathbf{L}_h is constructed as follows. In order to relax $\mathbf{L}_h \mathbf{u}_h = \mathbf{f}_h$, we introduce a new variable \mathbf{v}_h by $\mathbf{u}_h = \mathbf{C}_h \mathbf{v}_h$ and consider the transformed system $\mathbf{L}_h \mathbf{C}_h \mathbf{v}_h = \mathbf{f}_h$.

Ideally (compare with [7]), \mathbf{C}_h is chosen such that the resulting system $\mathbf{L}_h \mathbf{C}_h$ is triangular and the diagonal elements of $\mathbf{L}_h \mathbf{C}_h$ are composed of the factors of $\det(\mathbf{L}_h)$. Then, the resulting transformed system is suited for the second variant of *decoupled* smoothing (compare with (5.32)), i.e., each equation can be treated separately.

Example 5.16 (Distributive relaxation for the Stokes equations). A second-order discretization of the two-dimensional Stokes equations and an appropriate distributor are given by

$$\mathbf{L}_h = \begin{pmatrix} -\Delta_h & 0 & D_h^1 \\ 0 & -\Delta_h & D_h^2 \\ D_h^1 & D_h^2 & -\mu h^2 \Delta_h \end{pmatrix} \quad \text{with } \mu \geq 0 \quad \text{and} \quad \mathbf{C}_h = \begin{pmatrix} I_h & 0 & -D_h^1 \\ 0 & I_h & -D_h^2 \\ 0 & 0 & -\Delta_h \end{pmatrix},$$

respectively, leading to the transformed system

$$\mathbf{L}_h \mathbf{C}_h = \begin{pmatrix} -\Delta_h & 0 & 0 \\ 0 & -\Delta_h & 0 \\ D_h^1 & D_h^2 & -\Delta_{2h} + \mu h^2 \Delta_h^2 \end{pmatrix} \quad \text{with } -\Delta_{2h} \triangleq \frac{1}{4h^2} \begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 4 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix}_h$$

and discrete biharmonic operator Δ_h^2 from Section 4.1.5. We have $\det(\mathbf{L}_h) = -\Delta_h(-\Delta_{2h} + \mu h^2 \Delta_h^2)$. ◀

For an implementation of distributive relaxations it is convenient to consider the *correction* equations

$$\mathbf{L}_h \delta \mathbf{u}_h^{(i+1)} = \mathbf{r}_h^{(i)} \quad \text{and} \quad \mathbf{L}_h \mathbf{C}_h \delta \mathbf{v}_h^{(i+1)} = \mathbf{r}_h^{(i)}$$

with an update $\delta \mathbf{u}_h^{(i+1)} = \mathbf{C}_h \delta \mathbf{v}_h^{(i+1)} = \mathbf{u}_h - \mathbf{u}_h^{(i+1)}$ and the residual $\mathbf{r}_h^{(i)} = \mathbf{L}_h \mathbf{u}_h^{(i)} - \mathbf{f}_h$. $\mathbf{u}_h^{(i)}$ denotes the approximation after the i -th iteration of the exact discrete solution \mathbf{u}_h .

The distributive relaxation consists of two steps. In the first step, a new approximation $\delta \mathbf{v}_h^{(i+1)}$ to the “ghost variable” $\delta \mathbf{v}_h = (\delta v_h^1, \dots, \delta v_h^q)^T$ is calculated. This will be done by a decoupled relaxation, due to the (hopefully) favorable structure of the transformed system. In the second step, a new approximation for \mathbf{u}_h is computed by

$$\mathbf{u}_h^{(i+1)} = \mathbf{u}_h^{(i)} + \delta \mathbf{u}_h^{(i+1)} = \mathbf{u}_h^{(i)} + \mathbf{C}_h \delta \mathbf{v}_h^{(i+1)}.$$

Only the first step matters for the Fourier smoothing analysis. More precisely, the smoothing factor for a distributive relaxation method equals the smoothing factor of the corresponding decoupled relaxation applied to $\mathbf{L}_h \mathbf{C}_h$. The situation simplifies, if the distributor yields—as desired—a triangular

transformed system. Then, the smoothing factor $\rho_1(\nu, \omega; \mathbf{L}_h)$ is simply governed by the *scalar* diagonal blocks of $\mathbf{L}_h \mathbf{C}_h$ [7, 62], i.e.,

$$\rho_1(\nu, \omega; \mathbf{L}_h) = \max_{j=1, \dots, q} \rho_1(\nu, \omega; (L_h C_h)^{j,j}).$$

For the Stokes equations from Example 5.16 this means that

$$\rho_1(\nu, \omega; \mathbf{L}_h) = \max \{ \rho_1(\nu, \omega; -\Delta_h), \rho_1(\nu, \omega; -\Delta_{2h} + \mu h^2 \Delta_h^2) \}.$$

Remark 5.12 (Kaczmarz relaxation). Applying the adjoint of the underlying system as distributor, $\mathbf{C}_h = \mathbf{L}_h^*$, yields a relaxation method on the normal equations known as Kaczmarz relaxation. However, such a choice for the distributor will not lead to a triangular transformed system in general and the resulting smoothing factors are often not satisfactory considering the computational work.

Of course, this type of relaxation can also be applied to scalar equations. For example, the Kaczmarz variant of RB-JAC point relaxation for the *symmetric* discrete Laplacian involves $\Delta_h \Delta_h^* = \Delta_h^2$ yielding a smoothing factor of $\rho_1^P = 0.640$ for the two-dimensional variant, which is much worse than the smoothing factor of RB-JAC point relaxation for the *not transformed* operator ($\rho_1^P = 0.250$). Kaczmarz relaxation is mainly applied to difficult (nonsymmetric) applications where no efficient or even no converging relaxation method is known. For such a situation, it is convenient to apply a Kaczmarz variant of a classical relaxation because it always converges, since $\mathbf{L}_h \mathbf{L}_h^*$ is a symmetric operator; compare with Remark 5.16. ►

5.6 MULTISTAGE (MS) RELAXATIONS

Recall that in Section 3.5.3 we introduced the modification of an arbitrary (undamped) smoothing method $S_h(\omega = 1) = S_h$ to a multistage variant by the following expression:

$$P_m(S_h) := \prod_{n=1}^m ((1 - \omega_n)I_h + \omega_n S_h)$$

with multistage parameters ω_n ($n = 1, \dots, m$). We collect these parameters in an m -tuple $\boldsymbol{\omega} = (\omega_1, \dots, \omega_m)$.

Remark 5.13 (Relation between relaxation and multistage parameters). It can be easily seen that the consecutive application of m point/block ω_n -JAC relaxations ($n = 1, \dots, m$) is equivalent to point/block multistage Jacobi (MS-JAC) relaxation with MS parameters $\omega_1, \dots, \omega_m$ since unknowns

are updated *after* each complete Jacobi sweep. Such a simple relation does not exist for pattern or ω -GS-LEX relaxations. Here, it is not possible to express the relaxation parameters in terms of the multistage parameters. In the multistage variant the overrelaxation is performed *after a complete* relaxation step whereas in the ω -RB-JAC and ω -GS-LEX methods the overrelaxation is applied *dynamically within* the smoothing step.

In principle it is also possible to combine relaxation and multistage parameters, i.e., to construct multistage variants of damped relaxation methods $S_h(\omega)$. ►

The Fourier symbol of a general m -stage method and the corresponding smoothing factor are given by

$$P_m(\tilde{S}_h(\theta)) \quad \text{and} \quad \rho_1(\omega) := \sup_{\theta \in \Theta_{\text{low}}} \rho\left(P_m\left(\tilde{Q}_h^H \tilde{S}_h(\theta)\right)\right) \quad (5.35)$$

due to the diagonal structure of Q_h^H . To construct an optimal multistage relaxation we search the polynomial P_m with coefficients $\omega_1, \dots, \omega_m$, which minimizes the corresponding smoothing factor. This means that one has to solve the following minimization problem:

$$\min_{\omega_n} \sup_{z \in \sigma_S} |P_m(z)| \quad \text{with} \quad \sigma_S = \left\{ \text{spectrum of } \left(\tilde{Q}_h^H \tilde{S}_h(\theta) \right) \mid \theta \in \Theta_{\text{low}} \right\}. \quad (5.36)$$

The situation is particularly transparent, if we assume a nondiverging relaxation S_h having a real-valued spectrum $\sigma_S \subset [S_{\min}, S_{\max}] \subset [-1, 1]$. Then, (5.36) reduces to a classical minimization problem,

$$\min_{\omega_n} \sup_{-1 \leq S_{\min} \leq z \leq S_{\max} \leq 1} |P_m(z)|;$$

see, for example, [63]. Its solution can be found in terms of Chebychev polynomials T_m , defined by

$$T_m(t) = \cos(m \cos^{-1}(t)) \quad \text{for} \quad -1 \leq t \leq 1 \quad \text{and} \quad m \geq 0.$$

$y = (2t - (S_{\max} + S_{\min})) / (S_{\max} - S_{\min})$ is a mapping of $S_{\min} \leq t \leq S_{\max}$ onto $-1 \leq y \leq 1$. The classical solution of (5.36) is then attained by the polynomial

$$P_m(t) = T_m\left(\frac{2t - (S_{\max} + S_{\min})}{S_{\max} - S_{\min}}\right) / T_m\left(\frac{2 - (S_{\max} + S_{\min})}{S_{\max} - S_{\min}}\right).$$

The zeroes of $T_m(t)$ are given by $t_n = \cos((2n-1)\frac{\pi}{2m})$ for $n = 1, \dots, m$. Using t_n , we are able to express the zeroes S_n of the optimal polynomial $P_m(t)$ in terms of S_{\min} and S_{\max} :

$$S_n = \frac{S_{\max} - S_{\min}}{2} \cos\left((2n-1)\frac{\pi}{2m}\right) + \frac{S_{\max} + S_{\min}}{2} \quad \text{for} \quad n = 1, \dots, m. \quad (5.37)$$

From these S_n it is possible to calculate the optimal multistage parameters ω_n and the corresponding smoothing factor $\rho_1(\omega)$, if S_{\min} and S_{\max} are known:

$$\omega_n = \frac{1}{1 - S_n} \quad \text{for } n = 1, \dots, m, \quad \rho_1(\omega) = \left| \prod_{n=1}^m [(1 - \omega_n) + \omega_n S_{\min}] \right|. \quad (5.38)$$

An optimal 1-stage method, for example, is characterized by

$$\omega_1 = \frac{2}{2 - S_{\max} - S_{\min}} \quad \text{and} \quad \rho_1(\omega_1) = \frac{S_{\max} - S_{\min}}{2 - S_{\max} - S_{\min}}. \quad (5.39)$$

Thus, one simply has to calculate expressions for S_{\min} and S_{\max} in order to find an optimal multistage method.

Example 5.17 (Optimal 1-stage JAC relaxation for the Laplacian).

As an example we consider Jacobi point relaxation for the two-dimensional Laplacian. The corresponding Fourier representation is given by

$$\begin{aligned} \tilde{S}_h^{\text{JAC}}(\theta) &= \text{diag}\{A^{00}, A^{11}, A^{10}, A^{01}\} \in \mathbb{C}^{4 \times 4} \quad \text{with} \\ A^\alpha &= A(\theta^\alpha, \omega = 1) = 1 - \frac{4 - 2 \cos(\theta_1^\alpha) - 2 \cos(\theta_2^\alpha)}{4} = \frac{1}{2}(\cos(\theta_1^\alpha) + \cos(\theta_2^\alpha)). \end{aligned}$$

From the above Fourier representation of S_h^{JAC} , we easily obtain

$$\sigma_S = \left[S_{\min} = \frac{1}{2}(\cos(\pi) + \cos(\pi)) = -1, S_{\max} = \frac{1}{2}(\cos(-\pi/2) + \cos(0)) = 1/2 \right].$$

Applying (5.39) yields the well-known optimal, damped Jacobi smoother for the two-dimensional Laplacian:

$$\omega_1 = 4/5 \quad \text{and} \quad \rho_1(\omega_1) = 3/5.$$

Moreover, one can easily calculate the optimal one-stage method for the d -dimensional Laplacian using Example 5.5:

$$\begin{aligned} S_{\min} &= -1, & S_{\max} &= \frac{d-1}{d} & \text{and} \\ \omega_1 &= \frac{2d}{2d+1}, & \rho_1(\omega_1) &= \frac{2d-1}{2d+1}. \end{aligned}$$

◀

Example 5.18 (Optimal 1-stage RB-JAC relaxation). The optimal one-stage parameter for RB-JAC relaxation applied to the two-dimensional Laplacian is given by $\omega_1 = 16/15$ leading to $\rho_1(\omega_1) = 1/5$, whereas for the two-dimensional biharmonic operator we have $\omega_1 = 25/18$ yielding $\rho_1(\omega_1) = 1/2$. Compare with Example 4.3.1 and Proposition 6.6.1 from [67], respectively. ◀

Remark 5.14 (Determination of optimal multistage parameters by `xlfa`). It is interesting to know that S_{\min} and S_{\max} (and thus optimal multistage parameters via (5.38)) can be easily estimated with the help of the accompanying software which provides a plot of the high-frequency spectrum σ_S of the relaxation method under consideration. Figure 5.7 shows the corresponding spectrum of (undamped) JAC point relaxation applied to the two-dimensional Laplacian with $h = 1/128$ yielding the desired information $S_{\min} \approx -1$ and $S_{\max} \approx 1/2$ which is in good agreement with the analytical derivation from Example 5.17. This is a very convenient feature of the soft-

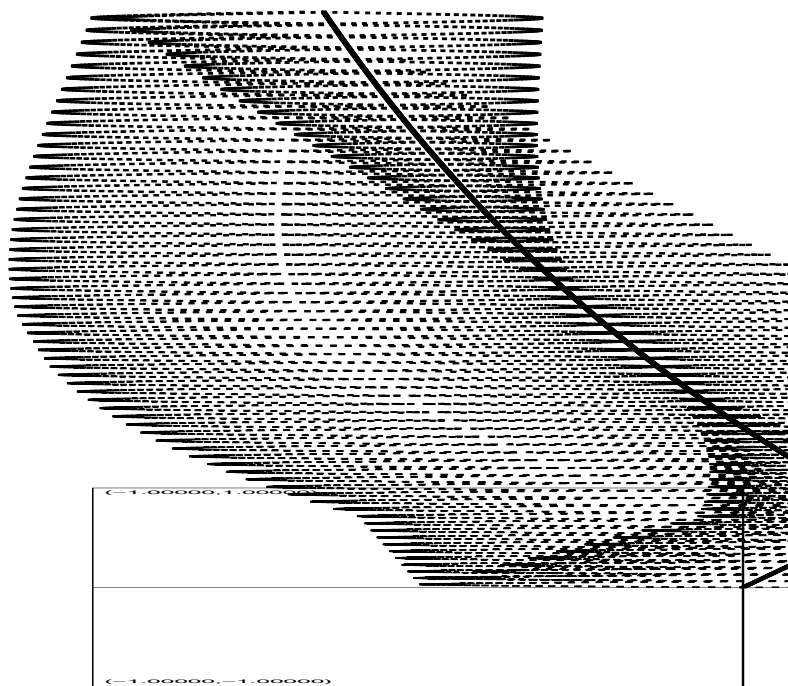


FIGURE 5.7: High-frequency eigenvalue distribution applied to the Poisson equation; $d = 2$.

ware because the usually cumbersome derivation is avoided by a simple mouse click.

Remark 5.15 (Complex Spectrum σ_S). The software also handles complex eigenvalues as well. Then it may be possible to find the exact solution of (5.36). However, for polynomials it is possible to obtain an asymptotic minimization problem [53]. Again, the distribution of the software can be used for the derivation of multistage parameters [69].

The adaptation of the above considerations in the two-dimensional case is straightforward.

5.7 FURTHER RELAXATION METHODS

So far, we provided a detailed discussion of Fourier smoothing analysis for Jacobi, Gauss-Seidel, and pattern relaxations applied to d -dimensional systems of equations including multistage, distributive, and Kaczmarz variants. In this section we give a brief overview on further relaxation methods as well as some pointers to the literature for the corresponding smoothing analysis.

ILU: Relaxations based on an incomplete matrix decomposition into lower and upper triangular parts—known as ILU-type smoothers—represent an important class of smoothing methods, especially for problems with mixed derivatives [60, 62, 72].

ADI: The alternating direction implicit (ADI) iteration is a classical relaxation method that is based on a splitting of the operator under consideration with regard to the different space directions. It has been mainly designed for the solution of (anisotropic) diffusion-type problems (see, for example, [63]) but it might also be applied as a smoother within a multigrid method [28].

Composite relaxation: Composite relaxation schemes [42] are designed for discrete operators that can be written as a product of two discrete operators, i.e., $L_h = L_1 L_2$. A famous example is the biharmonic operator $\Delta_h^2 = \Delta_h \Delta_h$; compare with Sections 4.1.1 and 4.2.1. If efficient relaxation methods are known for each factor L_1 and L_2 , they can be applied within the framework of composite relaxation yielding an efficient smoother for the product operator.

SPAI : Sparse approximate inverses (SPAI) [27] have recently been used as smoothers for multigrid. SPAI algorithms construct a sparse approximate inverse \mathbf{M} of a matrix \mathbf{L} by minimizing $\mathbf{I} - \mathbf{ML}$ in the Frobenius norm. Their smoothing properties are investigated in [14].

Krylov: Another possible choice for smoothing is the application of Krylov subspace methods like GMRES [22, 47]. A smoothing analysis for the indefinite Helmholtz equation is performed in [22].

KAPPA: KAPPA smoothers are certain line-relaxation methods that are especially designed for higher-order upwind discretizations based on the κ -scheme [39] from computational fluid dynamics; compare with Sections 4.1.8 and 4.3.4. They are motivated by the “defect correction approach.” That is, KAPPA smoothers are based on a splitting of the discretization into a “positive” part on the left-hand side and the remaining part on the right-hand side. Positive parts (a positive main diagonal and nonpositive off-diagonal elements) are required in the left-hand side in order to assure that a splitting has smoothing properties. KAPPA smoothers distinguish two different types

of splittings (denoted by “splitting 1” and “splitting 2”). Splitting 2 is implemented in the accompanying software. The design and the related Fourier smoothing analysis of the KAPPA smoothers can be found in [49].

Box: Another popular class of smoothing methods from computational fluid dynamics are so-called box relaxations. The basic idea of box relaxation is to divide the computational grid into certain cells or boxes. Then, a subset of unknowns located in a box are updated simultaneously. For an overview on box smoothing and the related smoothing analysis, we refer to [62].

5.8 THE MEASURE OF h -ELLIPTICITY

In this section, the concept of h -ellipticity is discussed, which is fundamental for the existence of efficient pointwise smoothers; see [6, 7] and especially [10]. More precisely, the “measure of h -ellipticity” is often used to decide whether a certain discretization is appropriate for a multigrid treatment. A “sufficient” amount of h -ellipticity (some form of “ellipticity” in the discretization) indicates that pointwise error-smoothing procedures can be constructed.

Definition 5.10 (Measure of h -Ellipticity): The measure of h -ellipticity of a given operator L_h is defined by

$$E_h(L_h) := \frac{\min \left\{ \left| \tilde{L}_h(\boldsymbol{\theta}) \right| : \boldsymbol{\theta} \in \Theta_{\text{high}} \right\}}{\max \left\{ \left| \tilde{L}_h(\boldsymbol{\theta}) \right| : \boldsymbol{\theta} \in \Theta \right\}} =: \frac{m}{M}.$$

▲

Considering a fixed coarsening strategy, it gives a first impression whether it is possible to construct an efficient point smoother based on a splitting of the discretization operator, $L_h = L_h^+ + L_h^0 + L_h^-$, see (5.6). For simplicity, we consider only undamped relaxations, $\omega = 1$ in (5.6). $E_h(L_h) = 0$ implies that there is a high frequency $\boldsymbol{\theta}^*$ with $\tilde{L}_h^+(\boldsymbol{\theta}^*) + \tilde{L}_h^0(\boldsymbol{\theta}^*) = -\tilde{L}_h^-(\boldsymbol{\theta}^*)$. Assuming $\tilde{L}_h^+(\boldsymbol{\theta}^*) + \tilde{L}_h^0(\boldsymbol{\theta}^*) \neq 0$, this yields

$$\rho_1 = \sup_{\boldsymbol{\theta} \in \Theta_{\text{high}}} \left| \frac{\tilde{L}_h^-(\boldsymbol{\theta})}{\tilde{L}_h^+(\boldsymbol{\theta}) + \tilde{L}_h^0(\boldsymbol{\theta})} \right| \geq \left| \frac{\tilde{L}_h^-(\boldsymbol{\theta}^*)}{\tilde{L}_h^+(\boldsymbol{\theta}^*) + \tilde{L}_h^0(\boldsymbol{\theta}^*)} \right| = 1 \quad (5.40)$$

for the smoothing factor.

On the other hand, a certain amount of h -ellipticity is a sufficient condition for the existence of an efficient point smoother of the kind seen in (5.6) [10, 6]. More precisely, the following theorem holds; see, for example, [41, 62].

Theorem 5.1 (Existence of a Point Smoother).

- If $E_h(L_h) = 0$, it follows that $\rho_1 \geq 1$ for any smoothing method defined by a splitting $L_h = L_h^+ + L_h^0 + L_h^-$ with $\tilde{L}_h^+(\boldsymbol{\theta}) + \tilde{L}_h^0(\boldsymbol{\theta}) \neq 0$ for each $\boldsymbol{\theta} \in \Theta_{\text{high}}$.
- If $E_h(L_h)$ is bounded away from 0 by some constant c (for $h_1, \dots, h_d \rightarrow 0$), there exists a pointwise smoothing procedure S_h with smoothing factor $\rho_1 < 1$ that is bounded away from 1 by some constant that only depends on c .
- Furthermore, if L_h is represented by an “even” [65] or “symmetric” [60, 62] stencil $[\ell_{\boldsymbol{\kappa}}]_h$, i.e.,

$$\ell_{\boldsymbol{\kappa}} = \ell_{\bar{\boldsymbol{\kappa}}} \quad \text{with} \quad \bar{\boldsymbol{\kappa}} = (\kappa_1, \dots, -\kappa_j, \dots, \kappa_d) \quad (j = 1, \dots, d), \quad (5.41)$$

with $\ell_{(0, \dots, 0)} > 0$ and if $\tilde{L}_h(\boldsymbol{\theta}) > 0$ for $\boldsymbol{\theta} \neq (0, \dots, 0)$, then we can construct an optimal ω -JAC point smoother with

$$\omega_{\text{opt}} = \frac{2\ell_{(0, \dots, 0)}}{m + M} \quad \text{and} \quad \rho_1(\omega_{\text{opt}}) = \frac{M - m}{M + m} = \frac{1 - E_h(L_h)}{1 + E_h(L_h)}.$$

Remark 5.16 (Kaczmarz JAC point relaxation) Similarly as in the above theorem, one might construct a Kaczmarz JAC point relaxation with optimal relaxation parameter and corresponding smoothing factor [6] given by

$$\omega_{\text{opt}} = \frac{2\ell_{(0, \dots, 0)}^K}{m^2 + M^2} \quad \text{and} \quad \rho_1(\omega_{\text{opt}}) = \frac{1 - (E_h(L_h))^2}{1 + (E_h(L_h))^2}.$$

Here, $\ell_{(0, \dots, 0)}^K$ denotes the central stencil element of the product operator $L_h L_h^*$ defining the normal equations. As a consequence it is always possible to design a point smoother for an arbitrary *nonsymmetric* discretization operator, as long as it has a sufficient amount of h -ellipticity. ►

In order to illustrate the measure of h -ellipticity, we calculate this quantity for an assortment of the case studies presented in Chapter 4. Comparing the measure of h -ellipticity and its implications with the corresponding multigrid algorithms proposed in Chapter 4 it becomes obvious that this simple and easy-to-calculate quantity already gives some valuable insights into possible difficulties concerning an efficient multigrid treatment. The sequence of test cases starts with an investigation of the anisotropic diffusion equation (Section 5.8.1). Sections 5.8.2 and 5.8.3 are mainly intended for multigrid experts interested in convection-dominated flow problems.

For certain model problems it is often possible to find analytical expressions for the measure of h -ellipticity, see below. However, for general applications including complicated systems of equations it is convenient (and sometimes the only possibility to obtain the desired quantity) to apply numerical software

like **LFA** in order to calculate $E_h(L_h)$. We restrict the considerations to a square grid (i.e., $h = h_1, \dots, h_d$) for ease of notation. The generalization to rectangular grids can be straightforwardly reduced to uniform grids by an appropriate scaling of the coefficients occurring in the discrete operators.

5.8.1 Example 1: anisotropic diffusion equation

We consider the central discretization of the anisotropic diffusion operator

$$L_h = - \sum_{j=1}^d \varepsilon_j D_h^{jj} \quad \text{with} \quad \varepsilon_j \geq 0 \quad (j = 1, \dots, d). \quad (5.42)$$

With respect to the analysis it is more convenient to scale the real positive coefficients ε_j as in [73, 74] and to replace (5.42) by

$$L_h = - \sum_{j=1}^d c_j D_h^{jj} \quad \text{with} \quad c_j = \varepsilon_j / \sum_{j=1}^d \varepsilon_j. \quad (5.43)$$

L_h in (5.43) becomes singularly perturbed, if at least one of the scaled coefficients c_j tends to 0. Due to the scaling we have $\sum_{j=1}^d c_j = 1$ and the Fourier symbols of L_h are given by

$$\tilde{L}_h(\boldsymbol{\theta}) = \frac{2}{h^2} \left[1 - \sum_{j=1}^d c_j \cos(\theta_j) \right] \quad (\boldsymbol{\theta} \in \Theta), \quad (5.44)$$

compare with Example 5.1. Using (5.44), one easily establishes that

$$m = \frac{2c_{\min}}{h^2} \quad \text{and} \quad M = \frac{4}{h^2}$$

with $c_{\min} = \min_{j \in \mathcal{I}=\{1, \dots, d\}} c_j$, yielding the first example of an h -ellipticity measure.

Example 5.19 (h -Ellipticity for the anisotropic diffusion operator). The measure of h -ellipticity for (5.43) assuming standard coarsening is given by

$$E_h(L_h) = \frac{c_{\min}}{2}.$$

◀

Regarding Theorem 5.1, we immediately see that there is no reasonable point smoother for $c_{\min} \ll 1$. $c_{\min} \ll 1$ may occur in the case of strong anisotropies where it is well known (compare with Sections 4.1.1, 4.2.1) that point smoothing in connection with standard coarsening leads to inefficient multigrid methods. However, considering the full isotropic d -dimensional Poisson equation we have $c_{\min} = c_1 = \dots = c_d = \frac{1}{d}$ yielding $E_h(L_h) = \frac{1}{2d}$. This

means that any point smoother will become less efficient with an increasing dimension d even if the problem remains fully isotropic. This deterioration of smoothing factors has already been observed in [Chapter 4](#) comparing the two- and three-dimensional Poisson equation.

Example 5.20 (Optimal ω -JAC point relaxation for the anisotropic diffusion operator). As the anisotropic diffusion operator is represented by an even stencil (5.41), the third item of Theorem 5.1 applies and we can construct an optimal ω -JAC point smoother. m and M are already given above. Together with $\ell_{(0,\dots,0)} = 2/h^2$ one finds

$$\omega_{\text{opt}} = \frac{2}{2 + c_{\min}} \quad \text{and} \quad \rho_1(\omega_{\text{opt}}) = \frac{2 - c_{\min}}{2 + c_{\min}}.$$

For the isotropic d -dimensional Poisson equation, we have $c_{\min} = c_j = 1/d$ for each $j \in \{1, \dots, d\}$ recovering the optimal Jacobi one-stage relaxation from Example 5.17. \blacktriangleleft

Remark 5.17 (Remedy in the case of strong anisotropies). If (5.43) becomes singularly perturbed, i.e., $c_j \rightarrow 0$ for some $j \in \mathcal{I}$, one usually applies block smoothing or partial coarsening; compare with Sections 4.1.1 and 4.2.1. The measure of h -ellipticity for the anisotropic diffusion operator with partial coarsening can be calculated as in the case of standard coarsening:

$$E_h(L_h) = \frac{c_{\min}^c}{2} \quad \text{with} \quad c_{\min}^c = \min_{j \in \mathcal{I}^c} c_j. \quad (5.45)$$

For \mathcal{I}^c we refer to Definition 5.4. As (5.45) indicates, the grid should only be coarsened along those coordinate directions where the corresponding coefficients c_j are reasonably bounded away from zero, in order to obtain a reasonable measure of h -ellipticity. The h -ellipticity with regard to partial coarsening is called semi- h -ellipticity in [7]. \blacktriangleright

Remark 5.18 ($E_h(L_h)$ for fourth-order discretizations of the Laplacian). The measure of h -ellipticity for the fourth-order discretization of the Laplacian based on a “larger” stencil (compare with Sections 4.1.2 and 4.2.2 for the two- and three-dimensional case, respectively) is given by

$$E_h(L_h) = \frac{7}{16d}.$$

Similar as for the second-order central discretization, we observe a deterioration of the measure of h -ellipticity with an increasing dimension.

For the two- and three-dimensional variant of the Mehrstellen discretization one obtains

$$E_h(L_h) = \frac{3}{8}$$

yielding an optimal point ω -JAC relaxation with smoothing factor $\rho_1(\omega_{\text{opt}}) = 5/11$ and optimal relaxation parameters $\omega_{\text{opt}} = 10/11$ ($d = 2$) and $\omega_{\text{opt}} = 12/11$ ($d = 3$). For this particular discretization, it is possible to retain the amount of h -ellipticity and consequently the smoothing properties of point ω -JAC relaxation in the transition from two to three dimensions which has already been reported in Section 4.2.3. \blacktriangleright

5.8.2 Example 2: convection diffusion equation

We continue our discussion of the measure of h -ellipticity by considering the two-dimensional convection diffusion equation. Analogous results may be derived for the three-dimensional variant. We focus on the first-order upwind discretization (4.10) of the convection diffusion operator from Section 4.1.7. For brevity it will be denoted by 1UD in the following. We would like to point out that 1UD is equivalent to a combination of a central discretization and an “artificial viscosity” term. For constant coefficients, it can be written as

$$\begin{aligned} L_h(\beta) = & -\varepsilon (D_h^{11} + D_h^{22}) + aD_h^1 + bD_h^2 \\ & - \frac{|a|h}{2}D_h^{11} - \frac{|b|h}{2}D_h^{22} \quad (\text{artificial viscosity}). \end{aligned}$$

Here, we restrict the considerations to convection angles with $a, b \geq 0$. This does not imply a loss of generality, as it will be seen below. The corresponding Fourier symbol can be found with the help of Example 5.2:

$$\begin{aligned} \tilde{L}_h(\theta) = & \frac{1}{h^2} (4\varepsilon + h(a+b) - (ha + \varepsilon) \exp(-i\theta_1) - \varepsilon \exp(i\theta_1) \\ & - (hb + \varepsilon) \exp(-i\theta_2) - \varepsilon \exp(i\theta_2)). \end{aligned}$$

Instead of deriving complicated and intricate formulae for the whole range of ε, h, a , and b we focus on some representative situations.

In the first case, the convection is aligned with the grid, for example, $b = 0$. This induces a highly anisotropic artificial viscosity which cannot be compensated by the isotropic diffusive part of the discretization, if the convection is dominating (i.e., $\varepsilon \ll |a|h/2$). Therefore, it might be difficult to find an efficient point smoother which is reflected by the following example.

Example 5.21 (h -Ellipticity of 1UD, aligned case). For $a = 1$ and $b = 0$ one easily finds $M = \tilde{L}_h(\pi, \pi) = \frac{1}{h^2}(8\varepsilon + 2h)$ and $m = \tilde{L}_h(0, \pi/2) = \frac{1}{h^2}2\varepsilon$ leading to

$$E_h(L_h) = \frac{\varepsilon}{4\varepsilon + h} \in \left(0, \frac{1}{4}\right).$$

For the remaining types of grid alignment ($a = -1, b = 0$ and $a = 0, b = \pm 1$), one obtains the same expression if $\tilde{L}_h(\theta)$ is adapted according to the respective discretization. \blacktriangleleft

If $\varepsilon \gg h$, the diffusive part of the operator dominates and $E_h(L_h)$ tends to $1/4$, the h -ellipticity measure of the two-dimensional isotropic Poisson equation; see Example 5.19. Then, one may apply a standard point-relaxation method like GS-LEX-SW.

However, in the case of dominant convection ($\varepsilon \rightarrow 0$) $E_h(L_h)$ tends to 0 as a consequence of the anisotropic artificial viscosity; see above. On the other hand it is well known [62, 65] that point GS-LEX-SW is an excellent solver (and thus an efficient smoother as well) for this problem in connection with Dirichlet boundary conditions which seems to be a contradiction in light of Theorem 5.1. This can be explained as follows. In the case of GS-LEX-SW, we have

$$\begin{aligned}\tilde{L}_h^-(\boldsymbol{\theta}) &= \frac{1}{h^2} (-\varepsilon \exp(i\theta_1) - \varepsilon \exp(i\theta_2)) \quad \text{and} \\ \tilde{L}_h^+(\boldsymbol{\theta}) + \tilde{L}_h^0(\boldsymbol{\theta}) &= \frac{1}{h^2} (4\varepsilon + h(a+b) - (ha + \varepsilon) \exp(-i\theta_1) \\ &\quad - (hb + \varepsilon) \exp(-i\theta_2)).\end{aligned}\tag{5.46}$$

For $a = 1, b = 0$ and the “problematic” high frequency, $\tilde{L}_h^+(0, \pi/2) + \tilde{L}_h^0(0, \pi/2)$ reduces to $\frac{\varepsilon}{h^2}(3 + i)$. This means that for the limiting value, $\varepsilon = 0$, Theorem 5.1 does not apply because $\tilde{L}_h^+(0, \pi/2) + \tilde{L}_h^0(0, \pi/2) = 0$. On the contrary, the efficiency of GS-LEX-SW can be deduced from local Fourier analysis. The smoothing factor for $\varepsilon \ll 1$ of GS-LEX-SW is attained for $\boldsymbol{\theta} = (0, \pi/2)$ and thus given by $\rho_1 = |\varepsilon(1 + i)|/|\varepsilon(3 + i)| = 1/\sqrt{5} \approx 0.44721$. This is still too pessimistic, since GS-LEX-SW becomes an exact solver for $\varepsilon = 0$ and Dirichlet boundary conditions. However, if we apply the heuristic definition of the smoothing factor in connection with Dirichlet boundary conditions from Definition 5.7, $\rho_1^D \rightarrow 0$ for $\varepsilon \rightarrow 0$. As an example, we consider $a = 1, b = 0, \varepsilon = 10^{-5}$, and $h = 1/128$ yielding $\rho_1^D = 0.03669$.

Summarizing, we have seen that GS-LEX-SW is an efficient smoother for $a = 1$ and $b = 0$ with smoothing factors $\rho_1 \in [1/\sqrt{5}, 1/2)$ and $\rho_1^D \in (0, 1/2)$ for varying ε and h .

Remark 5.19 (Failure of “direction-free” point smoothers). For direction-free point smoothers such as JAC or RB-JAC relaxation, we always have $\tilde{L}_h^+(\boldsymbol{\theta}) + \tilde{L}_h^0(\boldsymbol{\theta}) = \tilde{L}_h^0(\boldsymbol{\theta}) \neq 0$ for $(\boldsymbol{\theta} \in \Theta_{\text{high}})$ and hence Theorem 5.1 applies. This means that there is no efficient direction-free point smoother in the aligned case, regarding Example 5.21. However, alternating line ω -JAC and alternating line RB-JAC relaxations are robust smoothers for 1UD and the whole admissible range of a, b, ε , and h , as they can cope with the anisotropic artificial viscosity. ►

We continue the discussion by investigating the fully nonaligned case, $a = b = 1$. First of all, note that $\tilde{L}_h^+(\boldsymbol{\theta}) + \tilde{L}_h^0(\boldsymbol{\theta}) \neq 0$ for each $\boldsymbol{\theta} \in \Theta_{\text{high}}$ in the case of GS-LEX-SW, see (5.46). Then Theorem 5.1 applies for this relaxation in the nonaligned case.

Example 5.22 (h -Ellipticity of (1UD), nonaligned case). For $a = b = 1$ one may establish that $M = \tilde{L}_h(\pi, \pi) = \frac{1}{h^2}(8\varepsilon + 4h)$. It is not possible to give a uniform expression for m covering the whole range of ε and h . However, for $\varepsilon \rightarrow 0$ one finds $m = |\tilde{L}_h(\pi/2, \arctan(-1/2))|$ leading to $E_h(L_h) = \frac{1}{4}\sqrt{6 - 2\sqrt{5}} \approx 0.30902$ in the limit case, $\varepsilon = 0$, independent of h . If, on the other hand, ε dominates, one obtains $m = |\tilde{L}_h(0, \pi/2)|$ yielding $E_h(L_h) = \sqrt{(2\varepsilon + h)^2 + h^2}/(8\varepsilon + 4h)$. Therefore, $E_h(L_h)$ tends to $1/4$ for $\varepsilon \gg h$. Summarizing, we get the following range for the measure of h -ellipticity

$$E_h(L_h) \in \left(\frac{1}{4}, \frac{1}{4}\sqrt{6 - 2\sqrt{5}} \right) \approx (0.25, 0.30902).$$

◀

From the definition of the smoothing factor (see (5.40)) and from (5.46) one immediately sees that $\rho_1 \rightarrow 0$ for $\varepsilon \rightarrow 0$, since $\tilde{L}_h^+(\boldsymbol{\theta}) + \tilde{L}_h^0(\boldsymbol{\theta}) \neq 0$ and $\tilde{L}_h^-(\boldsymbol{\theta}) \rightarrow 0$ for $\varepsilon \rightarrow 0$ ($\boldsymbol{\theta} \in \Theta$). For example, $a = b = 1$, $\varepsilon = 10^{-5}$, and $h = 1/128$ yield $\rho_1 = 0.00128$ ($= \rho_1^D$) and $E_h(L_h) = 0.30895$. If $\varepsilon \gg h$ we observe as before that $\rho_1 \rightarrow 1/2$. This implies that with varying ε and h the smoothing factor ρ_1 (ρ_1^D) ranges in $(0, 1/2)$.

From the discussed types of grid (non-)alignment one can conclude that GS-LEX-SW is a robust smoother for $a, b \geq 0$ and for the admissible range of ε and h . Analogous considerations hold for $a, b \leq 0$, for $a \geq 0$ and $b \leq 0$, and for $a \leq 0$ and $b \geq 0$ in connection with GS-LEX-NE, GS-LEX-SE, and GS-LEX-NW, respectively. Thus, a robust smoother for the first-order upwind discretization of the convection diffusion operator is given by four-direction GS-LEX, i.e., four consecutive GS-LEX sweeps where each sweep starts in a different corner of the domain Ω_h .

An analytic treatment of the higher-order upwind discretization (HUD) from Section 4.1.8, is less transparent than the above considerations, since we have an additional parameter, κ . Systematic numerical tests indicate that HUD always shows a certain amount of h -ellipticity (i.e., for all admissible values of a, b, ε, h , and κ), despite the aligned case. The failure of direction-free point smoothers carries over to HUD as well. However, the determination of an efficient smoother is more complicated here, as HUD results in a matrix with positive off-diagonal elements. For this kind of matrix, the convergence theory of classical relaxation methods does not apply [63]. A detailed study of this discretization and the development of an appropriate robust smoother, the so-called KAPPA-smoother, can be found in [49, 62].

5.8.3 Example 3: Oseen equations

Definition 5.10 can be generalized to systems of q equations with the related Fourier symbol $\tilde{L}_h(\boldsymbol{\theta}) \in \mathbb{C}^{q \times q}$, see Section 5.1.2.

Definition 5.11 (Measure of h -Ellipticity for Systems of PDEs): The definition of the h -ellipticity measure of an operator \mathbf{L}_h representing a system of equations applies the determinant of its Fourier symbol, $\det(\tilde{\mathbf{L}}_h(\boldsymbol{\theta}))$:

$$E_h(\mathbf{L}_h) := \frac{\min \left\{ \left| \det(\tilde{\mathbf{L}}_h(\boldsymbol{\theta})) \right| : \boldsymbol{\theta} \in \Theta_{\text{high}} \right\}}{\max \left\{ \left| \det(\tilde{\mathbf{L}}_h(\boldsymbol{\theta})) \right| : \boldsymbol{\theta} \in \Theta \right\}}.$$

▲

A generalization of Theorem 5.1 concerning the existence of efficient point or block smoothers for systems of equations can be found in [7, 41], where the measure of semi- h -ellipticity is discussed.

The first-order flux difference-splitting discretization (1FD) given in Section 4.3.3 behaves qualitatively similar as 1UD: four-direction point GS-LEX is a robust smoother and direction-free point relaxations deteriorate with an increasing alignment. The derivation of analytical values for the measure of h -ellipticity is very cumbersome and does not lead to any further insights. Therefore, we only give some numerical examples. In the aligned case, $E_h(\mathbf{L}_h)$ tends to 0, if $\varepsilon \rightarrow 0$. The main difference in 1UD is that GS-LEX *does not* become an exact solver for $\varepsilon \rightarrow 0$. For example, $a = 1$, $b = 0$, $\varepsilon = 0$, and $h = 1/32$ yield $E_h(\mathbf{L}_h) = 0$, $\rho_1 = 1.0$, and $\rho_1^D = 0.88402$. If we replace $\varepsilon = 0$ by $\varepsilon = 10^{-4}$ and keep the remaining parameters fixed, one obtains $E_h(\mathbf{L}_h) = 0.00034$, $\rho_1 = 0.98894$, and $\rho_1^D = 0.88127$. As an example for the fully nonaligned case we consider $a = b = 1$, $\varepsilon = 10^{-4}$, and $h = 1/32$ leading to $E_h(\mathbf{L}_h) = 0.03815$ and $\rho_1 = \rho_1^D = 0.79081$. Obviously, the smoothing properties of GS-LEX are considerably worse for 1FD compared with 1UD.

For the higher-order flux difference-splitting discretization (HFD) from Section 4.3.4 similar considerations hold as for HUD. Apart from the aligned cases, one observes a small amount of h -ellipticity for the whole range of a, b, h, ε , and κ . The construction of an efficient smoothing procedure is, however, not straightforward due to positive definite off-diagonal blocks; see [45, 66] for details. Again, the KAPPA smoother from [49] is found to be a robust relaxation method for this discretization.

A second variant for HUD and HFD is to apply multistage point JAC relaxations, see [15, 20, 31, 40, 66]. This approach is very popular because it is fully parallel and can be easily implemented.

Chapter 6

FOURIER TWO- AND THREE-GRID ANALYSIS

In the previous chapter, dealing with Fourier smoothing analysis, we focused on the relaxation operator used in a multigrid method. The k -grid analysis ($k = 2, 3$) also takes into account the remaining operators which constitute the coarse-grid correction. It has been developed in order to investigate the interplay between relaxation and coarse-grid correction which is crucial for an efficient multigrid method. Fourier k -grid analysis yields *quantitative* convergence estimates—assuming a proper treatment of the boundary—as it has been demonstrated in [Chapter 4](#) by an extensive study of test cases. For nonsymmetric problems, it can be considered as the main analysis tool.

An important issue of this chapter is the presentation of the k -grid analysis to the same extent as it has been realized in the accompanying software. According to this aim, we give a comprehensive description of the state of the art of local Fourier k -grid analysis. In particular, our presentation includes three-dimensional applications, semicoarsening, and systems of equations. The corresponding k -grid analysis is technically rather complicated and usually not contained in multigrid textbooks (not even for $k = 2$).

Especially, the introduction of the three-grid analysis is of great benefit since for several multigrid components or cycle variants, the asymptotic *multigrid* convergence cannot be predicted accurately by a classical Fourier *two*-grid analysis. For example, one may use different discretizations on different grids. It can be beneficial to replace the $2h$ -, $4h$ -, etc. analogs of the fine-grid discretization on the coarser grids by other discretizations. The most prominent example of this kind is the Galerkin coarse-grid operator. As the entries of the Galerkin coarse-grid discretizations are in general not known in advance, they may not be favorable for the smoothing method applied. Investigations of the two-grid iteration cannot display possible smoothing difficulties on coarser grids induced by the different discretizations, since the direct solution of the $2h$ -problem is assumed. Furthermore, one might be interested in the influence on the asymptotic convergence factor of V-cycles versus W-cycles, of different numbers of pre- and postsmoothing, or of different smoothers on different grids. Another important advantage of Fourier three-grid analysis is the possibility to obtain additional valuable insight into coarse-grid correction difficulties occurring for singularly perturbed problems like the convection diffusion equation with dominating convection. Such relevant phenomena can be

investigated in a Fourier *three-grid* analysis which is therefore clearly superior to the classical two-grid analysis. A collection of illustrative examples is given in [Chapter 4](#).

Fourier three-grid analysis is based on a recursive application of the two-grid analysis. Hence, a profound understanding of the two-grid analysis should be accomplished before addressing the extension to three grids. Having this requirement in mind, we give a thorough description of the two-grid analysis for two-dimensional scalar applications in connection with standard coarsening to ensure a plain presentation of its basic principles (Section 6.2). The necessary modifications for other coarsening strategies, three-dimensional problems, and systems of equations are detailed in Sections 6.2.8, 6.3, and 6.4, respectively. Finally, Fourier three-grid analysis is introduced in Section 6.5. We start our presentation of the Fourier *k-grid* analysis with a short review of its basic assumptions which have already been outlined in Section 2.2.

6.1 BASIC ASSUMPTIONS

We consider a discrete linear boundary value problem that is iteratively solved by a *k-grid* method. Recall from the previous chapter that smoothing analysis turned out to be valid for fine-grid discretization operators and relaxation methods with *constant coefficients* that are defined on an *infinite grid*. In the *k-grid* ($k = 2, 3$) analysis, one furthermore assumes that

- coarse-grid discretizations and
- intergrid transfers

are given by operators with constant coefficients and that they are extended to infinite grids as well. Consequently, each operator involved in the coarse-grid correction can be represented by one stencil (see Sections 6.2.3 and 6.2.4 below, for the stencil notation of transfer operators). Therefore it is possible to calculate the corresponding Fourier representations based on small subspaces of Fourier components, similar as in [Chapter 5](#) for the relaxation method.

The discrete fine-grid solution u_h and a current approximation $u_h^{(i)}$ can be represented by linear combinations of Fourier components, because these generate the whole space of bounded infinite grid functions, see Definition 5.2. The same holds for the error

$$e_h^{(i-1)} := u_h^{(i-1)} - u_h \quad \text{before and} \quad e_h^{(i)} := u_h^{(i)} - u_h$$

after the *i*-th *k-grid* cycle. Therefore it is convenient to investigate the error transformation (or the residual transformation) by a *k-grid* cycle by an analysis of its Fourier representation. The derivation of these Fourier representations for a variety of *k-grid* methods (involving two- and three-dimensional

applications, scalar equations and systems of equations, standard and semi-coarsening, various transfer operators, direct and Galerkin coarse-grid discretization) is the main goal of the following sections. It turns out that the complete k -grid error (or defect) reduction operator is unitarily equivalent to a block-diagonal matrix with small blocks. Due to this simple block structure it is easily possible to calculate corresponding spectral radii or norm values yielding quantitative convergence estimates.

6.2 TWO-GRID ANALYSIS FOR 2D SCALAR PROBLEMS

The error transformation by a two-grid cycle reads $e_h^{(i)} = M_h^{2h} e_h^{(i-1)}$ with error-transformation operator (3.6)

$$M_h^{2h} = S_h^{\nu_2} K_h^{2h} S_h^{\nu_1} = S_h^{\nu_2} (I_h - P_{2h}^h L_{2h}^{-1} R_{2h}^{2h} L_h) S_h^{\nu_1}. \quad (6.1)$$

S_h is a smoothing operator on G_h , ν_1 and ν_2 indicate the number of pre- and postsmoothing iterations, K_h^{2h} is the coarse-grid correction operator, I_h the G_h -identity, L_{2h} the approximation of L_h on a coarse grid G_{2h} , P_{2h}^h and R_{2h}^h are transfer operators from coarse to fine grids and vice versa, and γ is the cycle index (for example, $\gamma = 1$ denotes a V-cycle and $\gamma = 2$ a W-cycle). L_{2h} may be defined by Galerkin coarsening, $L_{2h} = \tilde{R}_h^{2h} L_h \tilde{P}_{2h}^h$, or simply by a straightforward application of L_h on G_{2h} (called GCA and DCA for brevity; see Section 3.4.3). Note that the transfer operators in the Galerkin process need not necessarily match with R_{2h}^{2h} and P_{2h}^h from the multigrid iteration.

6.2.1 Spaces of $2h$ -harmonics

In the two-dimensional case, it is appropriate to divide the Fourier space into the following four-dimensional subspaces. This is a special case of the more general Definition 5.8 for d -dimensional problems.

Definition 6.1 ($2h$ -harmonics, $d = 2$). The $2h$ -harmonics $\mathcal{F}_{2h}(\theta)$ are given by

$$\mathcal{F}_{2h}(\theta) := \text{span}\{ \varphi_h(\theta^{00}, \cdot), \varphi_h(\theta^{11}, \cdot), \varphi_h(\theta^{10}, \cdot), \varphi_h(\theta^{01}, \cdot) \} \quad \text{with} \\ \theta = \theta^{00} \in \Theta_{2h} := (-\pi/2, \pi/2]^2, \quad \theta^\alpha := \theta^{00} - (\alpha_1 \text{sign}(\theta_1), \alpha_2 \text{sign}(\theta_2)) \pi.$$

▲

Definition 6.1 is particularly related to standard coarsening ($H = 2h$) in the following way: We have $\Theta_{2h} = \Theta_{\text{low}}$, where *low* refers to standard coarsening; compare with Section 5.2 for the distinction between low and high

frequencies with respect to different coarsening strategies. The definition of the $2h$ -harmonics is motivated by the fact that each *low frequency* $\theta^{00} \in \Theta_{2h}$ is coupled with three *high frequencies* θ^α with $\alpha \neq (00)$ in the transition from G_h to G_{2h} . That is, the three high-frequency components are not visible on the coarse grid as they coincide with the corresponding low-frequency component, that is,

$$\varphi_h(\theta^{00}, \mathbf{x}) = \varphi_h(\theta^{11}, \mathbf{x}) = \varphi_h(\theta^{10}, \mathbf{x}) = \varphi_h(\theta^{01}, \mathbf{x}) \quad \text{for } \mathbf{x} \in G_{2h}. \quad (6.2)$$

(6.2) might be easily verified applying the periodicity of the exponential function. Interpreting the Fourier components as *coarse-grid* functions gives

$$\varphi_h(\theta^{00}, \mathbf{x}) = \varphi_{2h}(2\theta^{00}, \mathbf{x}) \quad (= \varphi_{2h}(2\theta^\alpha, \mathbf{x})) \quad \text{with } \theta^{00} \in \Theta_{2h}, \mathbf{x} \in G_{2h}. \quad (6.3)$$

A sample set of coupled frequencies is illustrated in Figure 6.1.

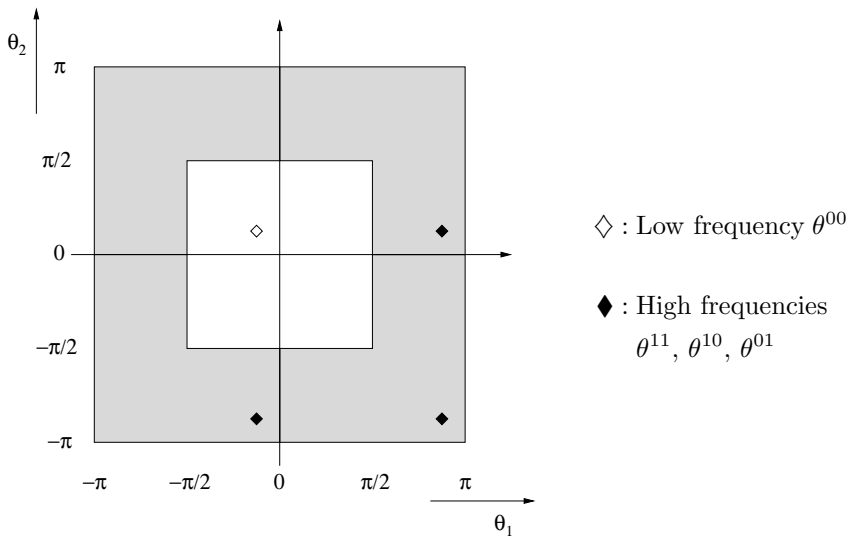


FIGURE 6.1: High and low Fourier frequencies generating a space of $2h$ -harmonics. Standard coarsening, $d = 2$.

It will turn out below that the coarse-grid correction

$$K_h^{2h} = I_h - P_{2h}^h L_{2h}^{-1} R_h^{2h} L_h$$

leaves the spaces of $2h$ -harmonics invariant yielding a simple block-diagonal representation for K_h^{2h} consisting of (4×4) -blocks

$$K_h^{2h}|_{\mathcal{F}_{2h}(\theta)} =: K^{2g}(\theta) \quad (\theta \in \Theta_{2h}).$$

In the following, we calculate the Fourier representations of the different operators involved in the coarse-grid correction of a two-grid method with respect to the $2h$ -harmonics. To ensure a unique definition, we need to prescribe an order of the multiindex α . For a consistent treatment of relaxation and coarse-grid correction we have to use the same order as in the Fourier smoothing analysis from the previous chapter:

$$(0, 0), \quad (1, 1), \quad (1, 0), \quad (0, 1).$$

6.2.2 Fourier representation of fine-grid discretization

The treatment of the fine-grid discretization is particularly simple, as the Fourier components are eigenfunctions for constant coefficient operators $L_h \triangleq [\ell_\kappa]_h$. Thus, the Fourier representation of L_h w.r.t. the $2h$ -harmonics is given by a diagonal matrix,

$$L_h|_{\mathcal{F}_{2h}(\theta)} \triangleq L^{2g}(\theta) = \begin{pmatrix} \tilde{L}_h(\theta^{00}) & 0 & 0 & 0 \\ 0 & \tilde{L}_h(\theta^{11}) & 0 & 0 \\ 0 & 0 & \tilde{L}_h(\theta^{10}) & 0 \\ 0 & 0 & 0 & \tilde{L}_h(\theta^{01}) \end{pmatrix} \in \mathbb{C}^{4 \times 4}$$

with Fourier symbols $\tilde{L}_h(\theta^\alpha) = \sum_{\kappa \in J} \ell_\kappa \exp(i\kappa\theta^\alpha)$ and the spaces of $2h$ -harmonics are invariant under an application of L_h , i.e.,

$$L_h : \mathcal{F}_{2h}(\theta) \longrightarrow \mathcal{F}_{2h}(\theta) \quad (\theta \in \Theta_{2h}).$$

6.2.3 Fourier representation of restriction

Consider an arbitrary restriction which is characterized by a constant coefficient stencil $R_h^{2h} \triangleq [r_\kappa]_h^{2h}$. That is, an infinite grid function $v_h : G_h \rightarrow \mathbb{C}$ is transferred to the coarse grid G_{2h} in the following way:

$$(R_h^{2h} v_h)(x) = \sum_{\kappa \in J} r_\kappa v_h(x + \kappa h) \quad (x \in G_{2h}).$$

We have $x + \kappa h \in G_h$ for $x \in G_{2h}$. For the Fourier components this leads to

$$\begin{aligned} (R_h^{2h} \varphi_h(\theta^\alpha, \cdot))(x) &= \sum_{\kappa \in J} r_\kappa \varphi_h(\theta^\alpha, x + \kappa h) = \sum_{\kappa \in J} r_\kappa \exp(i(x + \kappa h)\theta^\alpha/h) \\ &= \sum_{\kappa \in J} r_\kappa \exp(i\kappa\theta^\alpha) \varphi_h(\theta^\alpha, x) \\ &= \underbrace{\sum_{\kappa \in J} r_\kappa \exp(i\kappa\theta^\alpha)}_{\tilde{R}_h^{2h}(\theta^\alpha)} \varphi_{2h}(2\theta^{00}, x) \quad (x \in G_{2h}) \end{aligned} \quad (6.4)$$

TABLE 6.1: Fourier symbols for restriction operators implemented in the accompanying software; $d = 2$

restriction	Fourier symbol $\tilde{R}_h^{2h}(\boldsymbol{\theta}^\alpha)$
full weighing	$\frac{1}{4}(1 + \cos(\theta_1^\alpha))(1 + \cos(\theta_2^\alpha))$
half weighing	$\frac{1}{4}(2 + \cos(\theta_1^\alpha) + \cos(\theta_2^\alpha))$
injection	1
higher-order	$\frac{1}{256} (8 + 9 \cos(\theta_1^\alpha) - \cos(3\theta_1^\alpha))$ $(8 + 9 \cos(\theta_2^\alpha) - \cos(3\theta_2^\alpha))$
seven point	$\frac{1}{4}(1 + \cos(\theta_1^\alpha) + \cos(\theta_2^\alpha) + \cos(\theta_1^\alpha - \theta_2^\alpha))$

with Fourier symbol $\tilde{R}_h^{2h}(\boldsymbol{\theta}^\alpha)$. Equation (6.4) is due to the aliasing effect of the $2h$ -harmonics for $\boldsymbol{x} \in G_{2h}$; compare with (6.2) and (6.3). As a consequence we obtain

$$R_h^{2h} : \mathcal{F}_{2h}(\boldsymbol{\theta}) \longrightarrow \text{span}\{\varphi_{2h}(2\boldsymbol{\theta}, \cdot)\} \quad (\boldsymbol{\theta} \in \Theta_{2h}).$$

Example 6.1 (Fourier symbol for the full weighting operator). The Fourier representation of the full weighting operator

$$R_h^{2h} \triangleq \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}_h^{2h}$$

is governed by

$$\begin{aligned} \tilde{R}_h^{2h}(\boldsymbol{\theta}^\alpha) &= \frac{1}{16}(4 + 4 \cos(\theta_1^\alpha) + 4 \cos(\theta_2^\alpha) + 2 \cos(\theta_1^\alpha + \theta_2^\alpha) + 2 \cos(\theta_1^\alpha - \theta_2^\alpha)) \\ &= \frac{1}{4}(1 + \cos(\theta_1^\alpha))(1 + \cos(\theta_2^\alpha)). \end{aligned}$$

◀

Fourier symbols for two-dimensional restriction operators implemented in the accompanying software (see also Figure 3.7) are listed in Table 6.1. The corresponding stencil representations are given in Section 3.4.5.

6.2.4 Fourier representation of prolongation

Dealing with the prolongation operator is to some extent more complicated. Therefore we firstly derive the Fourier representation for a concrete example—bilinear interpolation—before the general case is described.

Recall the distinction of the two-dimensional infinite grid into four different types of grid points

$$G_h = G_h^{00} \cup G_h^{11} \cup G_h^{10} \cup G_h^{01},$$

as discussed in Section 5.4.3. The definition of G_h^α is specified in (5.18) and Table 5.4. Different subsets G_h^α are shown in Figure 6.2. In case of standard

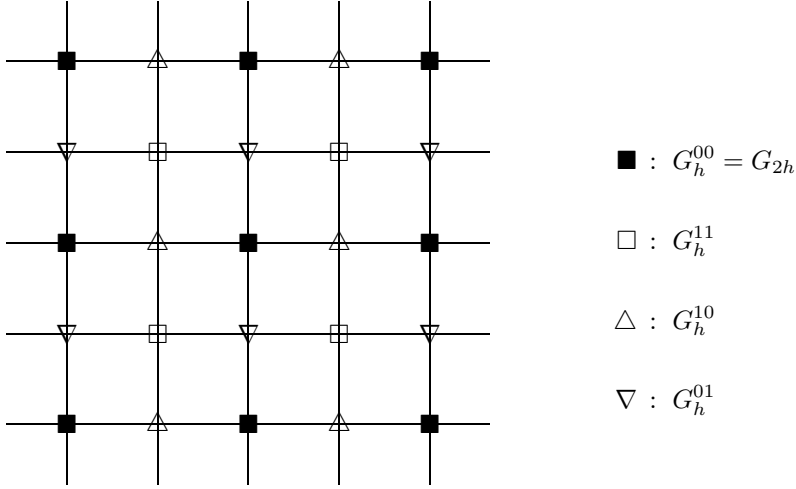


FIGURE 6.2: Different types of grid points in G_h .

coarsening the coarse grid is given by $G_{2h} = G_h^{00}$. Applying bilinear interpolation (P_{2h}^h) to map a coarse-grid function $v_{2h} : G_{2h} \rightarrow \mathbb{C}$ onto the fine grid yields a fine-grid function $(P_{2h}^h v_{2h})(\cdot) : G_h \rightarrow \mathbb{C}$ defined by

$$(P_{2h}^h v_{2h})(\mathbf{x}) = \begin{cases} v_{2h}(\mathbf{x}) & \text{for } \mathbf{x} \in G_h^{00} \\ \frac{1}{4}(v_{2h}(x_1 + h, x_2 + h) + v_{2h}(x_1 + h, x_2 - h) \\ \quad + v_{2h}(x_1 - h, x_2 + h) + v_{2h}(x_1 - h, x_2 - h)) & \text{for } \mathbf{x} \in G_h^{11} \\ \frac{1}{2}(v_{2h}(x_1 + h, x_2) + v_{2h}(x_1 - h, x_2)) & \text{for } \mathbf{x} \in G_h^{10} \\ \frac{1}{2}(v_{2h}(x_1, x_2 + h) + v_{2h}(x_1, x_2 - h)) & \text{for } \mathbf{x} \in G_h^{01}. \end{cases} \quad (6.5)$$

This *distribution* of the coarse-grid function governed by bilinear interpolation can be expressed in stencil notation by

$$P_{2h}^h \triangleq]p\kappa[_{2h}^h = \frac{1}{4} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}_{2h}^h. \quad (6.6)$$

with $\kappa \in J$. $J \subset \mathbb{Z}^2$ is a finite index set containing $(0,0)$, similar as for discretization and restriction operators. The stencil elements are weighting factors in a distribution process indicated by the reversed brackets. For example, each coarse-grid value $v_{2h}(\mathbf{x})$ ($\mathbf{x} \in G_{2h} = G_h^{00}$) is distributed to its four

diagonal neighbors weighted by $p_{(1,1)} = p_{(1,-1)} = p_{(-1,1)} = p_{(-1,-1)} = 1/4$ to obtain fine-grid values for $(P_{2h}^h v_{2h})(\mathbf{x})$ with $\mathbf{x} \in G_h^{11}$; compare with Figure 6.2 and (6.5).

For the derivation of the Fourier representation of bilinear interpolation with respect to the $2h$ -harmonics, we apply similar considerations as discussed in Section 5.4.3 in connection with smoothing analysis for RB-JAC relaxations. The relation between prolongation operators and RB-JAC smoothers is due to the fact that both kinds of operators are differently defined for different types of grid points (remember that RB-JAC relaxation distinguishes between red and black points whereas a prolongation operator like bilinear interpolation distinguishes between G_h^{00} , G_h^{11} , G_h^{10} , and G_h^{01}). In analogy to (5.19) one might verify that

$$\begin{aligned}\varphi_h(\boldsymbol{\theta}^{11}, \mathbf{x}) &= \begin{cases} \varphi_h(\boldsymbol{\theta}^{00}, \mathbf{x}) & \text{for } \mathbf{x} \in G_h^{00} \cup G_h^{11}, \\ -\varphi_h(\boldsymbol{\theta}^{00}, \mathbf{x}) & \text{for } \mathbf{x} \in G_h^{10} \cup G_h^{01}. \end{cases} \\ \varphi_h(\boldsymbol{\theta}^{10}, \mathbf{x}) &= \begin{cases} \varphi_h(\boldsymbol{\theta}^{00}, \mathbf{x}) & \text{for } \mathbf{x} \in G_h^{00} \cup G_h^{01}, \\ -\varphi_h(\boldsymbol{\theta}^{00}, \mathbf{x}) & \text{for } \mathbf{x} \in G_h^{11} \cup G_h^{10}. \end{cases} \\ \varphi_h(\boldsymbol{\theta}^{01}, \mathbf{x}) &= \begin{cases} \varphi_h(\boldsymbol{\theta}^{00}, \mathbf{x}) & \text{for } \mathbf{x} \in G_h^{00} \cup G_h^{10}, \\ -\varphi_h(\boldsymbol{\theta}^{00}, \mathbf{x}) & \text{for } \mathbf{x} \in G_h^{11} \cup G_h^{01}. \end{cases}\end{aligned}\quad (6.7)$$

Applying (6.7) it can be easily seen that for $\psi_1(\mathbf{x}), \dots, \psi_4(\mathbf{x})$, defined by

$$\begin{aligned}\psi_1(\mathbf{x}) &:= (\varphi_h(\boldsymbol{\theta}^{00}, \mathbf{x}) + \varphi_h(\boldsymbol{\theta}^{11}, \mathbf{x}) + \varphi_h(\boldsymbol{\theta}^{10}, \mathbf{x}) + \varphi_h(\boldsymbol{\theta}^{01}, \mathbf{x}))/4 \\ \psi_2(\mathbf{x}) &:= (\varphi_h(\boldsymbol{\theta}^{00}, \mathbf{x}) + \varphi_h(\boldsymbol{\theta}^{11}, \mathbf{x}) - \varphi_h(\boldsymbol{\theta}^{10}, \mathbf{x}) - \varphi_h(\boldsymbol{\theta}^{01}, \mathbf{x}))/4 \\ \psi_3(\mathbf{x}) &:= (\varphi_h(\boldsymbol{\theta}^{00}, \mathbf{x}) - \varphi_h(\boldsymbol{\theta}^{11}, \mathbf{x}) - \varphi_h(\boldsymbol{\theta}^{10}, \mathbf{x}) + \varphi_h(\boldsymbol{\theta}^{01}, \mathbf{x}))/4 \\ \psi_4(\mathbf{x}) &:= (\varphi_h(\boldsymbol{\theta}^{00}, \mathbf{x}) - \varphi_h(\boldsymbol{\theta}^{11}, \mathbf{x}) + \varphi_h(\boldsymbol{\theta}^{10}, \mathbf{x}) - \varphi_h(\boldsymbol{\theta}^{01}, \mathbf{x}))/4,\end{aligned}$$

the following relations are valid:

$$\begin{aligned}\psi_1(\mathbf{x}) &= \begin{cases} \varphi_h(\boldsymbol{\theta}^{00}, \mathbf{x}) & \text{for } \mathbf{x} \in G_h^{00}, \\ 0 & \text{for } \mathbf{x} \notin G_h^{00}. \end{cases} & \psi_2(\mathbf{x}) &= \begin{cases} \varphi_h(\boldsymbol{\theta}^{00}, \mathbf{x}) & \text{for } \mathbf{x} \in G_h^{11}, \\ 0 & \text{for } \mathbf{x} \notin G_h^{11}. \end{cases} \\ \psi_3(\mathbf{x}) &= \begin{cases} \varphi_h(\boldsymbol{\theta}^{00}, \mathbf{x}) & \text{for } \mathbf{x} \in G_h^{10}, \\ 0 & \text{for } \mathbf{x} \notin G_h^{10}. \end{cases} & \psi_4(\mathbf{x}) &= \begin{cases} \varphi_h(\boldsymbol{\theta}^{00}, \mathbf{x}) & \text{for } \mathbf{x} \in G_h^{01}, \\ 0 & \text{for } \mathbf{x} \notin G_h^{01}. \end{cases}\end{aligned}\quad (6.8)$$

After these preliminary remarks we investigate the mapping of coarse-grid Fourier components $\varphi_{2h}(2\boldsymbol{\theta}^{00}, \cdot)$ onto the fine grid by bilinear interpolation. Assuming $\mathbf{x} \in G_h^{00}$ leads to

$$(P_{2h}^h \varphi_{2h}(2\boldsymbol{\theta}^{00}, \cdot))(\mathbf{x}) = \varphi_{2h}(2\boldsymbol{\theta}^{00}, \mathbf{x}) = \varphi_h(\boldsymbol{\theta}^{00}, \mathbf{x}).$$

Using $\varphi_{2h}(2\theta^{00}, \mathbf{x}) = \varphi_h(\boldsymbol{\theta}, \mathbf{x})$ ($\mathbf{x} \in G_h^{00}$, $\boldsymbol{\theta} = \boldsymbol{\theta}^{00} \in \Theta_{2h}$) one obtains for $\mathbf{x} \in G_h^{11}$

$$\begin{aligned} (P_{2h}^h \varphi_{2h}(2\theta^{00}, \cdot))(\mathbf{x}) &= \frac{1}{4}(\varphi_h(\boldsymbol{\theta}, (x_1 + h, x_2 + h)) + \varphi_h(\boldsymbol{\theta}, (x_1 + h, x_2 - h)) \\ &\quad + \varphi_h(\boldsymbol{\theta}, (x_1 - h, x_2 + h)) + \varphi_h(\boldsymbol{\theta}, (x_1 - h, x_2 - h))) \\ &= \frac{1}{4}(\exp(i(\theta_1 + \theta_2)) + \exp(i(\theta_1 - \theta_2)) + \exp(i(\theta_2 - \theta_1)) \\ &\quad + \exp(-i(\theta_1 + \theta_2))) \varphi_h(\boldsymbol{\theta}, \mathbf{x}) \\ &= \cos(\theta_1^{00}) \cos(\theta_2^{00}) \varphi_h(\boldsymbol{\theta}^{00}, \mathbf{x}). \end{aligned}$$

Similar considerations concerning grid points $\mathbf{x} \in G_h^{10}$ and $\mathbf{x} \in G_h^{01}$ yield the following mapping of $\varphi_{2h}(2\theta^{00}, \cdot)$ onto the fine grid:

$$(P_{2h}^h \varphi_{2h}(2\theta^{00}, \cdot))(\mathbf{x}) = \begin{cases} \varphi_h(\boldsymbol{\theta}^{00}, \mathbf{x}) & \text{for } \mathbf{x} \in G_h^{00}, \\ \cos(\theta_1^{00}) \cos(\theta_2^{00}) \varphi_h(\boldsymbol{\theta}^{00}, \mathbf{x}) & \text{for } \mathbf{x} \in G_h^{11}, \\ \cos(\theta_1^{00}) \varphi_h(\boldsymbol{\theta}^{00}, \mathbf{x}) & \text{for } \mathbf{x} \in G_h^{10}, \\ \cos(\theta_2^{00}) \varphi_h(\boldsymbol{\theta}^{00}, \mathbf{x}) & \text{for } \mathbf{x} \in G_h^{01}. \end{cases} \quad (6.9)$$

Due to (6.8) this can be rewritten for an arbitrary $\mathbf{x} \in G_h$ as

$$\begin{aligned} (P_{2h}^h \varphi_{2h}(2\theta^{00}, \cdot))(\mathbf{x}) &= \psi_1(x) + \cos(\theta_1^{00}) \cos(\theta_2^{00}) \psi_2(x) \\ &\quad + \cos(\theta_1^{00}) \psi_3(x) + \cos(\theta_2^{00}) \psi_4(x) \\ &= \frac{1}{4}(1 + \cos(\theta_1^{00}))(1 + \cos(\theta_2^{00})) \varphi_h(\boldsymbol{\theta}^{00}, \mathbf{x}) \\ &\quad + \frac{1}{4}(1 - \cos(\theta_1^{00}))(1 - \cos(\theta_2^{00})) \varphi_h(\boldsymbol{\theta}^{11}, \mathbf{x}) \\ &\quad + \frac{1}{4}(1 - \cos(\theta_1^{00}))(1 + \cos(\theta_2^{00})) \varphi_h(\boldsymbol{\theta}^{10}, \mathbf{x}) \\ &\quad + \frac{1}{4}(1 + \cos(\theta_1^{00}))(1 - \cos(\theta_2^{00})) \varphi_h(\boldsymbol{\theta}^{01}, \mathbf{x}) \\ &= \frac{1}{4} \sum_{\alpha} (1 + \cos(\theta_1^{\alpha}))(1 + \cos(\theta_2^{\alpha})) \varphi_h(\boldsymbol{\theta}^{\alpha}, \mathbf{x}). \quad (6.10) \end{aligned}$$

Summarizing, bilinear interpolation maps a coarse-grid Fourier component into the corresponding space of $2h$ -harmonics.

For general prolongation operators

$$P_{2h}^h \stackrel{\Delta}{=}]p_{\kappa}[_{2h}^h = \begin{bmatrix} \vdots & \vdots & \vdots \\ \cdots & p_{(-1,1)} & p_{(0,1)} & p_{(1,1)} & \cdots \\ \cdots & p_{(-1,0)} & p_{(0,0)} & p_{(1,0)} & \cdots \\ \cdots & p_{(-1,-1)} & p_{(0,-1)} & p_{(1,-1)} & \cdots \\ \vdots & \vdots & \vdots \end{bmatrix}_{2h}^h,$$

we have to split the index set J referring to nonzero stencil entries:

$$\begin{aligned} J &= J^{00} \cup J^{11} \cup J^{10} \cup J^{01} \quad \text{with} \\ J^{00} &= \{\kappa \in J \mid \kappa_1, \kappa_2 \text{ even}\}, \quad J^{11} = \{\kappa \in J \mid \kappa_1, \kappa_2 \text{ odd}\}, \\ J^{10} &= \{\kappa \in J \mid \kappa_1 \text{ odd}, \kappa_2 \text{ even}\}, \quad J^{01} = \{\kappa \in J \mid \kappa_1 \text{ even}, \kappa_2 \text{ odd}\}. \end{aligned}$$

J^α obviously corresponds to G_h^α . Using this splitting, a general mapping of a coarse-grid function v_{2h} onto the fine grid can be written as

$$(P_{2h}^h v_{2h})(\mathbf{x}) = \sum_{\kappa \in J^\alpha} p_\kappa v_{2h}(\mathbf{x} + \kappa h) \quad \text{for } \mathbf{x} \in G_h^\alpha. \quad (6.11)$$

Example 6.2 (Splitting of J for bilinear interpolation). Considering bilinear interpolation (6.6), the index set J is subdivided into

$$\begin{aligned} J^{00} &= \{(0, 0)\}, \quad J^{11} = \{(1, 1), (1, -1), (-1, 1), (-1, -1)\}, \\ J^{10} &= \{(1, 0), (-1, 0)\}, \quad J^{01} = \{(0, 1), (0, -1)\}. \end{aligned}$$

The concise definition (6.11) of a general prolongation is broken down into the different subgrids in (6.5). ▲

Applying (6.11) yields the mapping of a coarse-grid Fourier component onto G_h :

$$\begin{aligned} (P_{2h}^h \varphi_{2h}(2\theta^{00}, \cdot))(\mathbf{x}) &= \tilde{P}^\alpha(\theta^{00}) \varphi_h(\theta^{00}, \mathbf{x}) \quad \text{for } \mathbf{x} \in G_h^\alpha \quad (6.12) \\ \text{with } \tilde{P}^\alpha(\theta^{00}) &= \sum_{\kappa \in J^\alpha} p_\kappa \exp(i\theta^{00} \kappa). \end{aligned}$$

For example, for bilinear interpolation (6.9) we have

$$\begin{aligned} \tilde{P}^{00}(\theta^{00}) &= 1, \quad \tilde{P}^{11}(\theta^{00}) = \cos(\theta_1^{00}) \cos(\theta_2^{00}), \\ \tilde{P}^{10}(\theta^{00}) &= \cos(\theta_1^{00}), \quad \tilde{P}^{01}(\theta^{00}) = \cos(\theta_2^{00}). \end{aligned}$$

Combining (6.12) and (6.8) gives for arbitrary $\mathbf{x} \in G_h$:

$$\begin{aligned} (P_{2h}^h \varphi_{2h}(2\theta^{00}, \cdot))(\mathbf{x}) &= \tilde{P}^{00}(\theta^{00}) \psi_1(\mathbf{x}) + \tilde{P}^{11}(\theta^{00}) \psi_2(\mathbf{x}) \\ &\quad + \tilde{P}^{10}(\theta^{00}) \psi_3(\mathbf{x}) + \tilde{P}^{01}(\theta^{00}) \psi_4(\mathbf{x}) \\ &= \frac{1}{4}(\tilde{P}^{00}(\theta^{00}) + \tilde{P}^{11}(\theta^{00}) + \tilde{P}^{10}(\theta^{00}) + \tilde{P}^{01}(\theta^{00})) \varphi_h(\theta^{00}, \mathbf{x}) \\ &\quad + \frac{1}{4}(\tilde{P}^{00}(\theta^{00}) + \tilde{P}^{11}(\theta^{00}) - \tilde{P}^{10}(\theta^{00}) - \tilde{P}^{01}(\theta^{00})) \varphi_h(\theta^{11}, \mathbf{x}) \\ &\quad + \frac{1}{4}(\tilde{P}^{00}(\theta^{00}) - \tilde{P}^{11}(\theta^{00}) - \tilde{P}^{10}(\theta^{00}) + \tilde{P}^{01}(\theta^{00})) \varphi_h(\theta^{10}, \mathbf{x}) \\ &\quad + \frac{1}{4}(\tilde{P}^{00}(\theta^{00}) - \tilde{P}^{11}(\theta^{00}) + \tilde{P}^{10}(\theta^{00}) - \tilde{P}^{01}(\theta^{00})) \varphi_h(\theta^{01}, \mathbf{x}). \end{aligned} \quad (6.13)$$

A short calculation exploiting the periodicity of the exponential function yields

$$\begin{aligned}\exp(i\boldsymbol{\theta}^{00}\boldsymbol{\kappa}) &= \begin{cases} \exp(i\boldsymbol{\theta}^{11}\boldsymbol{\kappa}) & \text{for } \boldsymbol{\kappa} \in J^{00} \cup J^{11}, \\ -\exp(i\boldsymbol{\theta}^{11}\boldsymbol{\kappa}) & \text{for } \boldsymbol{\kappa} \in J^{10} \cup J^{01}. \end{cases} \\ \exp(i\boldsymbol{\theta}^{00}\boldsymbol{\kappa}) &= \begin{cases} \exp(i\boldsymbol{\theta}^{10}\boldsymbol{\kappa}) & \text{for } \boldsymbol{\kappa} \in J^{00} \cup J^{01}, \\ -\exp(i\boldsymbol{\theta}^{10}\boldsymbol{\kappa}) & \text{for } \boldsymbol{\kappa} \in J^{11} \cup J^{10}. \end{cases} \\ \exp(i\boldsymbol{\theta}^{00}\boldsymbol{\kappa}) &= \begin{cases} \exp(i\boldsymbol{\theta}^{01}\boldsymbol{\kappa}) & \text{for } \boldsymbol{\kappa} \in J^{00} \cup J^{10}, \\ -\exp(i\boldsymbol{\theta}^{01}\boldsymbol{\kappa}) & \text{for } \boldsymbol{\kappa} \in J^{11} \cup J^{01}. \end{cases}\end{aligned}\quad (6.14)$$

Note that there is a direct correspondence between (6.14) and (6.7). From (6.14) it can be easily deduced that

$$\begin{aligned}\tilde{P}^{00}(\boldsymbol{\theta}^{00}) + \tilde{P}^{11}(\boldsymbol{\theta}^{00}) - \tilde{P}^{10}(\boldsymbol{\theta}^{00}) - \tilde{P}^{01}(\boldsymbol{\theta}^{00}) &= \sum_{\boldsymbol{\kappa} \in J} p_{\boldsymbol{\kappa}} \exp(i\boldsymbol{\theta}^{11}\boldsymbol{\kappa}) =: \tilde{P}_{2h}^h(\boldsymbol{\theta}^{11}), \\ \tilde{P}^{00}(\boldsymbol{\theta}^{00}) - \tilde{P}^{11}(\boldsymbol{\theta}^{00}) - \tilde{P}^{10}(\boldsymbol{\theta}^{00}) + \tilde{P}^{01}(\boldsymbol{\theta}^{00}) &= \sum_{\boldsymbol{\kappa} \in J} p_{\boldsymbol{\kappa}} \exp(i\boldsymbol{\theta}^{10}\boldsymbol{\kappa}) =: \tilde{P}_{2h}^h(\boldsymbol{\theta}^{10}), \\ \tilde{P}^{00}(\boldsymbol{\theta}^{00}) - \tilde{P}^{11}(\boldsymbol{\theta}^{00}) + \tilde{P}^{10}(\boldsymbol{\theta}^{00}) - \tilde{P}^{01}(\boldsymbol{\theta}^{00}) &= \sum_{\boldsymbol{\kappa} \in J} p_{\boldsymbol{\kappa}} \exp(i\boldsymbol{\theta}^{01}\boldsymbol{\kappa}) =: \tilde{P}_{2h}^h(\boldsymbol{\theta}^{01}).\end{aligned}\quad (6.15)$$

Inserting (6.15) into (6.13) establishes the Fourier representation for general prolongation operators w.r.t. the $2h$ -harmonics:

$$(P_{2h}^h \varphi_{2h}(2\boldsymbol{\theta}^{00}, \cdot))(x) = \frac{1}{4} \sum_{\boldsymbol{\alpha}} \tilde{P}_{2h}^h(\boldsymbol{\theta}^{\boldsymbol{\alpha}}) \varphi_h(\boldsymbol{\theta}^{\boldsymbol{\alpha}}, x) \quad (\boldsymbol{\theta} \in \Theta_{2h}, x \in G_h). \quad (6.16)$$

In particular, one obtains

$$P_{2h}^h : \text{span}\{\varphi_{2h}(2\boldsymbol{\theta}, \cdot)\} \longrightarrow \mathcal{F}_{2h}(\boldsymbol{\theta}) \quad (\boldsymbol{\theta} \in \Theta_{2h}).$$

Example 6.3 (Fourier symbol for bilinear interpolation). The Fourier symbol for bilinear interpolation is given by

$$\tilde{P}_{2h}^h(\boldsymbol{\theta}^{\boldsymbol{\alpha}}) = (1 + \cos(\theta_1^{\boldsymbol{\alpha}}))(1 + \cos(\theta_2^{\boldsymbol{\alpha}}));$$

compare with (6.10). ◀

The Fourier symbols for other prolongation operators can be calculated analogously. Several examples are given in [Table 6.2](#).

Remark 6.1 (Adjoint transfer operators). The Fourier symbols of the full weighting operator and bilinear interpolation—given in Examples 6.1 and 6.3—are related by

$$\tilde{R}_h^{2h}(\boldsymbol{\theta}^{\boldsymbol{\alpha}}) = \frac{1}{4} \tilde{P}_{2h}^h(\boldsymbol{\theta}^{\boldsymbol{\alpha}}). \quad (6.17)$$

TABLE 6.2: Fourier symbols for prolongation operators implemented in the accompanying software; $d = 2$

prolongation	Fourier symbol $\tilde{P}_{2h}^h(\boldsymbol{\theta}^\alpha)$
bilinear	$(1 + \cos(\theta_1^\alpha))(1 + \cos(\theta_2^\alpha))$
bicubic	$\frac{1}{64} (8 + 9 \cos(\theta_1^\alpha) - \cos(3\theta_1^\alpha))$ $(8 + 9 \cos(\theta_2^\alpha) - \cos(3\theta_2^\alpha))$
biquintic	$\frac{1}{16384} (128 - 150 \cos(\theta_1^\alpha) + 25 \cos(3\theta_1^\alpha) - 3 \cos(5\theta_1^\alpha))$ $(128 - 150 \cos(\theta_2^\alpha) + 25 \cos(3\theta_2^\alpha) - 3 \cos(5\theta_2^\alpha))$
constant upwind	$(1 + \exp(i\theta_1^\alpha))(1 + \exp(i\theta_2^\alpha))$
seven point	$1 + \cos(\theta_1^\alpha) + \cos(\theta_2^\alpha) + \cos(\theta_1^\alpha - \theta_2^\alpha)$

The same relation holds for higher-order weighting and bicubic interpolation and for seven-point restriction and seven-point interpolation, respectively. (6.17) is generally valid if prolongation and restriction are adjoint to each other. For d -dimensional transfer operators, the proper scaling is given by $1/2^d$. \blacktriangleright

Remark 6.2 (Operator-dependent transfer operators). Note that operator-dependent transfers can be represented by a fixed stencil for each $\mathbf{x} \in G_h$ as long as the underlying discrete operator L_h has constant coefficients. Regarding this matter it is also possible to apply local Fourier analysis to the popular class of operator-dependent transfer operators [18, 77]. \blacktriangleright

6.2.5 Fourier representation of coarse-grid discretization

We distinguish two different types of coarse-grid discretizations: Discretization coarse-grid approximation (DCA) and Galerkin coarse-grid approximation (GCA). In the first case, L_{2h} is obtained by a straightforward application of the fine-grid operator on the coarse grid whereas in the second case we have

$$L_{2h} = \check{R}_h^{2h} L_h \check{P}_{2h}^h$$

with Galerkin transfer operators \check{R}_h^{2h} and \check{P}_{2h}^h .

Example 6.4 (DCA and GCA for the Laplacian). DCA for the two-dimensional Laplacian is given by

$$L_{2h}^{\text{DCA}} \triangleq \frac{1}{4h^2} \begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 4 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix}_h = \frac{1}{4h^2} \begin{bmatrix} & -1 \\ -1 & 4 & -1 \\ & -1 \end{bmatrix}_{2h}.$$

GCA based on full weighting (\tilde{R}_h^{2h}) and bilinear interpolation (\tilde{P}_{2h}^h) reads

$$L_{2h}^{\text{GCA}} \triangleq \frac{1}{4h^2} \begin{bmatrix} -1/4 & 0 & -1/2 & 0 & -1/4 \\ 0 & 0 & 0 & 0 & 0 \\ -1/2 & 0 & 3 & 0 & -1/2 \\ 0 & 0 & 0 & 0 & 0 \\ -1/4 & 0 & -1/2 & 0 & -1/4 \end{bmatrix}_h = \frac{1}{4h^2} \begin{bmatrix} -1/4 & -1/2 & -1/4 \\ -1/2 & 3 & -1/2 \\ -1/4 & -1/2 & -1/4 \end{bmatrix}_{2h}$$

The stencil representation of L_{2h}^{GCA} is calculated using (1.9). ◀

Both types of operators can be represented by *constant* coefficient stencils, $L_{2h} \triangleq [\ell_\kappa]_{2h}$, as long as L_h , \tilde{R}_h^{2h} , and \tilde{P}_{2h}^h are constant coefficient operators as well. Hence, the coarse-grid Fourier components are eigenfunctions of L_{2h} yielding

$$L_{2h} : \text{span}\{\varphi_{2h}(2\theta^{00}, \cdot)\} \longrightarrow \text{span}\{\varphi_{2h}(2\theta^{00}, \cdot)\} \quad (\theta^{00} \in \Theta_{2h}).$$

The corresponding Fourier symbols read

$$\tilde{L}_{2h}(\theta^{00}) = \sum_{\kappa \in J} \ell_\kappa \exp(i 2\theta^{00} \kappa) \quad (\theta^{00} \in \Theta_{2h}),$$

similar as for the fine-grid operator.

Example 6.5 (Fourier symbols for coarse-grid discretizations). The Fourier symbols for the coarse-grid operators from Example 6.4 are given by

$$\begin{aligned} \tilde{L}_{2h}^{\text{DCA}}(2\theta^{00}) &= \frac{1}{4h^2} (4 - 2 \cos(2\theta_1^{00}) - 2 \cos(2\theta_2^{00})) \quad \text{and} \\ \tilde{L}_{2h}^{\text{GCA}}(2\theta^{00}) &= \frac{1}{4h^2} (3 - \cos(2\theta_1^{00}) - \cos(2\theta_2^{00}) \\ &\quad - \frac{1}{2} \cos(2(\theta_1^{00} + \theta_2^{00})) - \frac{1}{2} \cos(2(\theta_1^{00} - \theta_2^{00}))). \end{aligned}$$

Remark 6.3 (Fourier symbol for Galerkin coarse-grid approximation). There is an alternative way to calculate the Fourier symbol for a Galerkin coarse-grid discretization. Instead of computing the stencil representation of L_{2h}^{GCA} , $\tilde{L}_{2h}^{\text{GCA}}(2\theta^{00})$ might also be obtained using the Fourier representations of L_h , \tilde{R}_h^{2h} , and \tilde{P}_{2h}^h :

$$\begin{aligned} \tilde{L}_{2h}^{\text{GCA}}(2\theta^{00}) &= \\ &\begin{pmatrix} \tilde{R}_h^{2h}(2\theta^{00}) \\ \tilde{R}_h^{2h}(2\theta^{11}) \\ \tilde{R}_h^{2h}(2\theta^{10}) \\ \tilde{R}_h^{2h}(2\theta^{01}) \end{pmatrix}^T \begin{pmatrix} \tilde{L}_h(2\theta^{00}) & 0 & 0 & 0 \\ 0 & \tilde{L}_h(2\theta^{11}) & 0 & 0 \\ 0 & 0 & \tilde{L}_h(2\theta^{10}) & 0 \\ 0 & 0 & 0 & \tilde{L}_h(2\theta^{01}) \end{pmatrix} \frac{1}{4} \begin{pmatrix} \tilde{P}_{2h}^h(2\theta^{00}) \\ \tilde{P}_{2h}^h(2\theta^{11}) \\ \tilde{P}_{2h}^h(2\theta^{10}) \\ \tilde{P}_{2h}^h(2\theta^{01}) \end{pmatrix}. \end{aligned}$$

Of course, both approaches yield the same results. ▶

6.2.6 Invariance property of the two-grid operator

Combining the above results concerning the different operators constituting the coarse-grid correction leads to the invariance property of the complete two-grid operator with respect to the $2h$ -harmonics.

In order to ensure that we deal with nonvanishing Fourier symbols $\tilde{L}_h(\boldsymbol{\theta})$ and $\tilde{L}_{2h}(2\boldsymbol{\theta})$, we restrict our considerations to the following slightly shrunken subspace (see also [60]):

$$\mathcal{F}_{2g} := \mathcal{F} \setminus \bigcup_{\boldsymbol{\theta} \in \Psi_{2g}} \mathcal{F}_{2h}(\boldsymbol{\theta}) \quad \text{with} \\ \Psi_{2g} := \left\{ \boldsymbol{\theta} \in \Theta_{2h} \mid \tilde{L}_{2h}(2\boldsymbol{\theta}^{00}) = 0 \text{ or } \tilde{L}_h(\boldsymbol{\theta}^\alpha) = 0 \right\}.$$

The crucial observation is that the coarse-grid correction operator K_h^{2h} (6.1) leaves the spaces of $2h$ -harmonics invariant for an arbitrary Fourier frequency $\boldsymbol{\theta} \in \Theta_{2g} := \Theta_{2h} \setminus \Psi_{2g}$; that means

$$K_h^{2h} = I_h - P_{2h}^h L_{2h}^{-1} R_h^{2h} L_h : \mathcal{F}_{2h}(\boldsymbol{\theta}) \longrightarrow \mathcal{F}_{2h}(\boldsymbol{\theta}) \quad (\boldsymbol{\theta} \in \Theta_{2g}).$$

This invariance property is due to the previously derived relations

$$\begin{aligned} I_h, L_h &: \mathcal{F}_{2h}(\boldsymbol{\theta}) \longrightarrow \mathcal{F}_{2h}(\boldsymbol{\theta}), \\ R_h^{2h} &: \mathcal{F}_{2h}(\boldsymbol{\theta}) \longrightarrow \text{span}\{\varphi_{2h}(\boldsymbol{\theta}^{00}, \cdot)\}, \\ L_{2h} &: \text{span}\{\varphi_{2h}(\boldsymbol{\theta}^{00}, \cdot)\} \longrightarrow \text{span}\{\varphi_{2h}(\boldsymbol{\theta}^{00}, \cdot)\}, \\ P_{2h}^h &: \text{span}\{\varphi_{2h}(\boldsymbol{\theta}^{00}, \cdot)\} \longrightarrow \mathcal{F}_{2h}(\boldsymbol{\theta}). \end{aligned} \tag{6.18}$$

From Chapter 5 it is known that the same invariance property holds for many relaxation methods (JAC, GS-LEX, pattern relaxations, etc.), i.e.,

$$S_h(\omega) : \mathcal{F}_{2h}(\boldsymbol{\theta}) \longrightarrow \mathcal{F}_{2h}(\boldsymbol{\theta}) \quad (\boldsymbol{\theta} \in \Theta_{2g}). \tag{6.19}$$

For the concrete Fourier representations $S_h(\omega)|_{\mathcal{F}_{2h}(\boldsymbol{\theta})} \stackrel{\wedge}{=} S^{2g}(\boldsymbol{\theta}) \in \mathbb{C}^{4 \times 4}$ of several relaxation methods we refer to Chapter 5 and Appendix A.1.

Summarizing, we have that the two-grid operator (6.1) leaves the spaces of $2h$ -harmonics invariant:

$$M_h^{2h} : \mathcal{F}_{2h}(\boldsymbol{\theta}) \longrightarrow \mathcal{F}_{2h}(\boldsymbol{\theta}) \quad (\boldsymbol{\theta} \in \Theta_{2g}).$$

More precisely, for each $\boldsymbol{\theta} \in \Theta_{2g}$ the Fourier representation of the two-grid operator reads:

$$\begin{aligned} M_h^{2h}|_{\mathcal{F}_{2h}(\boldsymbol{\theta})} &\stackrel{\wedge}{=} M^{2g}(\boldsymbol{\theta}) \in \mathbb{C}^{4 \times 4} \quad \text{with} \\ M^{2g}(\boldsymbol{\theta}) &= (S^{2g}(\boldsymbol{\theta}))^{\nu_2} (I^{2g} - P^{2g}(\boldsymbol{\theta})(\mathcal{L}^{2g}(\boldsymbol{\theta}))^{-1} R^{2g}(\boldsymbol{\theta}) L^{2g}(\boldsymbol{\theta})) (S^{2g}(\boldsymbol{\theta}))^{\nu_1}, \end{aligned} \tag{6.20}$$

where the representation of the coarse-grid correction block matrices is given by

$$\begin{aligned}
 I^{2g} &= \text{diag}\{1, 1, 1, 1\} \in \mathbb{C}^{4 \times 4}, \quad \mathcal{L}^{2g}(\boldsymbol{\theta}) = \tilde{L}_{2h}(2\boldsymbol{\theta}^{00}) \in \mathbb{C}^{1 \times 1}, \\
 L^{2g}(\boldsymbol{\theta}) &= \text{diag}\{\tilde{L}_h(\boldsymbol{\theta}^{00}), \tilde{L}_h(\boldsymbol{\theta}^{11}), \tilde{L}_h(\boldsymbol{\theta}^{10}), \tilde{L}_h(\boldsymbol{\theta}^{01})\} \in \mathbb{C}^{4 \times 4}, \\
 R^{2g}(\boldsymbol{\theta}) &= \left(\tilde{R}_h^{2h}(\boldsymbol{\theta}^{00}) \quad \tilde{R}_h^{2h}(\boldsymbol{\theta}^{11}) \quad \tilde{R}_h^{2h}(\boldsymbol{\theta}^{10}) \quad \tilde{R}_h^{2h}(\boldsymbol{\theta}^{01}) \right) \in \mathbb{C}^{1 \times 4}, \\
 P^{2g}(\boldsymbol{\theta}) &= \frac{1}{4} \left(\tilde{P}_{2h}^h(\boldsymbol{\theta}^{00}) \quad \tilde{P}_{2h}^h(\boldsymbol{\theta}^{11}) \quad \tilde{P}_{2h}^h(\boldsymbol{\theta}^{10}) \quad \tilde{P}_{2h}^h(\boldsymbol{\theta}^{01}) \right)^T \in \mathbb{C}^{4 \times 1}.
 \end{aligned} \tag{6.21}$$

6.2.7 Definition of the two-grid convergence factor

Using the block representation from (6.20), the spectral radius of the two-grid iteration matrix and thus the asymptotic two-grid or two-level (2L) convergence factor can be approximated by

$$\rho(M_{2L}) := \sup_{\boldsymbol{\theta} \in \Theta_{2g}} \rho(M^{2g}(\boldsymbol{\theta})). \tag{6.22}$$

Similarly, one might estimate the spectral norm of the error-reduction operator and the defect-reduction operator by

$$\begin{aligned}
 \|M_{2L}\|_s &:= \sup_{\boldsymbol{\theta} \in \Theta_{2g}} \sqrt{\rho(M^{2g}(\boldsymbol{\theta})(M^{2g}(\boldsymbol{\theta}))^*)}, \\
 \|M_{2L}\|_d &:= \sup_{\boldsymbol{\theta} \in \Theta_{2g}} \sqrt{\rho(L^{2g}(\boldsymbol{\theta})M^{2g}(\boldsymbol{\theta})(L^{2g}(\boldsymbol{\theta}))^{-1}(L^{2g}(\boldsymbol{\theta})M^{2g}(\boldsymbol{\theta})(L^{2g}(\boldsymbol{\theta}))^{-1})^*)}.
 \end{aligned}$$

Remark 6.4 (Boundedness of two-grid factors). In all examples considered in this monograph, we have $\Theta_{2g} = (-\pi/2, \pi/2] \setminus \{(0, 0)\}$, as only $\tilde{L}_h((0, 0))$ and $\tilde{L}_{2h}(2(0, 0))$ are zero. However, the supremum in (6.22) remains finite, since $\tilde{R}_h^{2h}((0, 0))\tilde{L}_h((0, 0))$ is rank deficient, too, for all restrictions under consideration, in such a way that $\lim_{\boldsymbol{\theta} \rightarrow (0, 0)} \rho(M^{2g}(\boldsymbol{\theta}))$ is bounded; see [9]. This is, however, not generally true for the norm values $\|M_{2L}\|_s$ and $\|M_{2L}\|_d$. For example, in connection with injection it may happen that $\|M_{2L}\|_s$ is unbounded; compare with Example 6.6. Such phenomena are directly related to the “high-frequency orders” of the transfer operators under consideration which are discussed in Section 7.1.2. ►

Remark 6.5 (Error versus residual reduction). The *residual*-reduction norm $\|M_{2L}\|_d$ decreases with an increasing number of postrelaxations ν_2 in contrast to the *error*-reduction norm $\|M_{2L}\|_s$ which decreases with an increasing ν_1 . According to this observation, one should choose similar values for ν_1 and ν_2 and rather $\nu_1 \geq \nu_2$ than $\nu_1 \leq \nu_2$ when multigrid is selected as a solver [60]. This observation carries over to the related three-grid values derived in Section 6.5. ►

The numerical calculation of $\rho(M_{2L})$, $\|M_{2L}\|_s$, and $\|M_{2L}\|_d$ provided by **LFA** is based on the discrete set $\Theta_{2g}^P := \Theta_{2g} \cap G_{h_\theta}$ in an analogy to the smoothing analysis, see Remarks 5.2 and 5.4. Then, the two-grid analysis becomes an exact analysis for certain model problems on rectangular domains with periodic boundary conditions. Pure periodic boundary conditions lead to a singular boundary value problem in general. This necessitates a compatibility condition for every iterative solution method (see, for example, [62]), which directly corresponds to the exclusion of the *zero* frequency $(0, 0)$ in the analysis, as discussed in the preceding Remark.

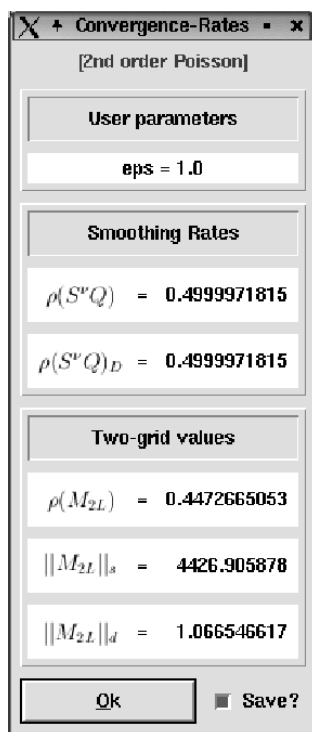


FIGURE 6.3: Two-grid factors for the Poisson equation; $d = 2$.

Example 6.6 (Two-grid convergence for the Laplacian). A two-grid method for the second-order central discretization of the two-dimensional Laplacian ($h = 1/256$) consisting of point GS-LEX prerelaxation, no postrelaxation ($\nu_1 = 1$, $\nu_2 = 0$), injection, bilinear interpolation, and DCA yields an asymptotic two-grid convergence factor of 0.5, shown in Figure 6.3. The corresponding error-reduction norm is actually unbounded. That is, for $h \rightarrow 0$

(and consequently $h_{\theta} \rightarrow 0$), we have $\|M_{2L}\|_s \rightarrow \infty$. Obviously, this limit cannot be obtained by the numerical software since the explicit calculation of $\|M_{2L}\|_s$ is based on a *fixed* h_{θ} ($= 2\pi h = \pi/128$ in the present example). However, the information provided by **xlfa** ($\|M_{2L}\|_s > 0.4 \cdot 10^4$) is of course sufficiently significant. \blacktriangleleft

Remark 6.6 (Analytical formulae for two-grid factors). Two-grid factors for certain simple examples can be obtained analytically as well [60, 62]. However, apart from such very special situations, the calculation of two-grid factors has to be done by numerical software. \blacktriangleright

6.2.8 Semicoarsening

Next, we consider the necessary modifications for the two-grid analysis in case of semicoarsening. For the definition of the spaces of harmonics we use the same criterion as in Section 6.2.1. That is, Fourier components $\varphi_h(\theta^{\alpha}, \cdot)$ that alias on the coarse grid are gathered in one subspace $\mathcal{F}_H(\theta)$.

Definition 6.2 (Spaces of harmonics for semicoarsening, $d = 2$). The spaces of harmonics for x_1 - and x_2 -semicoarsening are given by

$$\begin{aligned} \mathcal{F}_{2h,h}(\theta) &:= \text{span}\{ \varphi_h(\theta, \cdot), \varphi_h(\theta - (\text{sign}(\theta_1), 0)\pi, \cdot) \} \\ \text{with } \theta \in \Theta_{2h,h} &:= \{ \theta \in \Theta \mid -\pi/2 < \theta_1 \leq \pi/2 \} \quad (= \Theta_{\text{low}}), \\ \mathcal{F}_{h,2h}(\theta) &:= \text{span}\{ \varphi_h(\theta, \cdot), \varphi_h(\theta - (0, \text{sign}(\theta_2))\pi, \cdot) \} \\ \text{with } \theta \in \Theta_{h,2h} &:= \{ \theta \in \Theta \mid -\pi/2 < \theta_2 \leq \pi/2 \} \quad (= \Theta_{\text{low}}), \end{aligned}$$

respectively. Note that in both cases $\Theta_H = \Theta_{\text{low}}$ (*low* with respect to the particular coarsening strategy under consideration) holds. \blacktriangle

Remark 6.7 Spaces of harmonics for red-black coarsening and quadrupling). Similarly, one might define the spaces of harmonics for red-black coarsening and quadrupling. For red-black coarsening, we have

$$\begin{aligned} \mathcal{F}_{\sqrt{2}h}(\theta) &:= \text{span}\{ \varphi_h(\theta, \cdot), \varphi_h(\theta - (\text{sign}(\theta_1), \text{sign}(\theta_2))\pi, \cdot) \} \\ \text{with } \theta \in \Theta_{\sqrt{2}h} &:= \{ \theta \in \Theta \mid |\theta| \leq \pi \} \quad (= \Theta_{\text{low}}), \end{aligned}$$

whereas in the case of quadrupling, the spaces of harmonics coincide with the $4h$ -harmonics $\mathcal{F}_{4h}(\theta)$ ($\theta \in \Theta_{4h} := (-\pi/4, \pi/4]^2 = \Theta_{\text{low}}$) from Definition 6.3 in connection with Fourier three-grid analysis (see below). \blacktriangleright

Now it can be shown in the same way as it has been done above for standard coarsening, that the coarse-grid correction operator K_h^H based on semicoarsening leaves the related spaces of harmonics invariant. For example, for x_1 -semicoarsening ($H = (2h, h)$) we obtain

$$K_h^H : \mathcal{F}_{2h,h}(\theta) \longrightarrow \mathcal{F}_{2h,h}(\theta)$$

which is due to

$$\begin{aligned} I_h, L_h &: \mathcal{F}_{2h,h}(\boldsymbol{\theta}) \longrightarrow \mathcal{F}_{2h,h}(\boldsymbol{\theta}), \\ R_h^H &: \mathcal{F}_{2h,h}(\boldsymbol{\theta}) \longrightarrow \text{span}\{\varphi_H((2\theta_1, \theta_2), \cdot)\}, \\ L_H &: \text{span}\{\varphi_H((2\theta_1, \theta_2), \cdot)\} \longrightarrow \text{span}\{\varphi_H((2\theta_1, \theta_2), \cdot)\}, \\ P_H^h &: \text{span}\{\varphi_H((2\theta_1, \theta_2), \cdot)\} \longrightarrow \mathcal{F}_{2h,h}(\boldsymbol{\theta}). \end{aligned}$$

The invariance properties for the discretization operators L_h and L_H are trivial. The treatment of the transfer operators carries over from standard coarsening yielding analog formulae for their Fourier symbols. In fact it is even less complicated since we are dealing with “one-dimensional” transfer operators regarding semicoarsening for two-dimensional applications. As a consequence, we only have to distinguish between two types of grid points for the investigation of the prolongation operator instead of four types of grid points in case of standard coarsening; compare with Section 6.2.4.

Example 6.7 (Full weighting and linear interpolation for x_1 -semicoarsening, $d = 2$). Linear interpolation for x_1 -semicoarsening is defined by

$$(P_H^h v_H)(x) = \begin{cases} v_H(x) & \text{for } x \in G_h^{00} \cup G_h^{01} \\ \frac{1}{2}(v_H(x+h, y) + v_H(x-h, y)) & \text{for } x \in G_h^{10} \cup G_h^{11} \end{cases}$$

for a coarse-grid function $v_H : G_H \rightarrow \mathbb{C}$. Note that $G_H = G_h^{00} \cup G_h^{01}$ and $G_h \setminus G_H = G_h^{10} \cup G_h^{11}$ hold. The stencil representations for full weighting and linear interpolation are given by

$$R_h^H \triangleq \frac{1}{4} \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}_h^H \quad \text{and} \quad P_H^h \triangleq \frac{1}{2} \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}_{\frac{h}{H}},$$

respectively. The corresponding Fourier symbols read

$$\tilde{R}_h^H(\boldsymbol{\theta}) = \frac{1}{2}(1 + \cos(\theta_1)) \quad \text{and} \quad \tilde{P}_H^h(\boldsymbol{\theta}) = 1 + \cos(\theta_1);$$

compare with (6.4) and (6.15). ◀

The Fourier symbols for implemented transfer operators are shown in [Table 6.3](#).

Remark 6.8 (Fourier symbols for d -dimensional transfer operators).

Many well-known d -dimensional transfer operators are defined by tensor products of one-dimensional transfers. Popular examples are full weighting or bi- and trilinear interpolation as discussed in Section 3.4.5. The Fourier symbol of such a d -dimensional transfer operator is given by the product of Fourier symbols of the related one-dimensional transfers. For example, the Fourier symbol of d -dimensional full weighting reads

$$\tilde{R}_h^{2h}(\boldsymbol{\theta}^\alpha) = \prod_{j=1}^d \frac{1}{2} (1 + \cos(\theta_j^\alpha)).$$

TABLE 6.3: Fourier symbols for transfer operators implemented in the accompanying software; x_j -semicoarsening ($j = 1, 2$), $d = 2$

prolongation	Fourier symbol $\tilde{P}_h^H(\theta^\alpha)$
linear	$(1 + \cos(\theta_j^\alpha))$
cubic	$\frac{1}{8} (8 + 9 \cos(\theta_j^\alpha) - \cos(3\theta_j^\alpha))$
quintic	$\frac{1}{128} (128 - 150 \cos(\theta_j^\alpha) + 25 \cos(3\theta_j^\alpha) - 3 \cos(5\theta_j^\alpha))$
constant upwind	$1 + \exp(i\theta_j^\alpha)$
restriction	Fourier symbol $\tilde{R}_h^H(\theta^\alpha)$
full weighting	$\frac{1}{2} (1 + \cos(\theta_j^\alpha))$
injection	1
higher-order	$\frac{1}{16} (8 + 9 \cos(\theta_j^\alpha) - \cos(3\theta_j^\alpha))$

More examples can be found by comparing [Tables 6.1](#), [6.2](#), and [6.5](#) (referring to the two- and three-dimensional cases) with [Table 6.3](#). ►

Summarizing, a block diagonal representation of K_h^H consists of (2×2) -blocks. For JAC- and GS-LEX-type relaxations this means that the complete two-grid operator is invariant under the spaces of harmonics since the Fourier components are eigenfunctions of these smoothing methods (see [Section 5.3](#)).

However, for pattern relaxation methods the minimal invariant subspaces differ for the different methods, which has already been discussed in [Remark 5.6](#) for three-dimensional RB-JAC smoothers. The minimal invariant subspaces for two-dimensional pattern relaxations are specified in [Table 6.4](#).

TABLE 6.4: Minimal invariant subspaces for two-dimensional pattern relaxations

point RB-JAC, point RB-GS:	$\mathcal{F}_{\sqrt{2}h}(\theta)$	$(\theta \in \Theta_{\sqrt{2}h})$
4C-JAC, alternating line RB-JAC, 4C-GS, alternating line RB-GS:	$\mathcal{F}_{2h}(\theta)$	$(\theta \in \Theta_{2h})$
x_1 -line RB-JAC, x_1 -line RB-GS:	$\mathcal{F}_{h,2h}(\theta)$	$(\theta \in \Theta_{h,2h})$
x_2 -line RB-JAC, x_2 -line RB-GS:	$\mathcal{F}_{2h,h}(\theta)$	$(\theta \in \Theta_{2h,h})$

As a consequence, it is not possible to combine a Fourier representation (w.r.t. the harmonics from [Definition 6.2](#)) of K_h^H with the Fourier represen-

tation of an arbitrary pattern relaxation w.r.t. its minimal invariant subspace. For example, x_1 -semicoarsening and point RB-JAC relaxation are not compatible regarding this matter because the minimal invariant subspaces of the corresponding operators K_h^H and S_h are different. Therefore it is convenient to derive the Fourier representation of the coarse-grid correction operator in case of semicoarsening using the $2h$ -harmonics, as it has already been done for the relaxation methods in the previous chapter. Then both Fourier representations are compatible allowing for an easy calculation of two-grid factors.

Obviously, each space of $2h$ -harmonics is composed of two spaces of harmonics referring to semicoarsening. More precisely, we have

$$\begin{aligned}\mathcal{F}_{2h}(\boldsymbol{\theta}) &= \mathcal{F}_{2h,h}(\boldsymbol{\theta}^{00}) \cup \mathcal{F}_{2h,h}(\boldsymbol{\theta}^{01}) \quad (\boldsymbol{\theta}^{00} \in \Theta_{2h}, \boldsymbol{\theta}^{01} \in \Theta_{2h,h} \setminus \Theta_{2h}) \\ &= \text{span}\{\varphi_h(\boldsymbol{\theta}^{00}, \cdot), \varphi_h(\boldsymbol{\theta}^{10}, \cdot)\} \cup \text{span}\{\varphi_h(\boldsymbol{\theta}^{01}, \cdot), \varphi_h(\boldsymbol{\theta}^{11}, \cdot)\} \\ \mathcal{F}_{2h}(\boldsymbol{\theta}) &= \mathcal{F}_{h,2h}(\boldsymbol{\theta}^{00}) \cup \mathcal{F}_{h,2h}(\boldsymbol{\theta}^{10}) \quad (\boldsymbol{\theta}^{00} \in \Theta_{2h}, \boldsymbol{\theta}^{10} \in \Theta_{h,2h} \setminus \Theta_{2h}) \\ &= \text{span}\{\varphi_h(\boldsymbol{\theta}^{00}, \cdot), \varphi_h(\boldsymbol{\theta}^{01}, \cdot)\} \cup \text{span}\{\varphi_h(\boldsymbol{\theta}^{10}, \cdot), \varphi_h(\boldsymbol{\theta}^{11}, \cdot)\},\end{aligned}\tag{6.23}$$

for x_1 - and x_2 -semicoarsening, respectively. This observation is illustrated in Figure 6.4 showing sets of Fourier frequencies generating the different spaces of harmonics. Note that $\boldsymbol{\theta}^{00}, \boldsymbol{\theta}^{01}$ are low frequencies considering x_1 -semicoarsening, whereas for x_2 -semicoarsening $\boldsymbol{\theta}^{00}, \boldsymbol{\theta}^{10}$ are low frequencies.

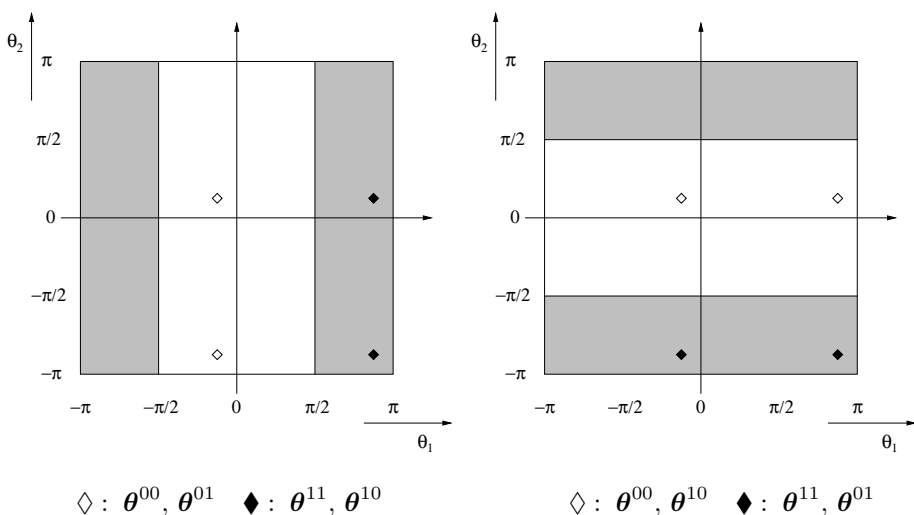


FIGURE 6.4: Sets of frequencies generating \mathcal{F}_{2h} , $\mathcal{F}_{2h,h}$, and $\mathcal{F}_{h,2h}$. x_1 -semicoarsening (left) and x_2 -semicoarsening (right).

Combining (6.23) and the invariance property of K_h^H w.r.t. the harmonics from Definition 6.2 one might easily establish that the Fourier representation of K_h^H is given by the following (4×4) -matrices in case of x_1 -semicoarsening:

$$\begin{aligned}
 K_h^H|_{\mathcal{F}_{2h}(\theta)} &\triangleq K^{2g}(\theta) \in \mathbb{C}^{4 \times 4} \quad \text{with} \\
 K^{2g}(\theta) &= I^{2g} - P^{2g}(\theta)(\mathcal{L}^{2g}(\theta))^{-1}R^{2g}(\theta)L^{2g}(\theta) \quad \text{and} \\
 I^{2g} &= \text{diag}\{1, 1, 1, 1\} \in \mathbb{C}^{4 \times 4}, \\
 P^{2g}(\theta) &= \frac{1}{2} \begin{pmatrix} \tilde{P}_H^h(\theta^{00}) & 0 \\ 0 & \tilde{P}_H^h(\theta^{11}) \\ \tilde{P}_H^h(\theta^{10}) & 0 \\ 0 & \tilde{P}_H^h(\theta^{01}) \end{pmatrix} \in \mathbb{C}^{4 \times 2}, \\
 \mathcal{L}^{2g}(\theta) &= \text{diag}\{\tilde{L}_H(2\theta_1^{00}, \theta_2^{00}), \tilde{L}_H(2\theta_1^{01}, \theta_2^{01})\} \in \mathbb{C}^{2 \times 2}, \\
 R^{2g}(\theta) &= \begin{pmatrix} \tilde{R}_h^H(\theta^{00}) & 0 & \tilde{R}_h^H(\theta^{10}) & 0 \\ 0 & \tilde{R}_h^H(\theta^{11}) & 0 & \tilde{R}_h^H(\theta^{01}) \end{pmatrix} \in \mathbb{C}^{2 \times 4}, \\
 L^{2g}(\theta) &= \text{diag}\{\tilde{L}_h(\theta^{00}), \tilde{L}_h(\theta^{11}), \tilde{L}_h(\theta^{10}), \tilde{L}_h(\theta^{01})\} \in \mathbb{C}^{4 \times 4}.
 \end{aligned} \tag{6.24}$$

We would like to point out that the Fourier symbols $\tilde{P}_H^h(\theta^\alpha)$ are scaled with $1/2$ (instead of $1/4$ in connection with standard coarsening) according to the “one-dimensional” prolongation operator (see Remark 6.9 for the d -dimensional case).

For x_2 -semicoarsening ($H = (h, 2h)$), the Fourier representation of the transfer operators and the coarse-grid discretization has to be changed as follows:

$$\begin{aligned}
 P^{2g}(\theta) &= \frac{1}{2} \begin{pmatrix} \tilde{P}_H^h(\theta^{00}) & 0 \\ 0 & \tilde{P}_H^h(\theta^{11}) \\ 0 & \tilde{P}_H^h(\theta^{10}) \\ \tilde{P}_H^h(\theta^{01}) & 0 \end{pmatrix} \in \mathbb{C}^{4 \times 2}, \\
 \mathcal{L}^{2g}(\theta) &= \text{diag}\{\tilde{L}_H(\theta_1^{00}, 2\theta_2^{00}), \tilde{L}_H(\theta_1^{10}, 2\theta_2^{10})\} \in \mathbb{C}^{2 \times 2}, \\
 R^{2g}(\theta) &= \begin{pmatrix} \tilde{R}_h^H(\theta^{00}) & 0 & 0 & \tilde{R}_h^H(\theta^{01}) \\ 0 & \tilde{R}_h^H(\theta^{11}) & \tilde{R}_h^H(\theta^{10}) & 0 \end{pmatrix} \in \mathbb{C}^{2 \times 4}.
 \end{aligned}$$

Altogether, we obtain the Fourier representation of the complete two-grid operator using the $2h$ -harmonics:

$$M_h^H|_{\mathcal{F}_{2h}(\theta)} \triangleq (S^{2g}(\theta))^{\nu_2} K^{2g}(\theta) (S^{2g}(\theta))^{\nu_1},$$

and the definitions of the two-grid factors $(\rho(M_{2L}), \|M_{2L}\|_s, \|M_{2L}\|_d)$ carry over from standard coarsening. The set of Fourier frequencies that have to be removed from the analysis has to be modified in case of semicoarsening

according to

$$\Psi_{2g} := \{\theta \in \Theta_{2h} \mid \tilde{L}_H((2\theta_1^{00}, \theta_2^{00})) = 0 \text{ or } \tilde{L}_H((2\theta_1^{01}, \theta_2^{01})) = 0 \text{ or } \tilde{L}_h(\theta^\alpha) = 0\},$$

$$\Psi_{2g} := \{\theta \in \Theta_{2h} \mid \tilde{L}_H((\theta_1^{00}, 2\theta_2^{00})) = 0 \text{ or } \tilde{L}_H((\theta_1^{10}, 2\theta_2^{10})) = 0 \text{ or } \tilde{L}_h(\theta^\alpha) = 0\},$$

for x_1 - and x_2 -semicoarsening, respectively.

Example 6.8 (x_1 -semicoarsening for the anisotropic diffusion equation, $d = 2$). Consider the second-order discretization of the anisotropic diffusion equation from Section 4.1.1 with mesh size $h = 1/128$. For a severe anisotropy, governed by the parameter $\varepsilon = 100$, it is appropriate to apply x_1 -semicoarsening, linear interpolation, full weighting, DCA, and two sweeps of point RB-JAC relaxation (one pre- and one postrelaxation step). The resulting two-grid convergence is shown in Figure 6.5 ◀

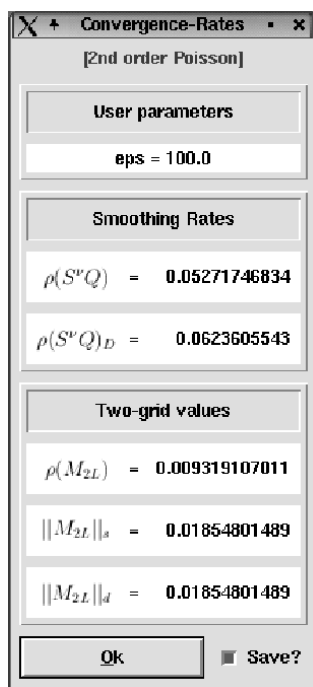


FIGURE 6.5: Two-grid factors for x_1 -semicoarsening applied to the anisotropic diffusion equation; $d = 2$.

6.3 TWO-GRID ANALYSIS FOR 3D SCALAR PROBLEMS

Fourier two-grid analysis for three-dimensional problems is a straightforward modification of the previously introduced analysis for the two-dimensional case. Accordingly, we focus on the necessary adjustments.

6.3.1 Standard coarsening

The definition of the spaces of $2h$ -harmonics for three-dimensional applications is contained in Definition 5.8 covering the d -dimensional case. They have been extensively used in Section 5.4 in connection with smoothing analysis for pattern relaxations. For three-dimensional applications, they consist of one low frequency and seven high frequencies. More generally, in d dimensions we have one low frequency and $2^d - 1$ high frequencies. Again, the spaces of $2h$ -harmonics turn out to be invariant under the two-grid operator and—as a consequence—the two grid factors are given by the suprema of the spectral radii of certain block matrices. Here, we are dealing with (8×8) -blocks due to the dimensionality of $\mathcal{F}_{2h}(\theta)$ ($\theta \in \Theta_{2h} = (-\pi/2, \pi/2]^3$).

In detail, the invariance property of the two-grid operator can be obtained by an investigation of the different multigrid components yielding analog relations as they are given in (6.18) and (6.19) for two dimensions. The Fourier representation of the smoother with respect to the $2h$ -harmonics, $S_h(\omega)|_{\mathcal{F}_{2h}(\theta)} \hat{=} S^{2g}(\theta)$ ($\theta \in \Theta_{2h}$), has been computed in Chapter 5 for many relaxation methods. The Fourier representation of the remaining multigrid components constituting the coarse-grid correction can be derived analogously to Sections 6.2.2 - 6.2.5.

The treatment of fine- and coarse-grid discretizations is trivial again since the Fourier components are eigenfunctions for these operators. For the restriction ($R_h^{2h} \hat{=} [r_\kappa]_h^{2h}$),

$$(R_h^{2h} \varphi_h(\theta^\alpha, \cdot))(x) = \tilde{R}_h^{2h}(\theta^\alpha) \varphi_{2h}(2\theta^{000}, x) \quad (x \in G_{2h})$$

results with a Fourier symbol $\tilde{R}_h^{2h}(\theta^\alpha) = \sum_{\kappa \in J} r_\kappa \exp(i\kappa\theta^\alpha)$ similar to that in (6.4). Several Fourier symbols for three-dimensional restriction operators can be found in Table 6.5.

Prolongation of three-dimensional grid functions distinguishes eight different types of grid points $x \in G_h$ (5.17). For a general treatment of prolongation operators $P_{2h}^h \hat{=} p_\kappa [p_\kappa]_h^{2h}$ in three dimensions, one has to expand the considerations from Section 6.2.4. This can be done in a straightforward way by applying the analog definitions and relations from Section 5.4.3 leading to

$$(P_{2h}^h \varphi_{2h}(2\theta^{000}, \cdot))(x) = \frac{1}{8} \sum_{\alpha} \tilde{P}_{2h}^h(\theta^\alpha) \varphi_h(\theta^\alpha, x) \quad (\theta \in \Theta_{2h}, x \in G_h)$$

TABLE 6.5: Fourier symbols for transfer operators implemented in the accompanying software, $d = 3$

restriction	Fourier symbol $\tilde{R}_h^{2h}(\boldsymbol{\theta}^\alpha)$
full weighting	$\prod_{j=1}^3 \frac{1}{2}(1 + \cos(\theta_j^\alpha))$
half weighting	$\frac{1}{6}(3 + \cos(\theta_1^\alpha) + \cos(\theta_2^\alpha) + \cos(\theta_3^\alpha))$
injection	1
higher order	$\prod_{j=1}^3 \frac{1}{16}(8 + 9\cos(\theta_j^\alpha) - \cos(3\theta_j^\alpha))$
adjoint of linear	$\frac{1}{8}(1 + \cos(\theta_1^\alpha) + \cos(\theta_2^\alpha) + \cos(\theta_2^\alpha) + \cos(\theta_1^\alpha - \theta_2^\alpha) \\ + \cos(\theta_1^\alpha - \theta_3^\alpha) + \cos(\theta_2^\alpha - \theta_3^\alpha) + \cos(\theta_1^\alpha + \theta_3^\alpha - \theta_2^\alpha))$
prolongation	Fourier symbol $\tilde{P}_{2h}^h(\boldsymbol{\theta}^\alpha)$
trilinear	$\prod_{j=1}^3 (1 + \cos(\theta_j^\alpha))$
tricubic	$\prod_{j=1}^3 \frac{1}{8}(8 + 9\cos(\theta_j^\alpha) - \cos(3\theta_j^\alpha))$
triquintic	$\prod_{j=1}^3 \frac{1}{128}(128 - 150\cos(\theta_j^\alpha) + 25\cos(3\theta_j^\alpha) - 3\cos(5\theta_j^\alpha))$
constant	$\prod_{j=1}^3 (1 + \exp(i\theta_j^\alpha))$
linear	$1 + \cos(\theta_1^\alpha) + \cos(\theta_2^\alpha) + \cos(\theta_2^\alpha) + \cos(\theta_1^\alpha - \theta_2^\alpha) \\ + \cos(\theta_1^\alpha - \theta_3^\alpha) + \cos(\theta_2^\alpha - \theta_3^\alpha) + \cos(\theta_1^\alpha + \theta_3^\alpha - \theta_2^\alpha)$

with Fourier symbol $\tilde{P}_{2h}^h(\boldsymbol{\theta}^\alpha) = \sum_{\boldsymbol{\kappa} \in J} p_{\boldsymbol{\kappa}} \exp(i\boldsymbol{\theta}^\alpha \boldsymbol{\kappa})$. Fourier symbols of implemented prolongation operators are shown in Table 6.5.

Remark 6.9 (Prolongation for d -dimensional applications). The scaling of Fourier symbols $\tilde{P}_{2h}^h(\boldsymbol{\theta}^\alpha)$ with $1/8$ is directly connected to the distinction between eight types of grid points for three-dimensional prolongation operators. Considering d -dimensional prolongation one has to deal with 2^d different types of grid points (compare with Remark 5.5) leading to a proper scaling by $1/2^d$; see (6.24) and (6.16) for the one- and two-dimensional cases, respectively. \blacktriangleright

Consequently, the Fourier representation of the two-grid operator reads

$$M_h^{2h}|_{\mathcal{F}_{2h}(\boldsymbol{\theta})} \stackrel{\Delta}{=} M^{2g}(\boldsymbol{\theta}) = (S^{2g}(\boldsymbol{\theta}))^{\nu_2} K^{2g}(\boldsymbol{\theta}) (S^{2g}(\boldsymbol{\theta}))^{\nu_1}, \quad (6.25)$$

where $K^{2g}(\boldsymbol{\theta})$ is composed of

$$\begin{aligned} I^{2g} &= \text{diag}\{1, 1, 1, 1, 1, 1, 1, 1\} \in \mathbb{C}^{8 \times 8}, \quad \mathcal{L}^{2g}(\boldsymbol{\theta}) = \tilde{L}_{2h}(2\boldsymbol{\theta}^{000}) \in \mathbb{C}^{1 \times 1}, \\ L^{2g}(\boldsymbol{\theta}) &= \text{diag}\{\ell^{000}, \ell^{111}, \ell^{100}, \ell^{011}, \ell^{010}, \ell^{101}, \ell^{001}, \ell^{110}\} \in \mathbb{C}^{8 \times 8}, \\ R^{2g}(\boldsymbol{\theta}) &= (r^{000} \ r^{111} \ r^{100} \ r^{011} \ r^{010} \ r^{101} \ r^{001} \ r^{110}) \in \mathbb{C}^{1 \times 8}, \\ P^{2g}(\boldsymbol{\theta}) &= (p^{000} \ p^{111} \ p^{100} \ p^{011} \ p^{010} \ p^{101} \ p^{001} \ p^{110})^T \in \mathbb{C}^{8 \times 1}. \end{aligned} \quad (6.26)$$

Here, we used the following abbreviations for the Fourier symbols of the different operators due to space limitations:

$$\ell^\alpha := \tilde{L}_h(\boldsymbol{\theta}^\alpha), \quad r^\alpha := \tilde{R}_h^{2h}(\boldsymbol{\theta}^\alpha), \quad p^\alpha := \tilde{P}_h^h(\boldsymbol{\theta}^\alpha).$$

Inserting $M^{2g}(\boldsymbol{\theta})$ from (6.25) into the definitions of $\rho(M_{2L})$, $\|M_{2L}\|_s$, and $\|M_{2L}\|_d$ from Section 6.2.7 yields the corresponding two-grid factors for three-dimensional applications. Again, certain frequencies $\boldsymbol{\theta} \in \Psi_{2g}$ are excluded from the analysis, with

$$\Psi_{2g} := \{\boldsymbol{\theta} \in \Theta_{2h} \mid \tilde{L}_{2h}(2\boldsymbol{\theta}^{000}) = 0 \text{ or } \tilde{L}_h(\boldsymbol{\theta}^\alpha) = 0\}.$$

6.3.2 Semicoarsening

Semicoarsening for three-dimensional applications means that the coarse grid is obtained by doubling the mesh size in one (x_1 -, x_2 -, x_3 -semicoarsening) or two ((x_1, x_2) -, (x_1, x_3) -, (x_2, x_3) -semicoarsening) space directions only; see Figure 3.7. The corresponding minimal invariant subspaces—the harmonics—for the coarse-grid correction operator K_h^H are two- or four-dimensional, respectively.

Example 6.9 (Spaces of harmonics for semicoarsening, $d = 3$). The spaces of harmonics for x_1 - and (x_1, x_2) -semicoarsening are given by

$$\begin{aligned} \mathcal{F}_{2h,h,h}(\boldsymbol{\theta}) &:= \text{span}\{ \varphi_h(\boldsymbol{\theta}, \cdot), \varphi_h(\boldsymbol{\theta} - (\text{sign}(\theta_1), 0, 0) \pi, \cdot) \} \\ \text{with } \boldsymbol{\theta} \in \Theta_{2h,h,h} &:= \{\boldsymbol{\theta} \in \Theta \mid -\pi/2 < \theta_1 \leq \pi/2\} \quad (= \Theta_{\text{low}}), \\ \mathcal{F}_{2h,2h,h}(\boldsymbol{\theta}) &:= \text{span}\{ \varphi_h(\boldsymbol{\theta}, \cdot), \varphi_h(\boldsymbol{\theta} - (\text{sign}(\theta_1), \text{sign}(\theta_2), 0) \pi, \cdot), \\ &\quad \varphi_h((\boldsymbol{\theta} - (\text{sign}(\theta_1), 0, 0) \pi, \cdot), \varphi_h(\boldsymbol{\theta} - (0, \text{sign}(\theta_2), 0) \pi, \cdot) \} \\ \text{with } \boldsymbol{\theta} \in \Theta_{2h,2h,h} &:= \{\boldsymbol{\theta} \in \Theta \mid -\pi/2 < \theta_1, \theta_2 \leq \pi/2\} \quad (= \Theta_{\text{low}}), \end{aligned}$$

respectively. ◀

However, to ensure compatible Fourier representations for coarse-grid correction and relaxation it is convenient to derive the Fourier representation of K_h^H with respect to the $2h$ -harmonics; compare with Section 6.2.8. We give examples for both types of semicoarsening, i.e., doubling the mesh size in one or two space directions.

Example 6.10 (x_1 -semicoarsening, $d = 3$). The spaces of harmonics for x_1 -semicoarsening ($\mathbf{H} = (2h, h, h)$) and the spaces of $2h$ -harmonics are connected by

$$\mathcal{F}_{2h}(\boldsymbol{\theta}) = \mathcal{F}_{2h,h,h}(\boldsymbol{\theta}^{000}) \cup \mathcal{F}_{2h,h,h}(\boldsymbol{\theta}^{011}) \cup \mathcal{F}_{2h,h,h}(\boldsymbol{\theta}^{010}) \cup \mathcal{F}_{2h,h,h}(\boldsymbol{\theta}^{001})$$

with $\boldsymbol{\theta} \in \Theta_{2h}$. Combining this relation with the invariance property of $K_h^{\mathbf{H}}$ with respect to the harmonics related to x_1 -semicoarsening yields

$$\begin{aligned} I_h, L_h &: \mathcal{F}_{2h}(\boldsymbol{\theta}) \longrightarrow \mathcal{F}_{2h}(\boldsymbol{\theta}), \\ R_h^{\mathbf{H}} &: \mathcal{F}_{2h}(\boldsymbol{\theta}) \longrightarrow \text{span}\{\varphi_{\mathbf{H}}((2\theta_1^\alpha, \theta_2^\alpha, \theta_3^\alpha), \cdot) \mid \alpha_1 = 0\}, \\ L_{\mathbf{H}} &: \text{span}\{\varphi_{\mathbf{H}}((2\theta_1^\alpha, \theta_2^\alpha, \theta_3^\alpha), \cdot) \mid \alpha_1 = 0\} \\ &\longrightarrow \text{span}\{\varphi_{\mathbf{H}}((2\theta_1^\alpha, \theta_2^\alpha, \theta_3^\alpha), \cdot) \mid \alpha_1 = 0\}, \\ P_{\mathbf{H}}^h &: \text{span}\{\{\varphi_{\mathbf{H}}((2\theta_1^\alpha, \theta_2^\alpha, \theta_3^\alpha), \cdot) \mid \alpha_1 = 0\} \longrightarrow \mathcal{F}_{2h}(\boldsymbol{\theta}). \end{aligned}$$

Hence the Fourier representation of $K_h^{\mathbf{H}}$ is governed by $I^{2g}, L^{2g}(\boldsymbol{\theta})$ from (6.26) and

$$\begin{aligned} \mathcal{L}^{2g}(\boldsymbol{\theta}) &= \text{diag}\{\tilde{L}_{\mathbf{H}}(2\theta_1^{000}, \theta_2^{000}, \theta_3^{000}), \tilde{L}_{\mathbf{H}}(2\theta_1^{011}, \theta_2^{011}, \theta_3^{011}), \\ &\quad \tilde{L}_{\mathbf{H}}(2\theta_1^{010}, \theta_2^{010}, \theta_3^{010}), \tilde{L}_{\mathbf{H}}(2\theta_1^{001}, \theta_2^{001}, \theta_3^{001})\} \in \mathbb{C}^{4 \times 4}, \\ R^{2g}(\boldsymbol{\theta}) &= \begin{pmatrix} r^{000} & 0 & r^{100} & 0 & 0 & 0 & 0 & 0 \\ 0 & r^{111} & 0 & r^{011} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & r^{010} & 0 & 0 & r^{110} \\ 0 & 0 & 0 & 0 & 0 & r^{101} & r^{001} & 0 \end{pmatrix} \in \mathbb{C}^{4 \times 8}, \\ P^{2g}(\boldsymbol{\theta}) &= \begin{pmatrix} p^{000} & 0 & 0 & 0 \\ 0 & p^{111} & 0 & 0 \\ p^{100} & 0 & 0 & 0 \\ 0 & p^{011} & 0 & 0 \\ 0 & 0 & p^{010} & 0 \\ 0 & 0 & 0 & p^{101} \\ 0 & 0 & 0 & p^{001} \\ 0 & 0 & p^{110} & 0 \end{pmatrix} \in \mathbb{C}^{8 \times 4}. \end{aligned}$$

For a well-defined two-grid operator, we exclude Fourier frequencies

$$\begin{aligned} \boldsymbol{\theta} \in \Psi_{2g} := \{ \boldsymbol{\theta} \in \Theta_{2h} \mid &\tilde{L}_{\mathbf{H}}((2\theta_1^{000}, \theta_2^{000}, \theta_3^{000})) = 0 \text{ or} \\ &\tilde{L}_{\mathbf{H}}((2\theta_1^{011}, \theta_2^{011}, \theta_3^{011})) = 0 \text{ or } \tilde{L}_{\mathbf{H}}((2\theta_1^{010}, \theta_2^{010}, \theta_3^{010})) = 0 \text{ or} \\ &\tilde{L}_{\mathbf{H}}((2\theta_1^{001}, \theta_2^{001}, \theta_3^{001})) = 0 \text{ or } \tilde{L}_h(\boldsymbol{\theta}^\alpha) = 0 \} \end{aligned}$$

from the two-grid analysis. ◀

Example 6.11 ((x_1, x_2) -semicoarsening, $d = 3$). Considering (x_1, x_2) -semicoarsening ($\mathbf{H} = (2h, 2h, h)$), we have

$$\mathcal{F}_{2h}(\boldsymbol{\theta}) = \mathcal{F}_{2h,2h,h}(\boldsymbol{\theta}^{000}) \cup \mathcal{F}_{2h,2h,h}(\boldsymbol{\theta}^{001})$$

with $\boldsymbol{\theta} \in \Theta_{2h}$ leading to

$$\begin{aligned} I_h, L_h &: \mathcal{F}_{2h}(\boldsymbol{\theta}) \longrightarrow \mathcal{F}_{2h}(\boldsymbol{\theta}), \\ R_h^H &: \mathcal{F}_{2h}(\boldsymbol{\theta}) \longrightarrow \text{span}\{\varphi_H((2\theta_1^\alpha, 2\theta_2^\alpha, \theta_3^\alpha), \cdot) \mid \alpha_1 = \alpha_2 = 0\}, \\ L_H &: \text{span}\{\varphi_H((2\theta_1^\alpha, 2\theta_2^\alpha, \theta_3^\alpha), \cdot) \mid \alpha_1 = \alpha_2 = 0\} \\ &\longrightarrow \text{span}\{\varphi_H((2\theta_1^\alpha, 2\theta_2^\alpha, \theta_3^\alpha), \cdot) \mid \alpha_1 = \alpha_2 = 0\}, \\ P_H^h &: \text{span}\{\varphi_H((2\theta_1^\alpha, 2\theta_2^\alpha, \theta_3^\alpha), \cdot) \mid \alpha_1 = \alpha_2 = 0\} \longrightarrow \mathcal{F}_{2h}(\boldsymbol{\theta}) \end{aligned}$$

similar to the previous example. The resulting Fourier representations for the transfer operators and the coarse-grid discretization read

$$\begin{aligned} \mathcal{L}^{2g}(\boldsymbol{\theta}) &= \text{diag}\{\tilde{L}_H(2\theta_1^{000}, 2\theta_2^{000}, \theta_3^{000}), \tilde{L}_H(2\theta_1^{001}, 2\theta_2^{001}, \theta_3^{001}), \} \in \mathbb{C}^{2 \times 2}, \\ R^{2g}(\boldsymbol{\theta}) &= \begin{pmatrix} r^{000} & 0 & r^{100} & 0 & r^{010} & 0 & 0 & r^{110} \\ 0 & r^{111} & 0 & r^{011} & 0 & r^{101} & r^{001} & 0 \end{pmatrix} \in \mathbb{C}^{2 \times 8}, \\ P^{2g}(\boldsymbol{\theta}) &= \begin{pmatrix} p^{000} & 0 \\ 0 & p^{111} \\ p^{100} & 0 \\ 0 & p^{011} \\ p^{010} & 0 \\ 0 & p^{101} \\ 0 & p^{001} \\ p^{110} & 0 \end{pmatrix} \in \mathbb{C}^{8 \times 2}. \end{aligned}$$

As in the previous example, we have to remove a certain subset of Fourier frequencies from the analysis: $\Psi_{2g} := \{\boldsymbol{\theta} \in \Theta_{2h} \mid \tilde{L}_H((2\theta_1^{000}, 2\theta_2^{000}, \theta_3^{000})) = 0 \text{ or } \tilde{L}_H((2\theta_1^{001}, 2\theta_2^{001}, \theta_3^{001})) = 0 \text{ or } \tilde{L}_h(\boldsymbol{\theta}^\alpha) = 0\}$. ◀

For the Fourier symbols $r^\alpha = \tilde{R}_h^{2h}(\boldsymbol{\theta}^\alpha)$, and $p^\alpha = \tilde{P}_h^{2h}(\boldsymbol{\theta}^\alpha)$ referring to the “one-” and “two-dimensional” transfer operators involved in semicoarsening for three-dimensional applications, we refer to [Tables 6.1, 6.2, and 6.3](#).

6.4 TWO-GRID ANALYSIS FOR SYSTEMS

Now we consider a discrete system of q partial differential equations

$$L_h \mathbf{u}_h(\mathbf{x}) = \begin{pmatrix} L_h^{1,1} & \dots & L_h^{1,q} \\ \vdots & \dots & \vdots \\ L_h^{q,1} & \dots & L_h^{q,q} \end{pmatrix} \begin{pmatrix} u_h^1(\mathbf{x}) \\ \vdots \\ u_h^q(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} f_h^1(\mathbf{x}) \\ \vdots \\ f_h^q(\mathbf{x}) \end{pmatrix} = \mathbf{f}_h(\mathbf{x}) \quad (\mathbf{x} \in G_h).$$

Fourier two-grid analysis for systems of equations is based on the vector-valued Fourier components $\varphi_h(\boldsymbol{\theta}, \cdot)$ introduced in Section 5.1.2 and the corresponding spaces of $2h$ -harmonics, $\mathcal{F}_{2h}(\boldsymbol{\theta})$ ($\boldsymbol{\theta} \in \Theta_{2h}$) which have already been applied in Section 5.5.

Similar to the scalar case, we exclude certain Fourier frequencies from the two-grid analysis to ensure a well-defined two-grid operator. Standard coarsening necessitates the exclusion of the following set:

$$\Psi_{2g} := \{\boldsymbol{\theta} \in \Theta_{2h} \mid \det(\tilde{\mathbf{L}}_h(\boldsymbol{\theta}^\alpha)) = 0 \text{ or } \det(\tilde{\mathbf{L}}_{2h}(2\boldsymbol{\theta}^{(0\dots 0)})) = 0\}.$$

An obvious generalization of the above considerations referring to scalar equations yields the invariance property of the two-grid operator w.r.t. the spaces of $2h$ -harmonics. This means that a two-grid operator \mathbf{M}_h^{2h} applied to a d -dimensional system is unitarily equivalent to a block matrix consisting of $(2^d q \times 2^d q)$ -blocks, i.e.,

$$\mathbf{M}_h^{2h}|_{\mathcal{F}_{2h}(\boldsymbol{\theta})} \stackrel{\wedge}{=} \mathbf{M}^{2g}(\boldsymbol{\theta}) = (\mathbf{S}^{2g}(\boldsymbol{\theta}))^{\nu_2} \mathbf{K}^{2g}(\boldsymbol{\theta}) (\mathbf{S}^{2g}(\boldsymbol{\theta}))^{\nu_1} \in \mathbb{C}^{2^d q \times 2^d q} \quad (\boldsymbol{\theta} \in \Theta_{2g})$$

with $\Theta_{2g} := \Theta_{2h} \setminus \Psi_{2g}$. Then, the asymptotic two-grid convergence factor can be easily calculated by

$$\rho(\mathbf{M}_{2L}) = \sup_{\boldsymbol{\theta} \in \Theta_{2g}} \rho(\mathbf{M}^{2g}(\boldsymbol{\theta}))$$

Similarly, the definitions of $\|\mathbf{M}_{2L}\|_s$ and $\|\mathbf{M}_{2L}\|_d$ carry over from the scalar case.

The Fourier representation for the relaxation method has been derived in Section 5.5. The Fourier representation of the coarse-grid correction for two-dimensional applications reads in detail as

$$\begin{aligned} \mathbf{K}^{2g}(\boldsymbol{\theta}) &= \mathbf{I}^{2g} - \mathbf{P}^{2g}(\boldsymbol{\theta})(\mathcal{L}^{2g}(\boldsymbol{\theta}))^{-1} \mathbf{R}^{2g}(\boldsymbol{\theta}) \mathbf{L}^{2g}(\boldsymbol{\theta}) \in \mathbb{C}^{4q \times 4q} \\ \text{with } \mathbf{I}^{2g} &= \text{diag}\{1, \dots, 1\} \in \mathbb{C}^{4q \times 4q}, \quad \mathcal{L}^{2g}(\boldsymbol{\theta}) = \tilde{\mathbf{L}}_{2h}(2\boldsymbol{\theta}) \in \mathbb{C}^{q \times q}, \\ \mathbf{L}^{2g}(\boldsymbol{\theta}) &= \text{bdiag}\{\tilde{\mathbf{L}}_h(\boldsymbol{\theta}^{00}), \tilde{\mathbf{L}}_h(\boldsymbol{\theta}^{11}), \tilde{\mathbf{L}}_h(\boldsymbol{\theta}^{10}), \tilde{\mathbf{L}}_h(\boldsymbol{\theta}^{01})\} \in \mathbb{C}^{4q \times 4q}, \\ \mathbf{R}^{2g}(\boldsymbol{\theta}) &= \left(\tilde{\mathbf{R}}_h^{2h}(\boldsymbol{\theta}^{00}) \tilde{\mathbf{R}}_h^{2h}(\boldsymbol{\theta}^{11}) \tilde{\mathbf{R}}_h^{2h}(\boldsymbol{\theta}^{10}) \tilde{\mathbf{R}}_h^{2h}(\boldsymbol{\theta}^{01}) \right) \in \mathbb{C}^{q \times 4q}, \\ \mathbf{P}^{2g}(\boldsymbol{\theta}) &= \frac{1}{4} \left(\tilde{\mathbf{P}}_{2h}^h(\boldsymbol{\theta}^{00}) \tilde{\mathbf{P}}_{2h}^h(\boldsymbol{\theta}^{11}) \tilde{\mathbf{P}}_{2h}^h(\boldsymbol{\theta}^{10}) \tilde{\mathbf{P}}_{2h}^h(\boldsymbol{\theta}^{01}) \right)^T \in \mathbb{C}^{4q \times q}. \end{aligned} \tag{6.27}$$

The Fourier symbols of the transfer operators in connection with systems of equations are given by $(q \times q)$ -matrices, similar as for the discretization operators \mathbf{L}_h and \mathbf{L}_{2h} (compare Section 5.1.2). They are composed of “scalar” Fourier symbols

$$\begin{aligned} \tilde{\mathbf{R}}_h^{2h}(\boldsymbol{\theta}^\alpha) &= \text{diag}\{(\tilde{R}_h^{2h}(\boldsymbol{\theta}^\alpha))^1, \dots, (\tilde{R}_h^{2h}(\boldsymbol{\theta}^\alpha))^q\} \in \mathbb{C}^{q \times q}, \\ \tilde{\mathbf{P}}_{2h}^h(\boldsymbol{\theta}^\alpha) &= \text{diag}\{(\tilde{P}_{2h}^h(\boldsymbol{\theta}^\alpha))^1, \dots, (\tilde{P}_{2h}^h(\boldsymbol{\theta}^\alpha))^q\} \in \mathbb{C}^{q \times q} \end{aligned}$$

where $(\tilde{R}_h^{2h}(\boldsymbol{\theta}^\alpha))^j$ and $(\tilde{P}_{2h}^h(\boldsymbol{\theta}^\alpha))^j$ ($j = 1, \dots, q$) refer to the transfer of defects and corrections for the j -th equation, respectively.

Example 6.12 (Transfer for two equations). Considering a two-dimensional system of two equations, one obtains the following Fourier representations for full weighting and bilinear interpolation:

$$\mathbf{R}^{2g}(\boldsymbol{\theta}) = \begin{pmatrix} r^{00} & 0 & r^{11} & 0 & r^{10} & 0 & r^{01} & 0 \\ 0 & r^{00} & 0 & r^{11} & 0 & r^{10} & 0 & r^{01} \end{pmatrix} \in \mathbb{C}^{2 \times 8},$$

$$\mathbf{P}^{2g}(\boldsymbol{\theta}) = \frac{1}{4} \begin{pmatrix} p^{00} & 0 \\ 0 & p^{00} \\ p^{11} & 0 \\ 0 & p^{11} \\ p^{10} & 0 \\ 0 & p^{10} \\ p^{01} & 0 \\ 0 & p^{01} \end{pmatrix} \in \mathbb{C}^{8 \times 2} \quad (\boldsymbol{\theta} \in \Theta_{2g})$$

with $r^\alpha = \tilde{R}_h^{2h}(\boldsymbol{\theta}^\alpha) = \frac{1}{4}(1 + \cos(\theta_1))(1 + \cos(\theta_2))$ and $p^\alpha = \tilde{P}_{2h}^h(\boldsymbol{\theta}^\alpha) = (1 + \cos(\theta_1))(1 + \cos(\theta_2))$. ◀

Remark 6.10 (Fourier symbols for transfer operators). If the transfer of defects and corrections is performed by the same transfer operators for each equation we have

$$(\tilde{R}_h^{2h}(\boldsymbol{\theta}^\alpha))^1 = \dots = (\tilde{R}_h^{2h}(\boldsymbol{\theta}^\alpha))^q \quad \text{and} \quad (\tilde{P}_{2h}^h(\boldsymbol{\theta}^\alpha))^1 = \dots = (\tilde{P}_{2h}^h(\boldsymbol{\theta}^\alpha))^q$$

as in the above example. However, in general one might apply different transfer operators for different equations. Famous examples are operator-dependent transfers (see Section 3.4.6) and staggered systems discussed in [10, 46, 70]. ▶

The necessary adjustments for three-dimensional systems or semicoarsening are obvious. This is demonstrated by the following examples.

Example 6.13 (Full weighting for three-dimensional systems). The Fourier representation of full weighting for three-dimensional systems of two equations applying standard coarsening is given by

$$\mathbf{R}^{2g}(\boldsymbol{\theta}) = \begin{pmatrix} r^{000} & 0 & r^{111} & 0 & r^{100} & 0 & r^{011} & 0 & r^{010} & 0 & r^{101} & 0 & r^{001} & 0 & r^{110} & 0 \\ 0 & r^{000} & 0 & r^{111} & 0 & r^{100} & 0 & r^{011} & 0 & r^{010} & 0 & r^{101} & 0 & r^{001} & 0 & r^{110} \end{pmatrix}$$

with $r^\alpha = \tilde{R}_h^{2h}(\boldsymbol{\theta}^\alpha) = \prod_{j=1}^3 \frac{1}{2}(1 + \cos(\theta_j))$. ◀

Example 6.14 (x_1 -semicoarsening for systems). Consider x_1 -semicoarsening for a two-dimensional system of two equations. Then, the Fourier representation of the coarse-grid correction with respect to the $2h$ -harmonics

is governed by \mathbf{I}^{2g} , \mathbf{L}^{2g} from (6.27) and

$$\mathbf{P}^{2g}(\boldsymbol{\theta}) = \frac{1}{2} \begin{pmatrix} p^{00} & 0 & 0 & 0 \\ 0 & p^{00} & 0 & 0 \\ 0 & 0 & p^{11} & 0 \\ 0 & 0 & 0 & p^{11} \\ p^{10} & 0 & 0 & 0 \\ 0 & p^{10} & 0 & 0 \\ 0 & 0 & p^{01} & 0 \\ 0 & 0 & 0 & p^{01} \end{pmatrix} \in \mathbb{C}^{8 \times 4}$$

$$\mathcal{L}^{2g}(\boldsymbol{\theta}) = \text{bdiag}\{\tilde{\mathbf{L}}_{\mathbf{H}}(2\boldsymbol{\theta}_1^{00}, \boldsymbol{\theta}_2^{00}), \tilde{\mathbf{L}}_{\mathbf{H}}(2\boldsymbol{\theta}_1^{01}, \boldsymbol{\theta}_2^{01})\} \in \mathbb{C}^{4 \times 4}$$

$$\mathbf{R}^{2g}(\boldsymbol{\theta}) = \begin{pmatrix} r^{00} & 0 & 0 & 0 & r^{10} & 0 & 0 & 0 \\ 0 & r^{00} & 0 & 0 & 0 & r^{10} & 0 & 0 \\ 0 & 0 & r^{11} & 0 & 0 & 0 & r^{01} & 0 \\ 0 & 0 & 0 & r^{11} & 0 & 0 & 0 & r^{01} \end{pmatrix} \in \mathbb{C}^{4 \times 8}$$

with $p^\alpha = \tilde{P}_{\mathbf{H}}^h(\boldsymbol{\theta}^\alpha)$ and $r^\alpha = \tilde{R}_{\mathbf{h}}^H(\boldsymbol{\theta}^\alpha)$; see (6.24) for the scalar case. The set of frequencies that has to be excluded from the analysis reads

$$\Psi_{2g} := \{ \boldsymbol{\theta} \in \Theta_{2h} \mid \det(\tilde{\mathbf{L}}_{\mathbf{H}}(2\boldsymbol{\theta}_1^{00}, \boldsymbol{\theta}_2^{00})) = 0 \text{ or } \det(\tilde{\mathbf{L}}_{\mathbf{H}}(2\boldsymbol{\theta}_1^{01}, \boldsymbol{\theta}_2^{01})) = 0 \\ \text{or } \det(\tilde{\mathbf{L}}_h(\boldsymbol{\theta}^\alpha)) = 0 \}.$$

◀

6.5 THREE-GRID ANALYSIS

In this last section of the chapter we give a comprehensive description of the Fourier three-grid analysis for two-dimensional scalar applications involving standard coarsening, to keep the presentation as simple as possible. However, the generalization to three-dimensional problems and systems of equations is discussed in Section 6.5.4.

The main idea of the three-grid analysis is to recursively apply the previously discussed two-grid analysis. This is obviously due to the recursive structure of a general three- or multigrid method. According to (3.7), the error transformation by a three-grid cycle is given by $e_h^{(i)} = M_h^{4h} e_h^{(i-1)}$ with

$$\begin{aligned} M_h^{4h} &= S_h^{v_2}(\omega) K_h^{4h} S_h^{v_1}(\omega) \\ &= S_h^{v_2}(\omega) (I_h - P_{2h}^h (I_{2h} - (M_{2h}^{4h})^\gamma) (L_{2h})^{-1} R_{2h}^{2h} L_h) S_h^{v_1}(\omega), \end{aligned} \quad (6.28)$$

where M_{2h}^{4h} , defined by (3.6) or (6.1) reads as

$$M_{2h}^{4h} = S_{2h}^{v_2}(\omega) (I_{2h} - P_{4h}^{2h} (L_{4h})^{-1} R_{4h}^{2h}) S_{2h}^{v_1}(\omega). \quad (6.29)$$

Of course, it is possible to vary the number of pre- and postsmoothing steps on the different grids leading to $\nu_1(h)$, $\nu_1(2h)$ and $\nu_2(h)$, $\nu_2(2h)$.

Instead of inverting L_{2h} , as it is done in the two-grid cycle (6.1), the $2h$ -equation is solved approximately by performing γ two-grid iterations M_{2h}^{4h} with zero initial approximation. This is reflected by replacing $(L_{2h})^{-1}$ from (6.1) by the expression

$$(\mathcal{L}_{2h}^{4h})^{-1} = (I_{2h} - (M_{2h}^{4h})^\gamma)(L_{2h})^{-1} \quad (6.30)$$

in (6.28). To see this, we consider an arbitrary nonsingular linear system (in matrix notation)

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

which is approximately solved by γ steps of a classical iterative method,

$$\mathbf{C}\mathbf{u}^{(i)} = (\mathbf{C} - \mathbf{L})\mathbf{u}^{(i-1)} + \mathbf{f},$$

based on the splitting $\mathbf{L} = \mathbf{C} + (\mathbf{L} - \mathbf{C})$; compare with Section 1.2. If a multigrid method is applied, we obtain

$$\mathbf{u}^{(i)} = \mathbf{M}\mathbf{u}^{(i-1)} + \mathbf{C}^{-1}\mathbf{f} \quad \text{with} \quad \mathbf{C}^{-1} = (\mathbf{I} - \mathbf{M})\mathbf{L}^{-1}.$$

\mathbf{M} denotes the error transformation matrix by one multigrid cycle. Starting with $\mathbf{u}^{(0)} = 0$, the γ -th iterate can easily be written as

$$\mathbf{u}^{(\gamma)} = (\mathbf{I} - \mathbf{M}^\gamma)\mathbf{L}^{-1}\mathbf{f}.$$

In a numerical algorithm, however, \mathbf{L}^{-1} (and in particular $(L_{2h})^{-1}$) is, of course, not applied explicitly.

6.5.1 Spaces of $4h$ -harmonics

Considering three-grid cycles, it is appropriate to divide the space of infinite grid functions into the following 16-dimensional subspaces.

Definition 6.3 ($4h$ -harmonics): The $4h$ -harmonics are defined by

$$\mathcal{F}_{4h}(\boldsymbol{\theta}) := \text{span}\{\varphi_h(\boldsymbol{\theta}_\beta^\alpha, \cdot) \mid \boldsymbol{\alpha} = (\alpha_1 \alpha_2), \boldsymbol{\beta} = (\beta_1 \beta_2);$$

$$\alpha_1, \alpha_2 \in \{0, 1\}; \beta_1, \beta_2 \in \{0, \frac{1}{2}\} \}$$

$$\text{with } \boldsymbol{\theta} = \boldsymbol{\theta}_{00}^{00} \in \Theta_{4h} := (-\pi/4, \pi/4]^2 \quad \text{and}$$

$$\boldsymbol{\theta}_\beta^{00} = \boldsymbol{\theta}_{00}^{00} - (\beta_1 \text{sign}(\theta_1), \beta_2 \text{sign}(\theta_2)) \pi,$$

$$\boldsymbol{\theta}_\beta^\alpha = \boldsymbol{\theta}_\beta^{00} - (\alpha_1 \text{sign}((\theta_1)_\beta), \alpha_2 \text{sign}((\theta_2)_\beta)) \pi.$$

▲

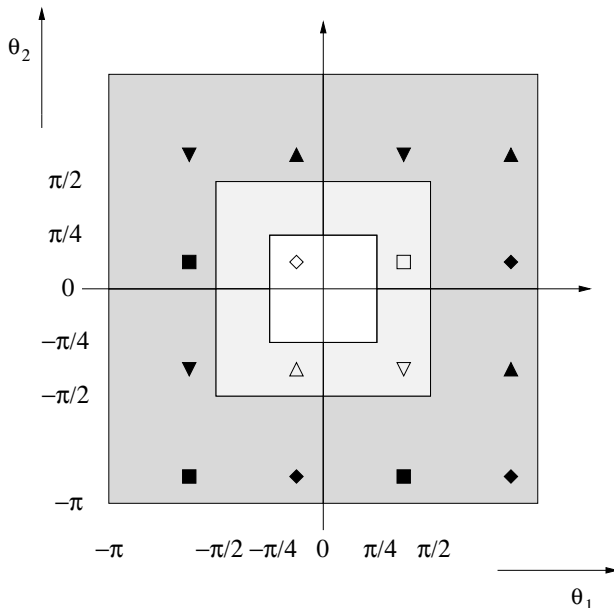


FIGURE 6.6: A set of Fourier frequencies that are coupled by a three-grid iteration generating a space of $4h$ -harmonics; see Definition 6.3.

Figure 6.6 illustrates this somewhat technical definition by indicating the location of the 16 different frequencies θ_{β}^{α} . It can be motivated in the same way as it has been done for the two-grid analysis concerning the $2h$ -harmonics. In the transition from G_{2h} to G_{4h} , a low frequency $\theta_{00}^{00} \in \Theta_{4h}$ is coupled with three high frequencies θ_{β}^{00} with $\beta \neq (00)$

$$\varphi_h(\theta_{00}^{00}, \mathbf{x}) = \varphi_h(\theta_{\frac{1}{2}\frac{1}{2}}^{00}, \mathbf{x}) = \varphi_h(\theta_{\frac{1}{2}0}^{00}, \mathbf{x}) = \varphi_h(\theta_{0\frac{1}{2}}^{00}, \mathbf{x}) \quad \text{with } \mathbf{x} \in G_{4h}. \quad (6.31)$$

Thinking in terms of Fourier components restricted to the coarser grids G_{2h} and G_{4h} this gives

$$\varphi_{2h}(2\theta_{00}^{00}, \mathbf{x}) = \varphi_{4h}(4\theta_{00}^{00}, \mathbf{x}) \quad (= \varphi_{4h}(4\theta_{\beta}^{\alpha}, \mathbf{x})) \quad \text{with } \theta_{00}^{00} \in \Theta_{4h}, \mathbf{x} \in G_{4h}.$$

We collect the Fourier components from (6.31) in the spaces of $(2h, 4h)$ -harmonics.

Definition 6.4 (Spaces of $(2h, 4h)$ -harmonics). The spaces of $(2h, 4h)$ -harmonics are defined by

$$\mathcal{F}_{2h}^{4h}(\theta) := \text{span}\{\varphi_h(\theta_{\beta}^{00}, \cdot) \mid \beta_1, \beta_2 \in \{0, \frac{1}{2}\}\} \quad \text{for } \theta_{00}^{00} = \theta \in \Theta_{4h}.$$

▲

Furthermore, each θ_{β}^{00} is coupled with three high-frequency components θ_{β}^{α} with $\alpha \neq (00)$ in the transition from G_h to G_{2h} , see Figure 6.6. It follows that a three-grid cycle couples 16 frequencies, indicating that the 15 high-frequency components alias on G_{4h} with the low-frequency component $\varphi_h(\theta_{00}^{00}, \cdot) = \varphi_{4h}(4\theta_{00}^{00}, \cdot)$.

Comparing Definition 6.1 and Definition 6.3, it immediately follows that each space of $4h$ -harmonics consists of four spaces of $2h$ -harmonics. For an arbitrary $\theta \in \Theta_{4h}$ we have

$$\mathcal{F}_{4h}(\theta) = \mathcal{F}_{2h}(\theta_{00}^{00}) \cup \mathcal{F}_{2h}(\theta_{\frac{1}{2}\frac{1}{2}}^{00}) \cup \mathcal{F}_{2h}(\theta_{\frac{1}{2}0}^{00}) \cup \mathcal{F}_{2h}(\theta_{0\frac{1}{2}}^{00}). \quad (6.32)$$

Remark 6.11 (Meaning of $2h$ -, $4h$ -, $(2h, 4h)$ -harmonics). The name $(2h, 4h)$ -harmonics refers to the fact that those Fourier components are gathered in the spaces of $(2h, 4h)$ -harmonics which coincide on G_{4h} in the transition from G_{2h} to G_{4h} . Similarly, the labeling $2h$ -harmonics and $4h$ -harmonics reminds of the aliasing of Fourier components on G_h in the transition from G_h to G_{2h} and the aliasing of Fourier components on G_{4h} in the transition from G_h to G_{4h} (via G_{2h}), respectively. ►

6.5.2 Invariance property of the three-grid operator

Analog to the two-grid analysis, we exclude certain frequencies to obtain a well-defined three-grid operator leading to the following slightly smaller space:

$$\mathcal{F}_{3g} := \mathcal{F} \setminus \bigcup_{\theta \in \Psi_{3g}} \mathcal{F}_{4h}(\theta) \quad \text{and} \quad \Theta_{3g} := \Theta_{4h} \setminus \Psi_{3g} \quad \text{with}$$

$$\Psi_{3g} := \left\{ \theta \in \Theta_{4h} \mid \tilde{L}_{4h}(4\theta_{00}^{00}) = 0 \text{ or } \tilde{L}_{2h}(2\theta_{\beta}^{00}) = 0 \text{ or } \tilde{L}_h(\theta_{\beta}^{\alpha}) = 0 \right\}.$$

Using Definition 6.4, (6.32), and the considerations from Section 6.2.6 concerning the invariance property of the two-grid operator, it follows for every $\theta \in \Theta_{3g} := \Theta_{4h} \setminus \Psi_{3g}$:

$$\begin{aligned} S_h(\omega), L_h, I_h : \mathcal{F}_{4h}(\theta) &\longrightarrow \mathcal{F}_{4h}(\theta), & \mathcal{L}_{2h}^{4h} : \mathcal{F}_{2h}^{4h}(\theta) &\longrightarrow \mathcal{F}_{2h}^{4h}(\theta), \\ R_h^{2h} : \mathcal{F}_{4h}(\theta) &\longrightarrow \mathcal{F}_{2h}^{4h}(\theta), & P_{2h}^h : \mathcal{F}_{2h}^{4h}(\theta) &\longrightarrow \mathcal{F}_{4h}(\theta). \end{aligned} \quad (6.33)$$

The relation for \mathcal{L}_{2h}^{4h} reads in more detail (see (6.30) and (6.29)):

$$\begin{aligned} S_{2h}(\omega), L_{2h}, I_{2h} : \mathcal{F}_{2h}^{4h}(\theta) &\longrightarrow \mathcal{F}_{2h}^{4h}(\theta), \\ L_{4h} : \text{span}\{\varphi_{4h}(4\theta_{00}^{00}, \cdot)\} &\longrightarrow \text{span}\{\varphi_{4h}(4\theta_{00}^{00}, \cdot)\} \\ R_{2h}^{4h} : \mathcal{F}_{2h}^{4h}(\theta) &\longrightarrow \text{span}\{\varphi_{4h}(4\theta_{00}^{00}, \cdot)\}, \\ P_{4h}^{2h} : \text{span}\{\varphi_{4h}(4\theta_{00}^{00}, \cdot)\} &\longrightarrow \mathcal{F}_{2h}^{4h}(\theta). \end{aligned}$$

Thus, the three-grid operator leaves the spaces of $4h$ -harmonics invariant,

$$M_h^{4h} : \mathcal{F}_{2h}^{4h}(\theta) \longrightarrow \mathcal{F}_{2h}^{4h}(\theta),$$

yielding a block diagonal representation of M_h^{4h} with the following (16×16) -block matrices:

$$M_h^{4h}|_{\mathcal{F}_{4h}(\boldsymbol{\theta})} \stackrel{\wedge}{=} M^{3g}(\boldsymbol{\theta}) \in \mathbb{C}^{16 \times 16} \quad \text{with} \quad (6.34)$$

$$M^{3g}(\boldsymbol{\theta}) = (S^{3g}(\boldsymbol{\theta}))^{\nu_2} (I^{3g} - P^{3g}(\boldsymbol{\theta})(\mathcal{L}^{3g}(\boldsymbol{\theta}))^{-1} R^{3g}(\boldsymbol{\theta}) L^{3g}(\boldsymbol{\theta})) (S^{3g}(\boldsymbol{\theta}))^{\nu_1}.$$

The different Fourier representations of the operators involved in the block-matrices $M^{3g}(\boldsymbol{\theta})$ can be expressed by the two-grid representations from Section 6.2:

$$\begin{aligned} I^{3g} &= \text{bdiag}\{I^{2g}, I^{2g}, I^{2g}, I^{2g}\} \in \mathbb{C}^{16 \times 16}, \\ S^{3g}(\boldsymbol{\theta}) &= \text{bdiag}\{S^{2g}(\boldsymbol{\theta}_{00}), S^{2g}(\boldsymbol{\theta}_{\frac{1}{2}\frac{1}{2}}), S^{2g}(\boldsymbol{\theta}_{\frac{1}{2}0}), S^{2g}(\boldsymbol{\theta}_{0\frac{1}{2}})\} \in \mathbb{C}^{16 \times 16}, \\ L^{3g}(\boldsymbol{\theta}) &= \text{bdiag}\{L^{2g}(\boldsymbol{\theta}_{00}), L^{2g}(\boldsymbol{\theta}_{\frac{1}{2}\frac{1}{2}}), L^{2g}(\boldsymbol{\theta}_{\frac{1}{2}0}), L^{2g}(\boldsymbol{\theta}_{0\frac{1}{2}})\} \in \mathbb{C}^{16 \times 16}, \\ R^{3g}(\boldsymbol{\theta}) &= \text{bdiag}\{R^{2g}(\boldsymbol{\theta}_{00}), R^{2g}(\boldsymbol{\theta}_{\frac{1}{2}\frac{1}{2}}), R^{2g}(\boldsymbol{\theta}_{\frac{1}{2}0}), R^{2g}(\boldsymbol{\theta}_{0\frac{1}{2}})\} \in \mathbb{C}^{4 \times 16}, \\ P^{3g}(\boldsymbol{\theta}) &= \text{bdiag}\{P^{2g}(\boldsymbol{\theta}_{00}), P^{2g}(\boldsymbol{\theta}_{\frac{1}{2}\frac{1}{2}}), P^{2g}(\boldsymbol{\theta}_{\frac{1}{2}0}), P^{2g}(\boldsymbol{\theta}_{0\frac{1}{2}})\} \in \mathbb{C}^{16 \times 4}, \\ (\mathcal{L}^{3g}(\boldsymbol{\theta}))^{-1} &= (I^{2g} - (M^{2g}(2\boldsymbol{\theta}, 2h))^\gamma) \\ &\quad (\text{diag}\{\mathcal{L}^{2g}(\boldsymbol{\theta}_{00}), \mathcal{L}^{2g}(\boldsymbol{\theta}_{\frac{1}{2}\frac{1}{2}}), \mathcal{L}^{2g}(\boldsymbol{\theta}_{\frac{1}{2}0}), \mathcal{L}^{2g}(\boldsymbol{\theta}_{0\frac{1}{2}})\})^{-1} \in \mathbb{C}^{4 \times 4}. \end{aligned} \quad (6.35)$$

Of course, $M^{2g}(2\boldsymbol{\theta}, 2h)$ can be calculated using $S^{2g}(\boldsymbol{\theta})$ and (6.21) if we replace h by $2h$ and $\boldsymbol{\theta}$ by $2\boldsymbol{\theta}$. Again an ordering for the second multiindex $\boldsymbol{\beta}$ has to be prescribed in order to obtain a unique representation, similar as for $\boldsymbol{\alpha}$. Here, (00) , $(\frac{1}{2}\frac{1}{2})$, $(\frac{1}{2}0)$, $(0\frac{1}{2})$ was applied.

6.5.3 Definition of three-grid convergence factor

Analogous to the two-grid factors from Section 6.2.7, we obtain the asymptotic three-grid convergence factor as the supremum of the spectral radii of certain block-matrices:

$$\rho(M_{3L}) := \sup_{\boldsymbol{\theta} \in \Theta_{3g}} \rho(M^{3g}(\boldsymbol{\theta})). \quad (6.36)$$

The spectral norm of the error and the defect reduction three-grid operator are computed by

$$\begin{aligned} \|M_{3L}\|_s &:= \sup_{\boldsymbol{\theta} \in \Theta_{3g}} \sqrt{\rho(M^{3g}(\boldsymbol{\theta})(M^{3g}(\boldsymbol{\theta}))^*)}, \\ \|M_{3L}\|_d &:= \sup_{\boldsymbol{\theta} \in \Theta_{3g}} \sqrt{\rho(L^{3g}(\boldsymbol{\theta})M^{3g}(\boldsymbol{\theta})(L^{3g}(\boldsymbol{\theta}))^{-1}(L^{3g}(\boldsymbol{\theta})M^{3g}(\boldsymbol{\theta})(L^{3g}(\boldsymbol{\theta}))^{-1})^*)}. \end{aligned}$$

The numerical calculation of three-grid factors is carried out with the help of the discrete set $\Theta_{3g}^P := \Theta_{3g} \cap G_{h_\theta}$ (compare Remarks 5.2 and 5.4).

The considerations from Remark 6.4 concerning the boundedness of $\rho(M_{2L})$ carry over to asymptotic three-grid factors.

Remark 6.12 (Variable Coefficients). A direct application of the k -grid analysis is not possible if we deal with operators $L_h(\mathbf{x})$ that are characterized by variable coefficients. However, Fourier k -grid analysis can be applied to the locally frozen operator at a fixed grid point $\boldsymbol{\xi}$, as it was done in the smoothing analysis; see Remark 5.8. In [8] it is motivated that the two-grid factor for $L_h(\mathbf{x})$ can be bounded by the supremum over the two-grid factors for the locally frozen operators. Thus, one may define the following k -grid convergence factors ($k = 2, 3$) in the case of variable coefficients:

$$\rho(M_{kL}(L_h(\mathbf{x}))) := \sup_{\boldsymbol{\xi} \in \Omega} \rho(M_{kL}(L_h(\boldsymbol{\xi}))). \quad (6.37)$$

Again, for an explicit calculation the supremum in the above definition has to be replaced by $\max_{\boldsymbol{\xi} \in \Omega_h}$. Some examples are discussed in 4.1.7 \blacktriangleright

6.5.4 Generalizations

The generalization of the three-grid analysis to three-dimensional applications or systems of equations is obvious assuming standard coarsening. One simply has to replace the two-grid Fourier representations (I^{2g} , $S^{2g}(\boldsymbol{\theta}_\beta)$, $L^{2g}(\boldsymbol{\theta}_\beta)$, $R^{2g}(\boldsymbol{\theta}_\beta)$, $P^{2g}(\boldsymbol{\theta}_\beta)$, $M^{2g}(2\boldsymbol{\theta}, 2h)$, $\mathcal{L}^{2g}(\boldsymbol{\theta}_\beta)$) occurring in (6.35) by their relevant counterparts for three dimensions or systems derived in Sections 6.3 and 6.4, respectively. Moreover, Ψ^{3g} has to be adapted properly.

Remark 6.13 (Generalization to d dimensions, k grids, and systems of equations). In the most general case, a d -dimensional problem has to be considered and k -grid cycles are applied to a system of q equations. Then, every low frequency $\boldsymbol{\theta} \in \Theta_{2^{1-k}g} := (-2^{1-k}\pi, 2^{1-k}\pi]^d$ is coupled with $2^{d(k-1)} - 1$ high frequencies. Accordingly, the related $2^{k-1}h$ -harmonics $\mathcal{F}_{2^{k-1}h}(\boldsymbol{\theta})$ are of dimension $2^{d(k-1)}$. Each operator of the k -grid cycle acts on the whole system and the dimensions of the corresponding block-matrices (see (6.21) and (6.35) for the 2-dimensional scalar case with two- and three-grid cycles, respectively) are given by

$$\begin{aligned} I^{kg}, S^{kg}(\boldsymbol{\theta}), L^{kg}(\boldsymbol{\theta}) &\in \mathbb{C}^{2^{d(k-1)}q \times 2^{d(k-1)}q}, & R^{kg}(\boldsymbol{\theta}) &\in \mathbb{C}^{2^{d(k-2)}q \times 2^{d(k-1)}q}, \\ P^{kg}(\boldsymbol{\theta}) &\in \mathbb{C}^{2^{d(k-1)}q \times 2^{d(k-2)}q}, & \mathcal{L}^{kg}(\boldsymbol{\theta}) &\in \mathbb{C}^{2^{d(k-2)}q \times 2^{d(k-2)}q}. \end{aligned}$$

The evaluation of k -grid cycles appears to be quite complicated and computationally expensive for many-level cycles, but one should take into account that the k -grid operators (3.7) are recursively defined and can be expressed in terms of two-grid operators. This means that the entries of a k -grid Fourier representation are given by two-grid Fourier symbols such as in (6.35).

In practice, however, it should be usually enough to perform a three-grid analysis to obtain sufficient insight into a multigrid method. \blacktriangleright

Chapter 7

FURTHER APPLICATIONS OF LOCAL FOURIER ANALYSIS

The last chapter of this monograph is dedicated to some further applications of local Fourier analysis which are not (yet) implemented in the accompanying software. In contrast to the previous chapters of Part II it is written in a more condensed style but we give several pointers to further reading.

If Fourier k -grid analysis predicts an unsatisfactory multigrid performance, it is sometimes not obvious which multigrid component has to be modified to overcome the particular difficulty at hand. For such situations it might be useful to analyze the different multigrid components *separately* in addition to the investigation of the complete k -grid operator. Two examples for this kind of analysis have already been given in [Chapter 5](#): The fine-grid discretization L_h might be analyzed by means of its measure of h -ellipticity whereas the relaxation method S_h can be optimized with the help of Fourier smoothing analysis. Similarly, there exist local Fourier analysis techniques for the remaining components in a multigrid cycle. The correct choice of transfer operators might be identified by an investigation of the corresponding “orders” specified in Section 7.1. The quality of the coarse-grid discretization is subject to the “simplified k -grid analysis” presented in Section 7.2. However, we would like to point out that even if each multigrid component has been selected with the help of the corresponding Fourier analysis technique (i.e., with the help of the measure of h -ellipticity, the smoothing factor, the orders of transfer operators, and the simplified k -grid analysis), it may happen that the resulting multigrid method exhibits an unsatisfactory convergence behavior. This crucial observation is due to the fact that the quality of every multigrid algorithm strongly depends on the correct *interplay* of the different multigrid components. This interplay can only be investigated by a complete k -grid analysis. As an example for such a behavior, recall the application of **MG2** to the Poisson equation discussed in Section 4.1.1. The choice of each component involved in **MG2** is reasonable for the application at hand, but the interplay of the selected components (in particular the combination of injection and point RB-JAC relaxation) yields a diverging algorithm which is very well predicted by Fourier k -grid analysis. Regarding this matter, one should consider these *disconnected* analysis techniques for each of the different multigrid components only as supplementary analysis tools which cannot replace the analysis of the complete multigrid operator.

So far we presented the Fourier k -grid analysis only w.r.t. vertex-centered grids because the accompanying software is based on such a grid structure. However, the analysis can be easily adapted to other common choices like cell-centered or staggered grids. This is demonstrated in Section 7.3 for the cell-centered case. For the analysis on staggered grids, we refer to the literature [10, 46, 70].

There are basically two principles for the tuning of multigrid components: optimality and robustness. Optimality means that the components are tuned in order to obtain a highly efficient but highly specialized solver for *one particular* problem whereas robustness means that the components are tuned to obtain a multigrid method which works well for a *large class* of problems. Both principles have their advantages and disadvantages in the scientific area. An optimized solver is the method of choice if a specialized problem has to be solved many times. Robustness comes into play when different problems with a similar structure are to be solved, for example, in scientific software for solving PDEs. In Chapter 4, we applied local Fourier k -grid analysis in order to *optimize* different multigrid components for a large variety of particular problems and discretizations. In general, it is difficult to construct *robust* multigrid methods with *fixed* multigrid components for large classes of problems. A popular alternative to construct a robust solver is the use of multigrid as a preconditioner for a Krylov subspace acceleration method like GMRES. Hence, it is useful to apply local Fourier analysis to such a combined solution method yielding quantitative convergence estimates. This kind of analysis is briefly described in Section 7.4 dealing with restarted GMRES (GMRES(m)) [52] preconditioned by multigrid.

7.1 ORDERS OF TRANSFER OPERATORS

The order of prolongation and restriction is an important issue for the selection of transfer operators within a multigrid algorithm. We distinguish between polynomial order, high-, and low-frequency order specified in the following subsections.

7.1.1 Polynomial order

The polynomial orders m_p^{poly} and m_r^{poly} of prolongation and restriction operators, respectively, play a crucial role for the convergence analysis of multigrid methods. m_p^{poly} is defined as the highest order plus one of polynomials that are interpolated exactly by the prolongation under consideration. The polynomial order of a restriction operator is defined as that of the adjoint

prolongation. The well-known rule

$$m_p^{\text{poly}} + m_r^{\text{poly}} > M, \quad (7.1)$$

demands that the sum of the polynomial orders of prolongation m_p^{poly} and restriction m_r^{poly} is larger than the highest order M of differentiation of the original differential operator. Relation (7.1) is a necessary condition to prove mesh size independent convergence rates for multigrid following the analysis from [28]. However, at least for cell-centered multigrid discussed in Section 7.3 this condition can be weakened as shown in [3]. As an alternative one might consider low- and high-frequency orders yielding a similar (but slightly less restrictive) rule given in (7.2) below.

7.1.2 High- and low-frequency order

The definition of the orders of intergrid transfer operators in the context of the local Fourier analysis is taken from [7, 30]. It differs from the polynomial order defined above. R_h^{2h} has a low-frequency order m_r^{low} and a high-frequency order m_r^{high} , if for $|\boldsymbol{\theta}^{(0\dots 0)}| \rightarrow 0$ m_r^{low} and m_r^{high} are the largest numbers satisfying

$$\begin{aligned} \tilde{R}_h^{2h}(\boldsymbol{\theta}^{(0\dots 0)}) &= 1 + O\left(|\boldsymbol{\theta}^{(0\dots 0)}| m_r^{\text{low}}\right), \\ \tilde{R}_h^{2h}(\boldsymbol{\theta}^\alpha) &= O\left(|\boldsymbol{\theta}^{(0\dots 0)}| m_r^\alpha\right) \quad \text{with } \alpha \neq (0\dots 0) \quad \text{and} \\ m_r^{\text{high}} &= \min \{m_r^\alpha \mid \alpha = (\alpha_1, \dots, \alpha_d) \neq (0, \dots, 0), \alpha_j = 0, 1 \ (j = 1, \dots, d)\}. \end{aligned}$$

Analogously, we define the low-frequency order m_p^{low} and the high-frequency order m_p^{high} of a prolongation P_h^{2h} . Several examples for the different orders of transfer operators are shown in Table 7.1. They can easily be calculated by Taylor expansion using the corresponding Fourier symbols of P_h^{2h} and R_h^{2h} given in the previous chapter.

Remark 7.1 (Relation between m_p^{poly} and $m_p^{\text{low}}, m_p^{\text{high}}$). The polynomial and the high- and low-frequency orders are directly connected as shown in [30]: If a prolongation has polynomial order n , then both high- and low-frequency orders are at least n ; i.e., $m_p^{\text{poly}} \leq m_p^{\text{low}}, m_p^{\text{high}}$. ►

First of all, it is heuristically clear that $m_r^{\text{low}}, m_r^{\text{high}} > 0$ is a desirable property. According to the two-(multi-)grid idea of correcting smooth (or “low”) error components on the coarse grid, the restriction operator should have the property that $R_h^{2h}\varphi_h(\boldsymbol{\theta}^{00}, \cdot)$ is a “good” approximation to $\varphi_{2h}(2\boldsymbol{\theta}^{00}, \cdot)$ for $|\boldsymbol{\theta}| \rightarrow 0$, i.e., $m_r^{\text{low}} > 0$ must hold. Moreover, for $m_r^{\text{high}} > 0$, the restriction acts like a filter. Rough (or “high”) error components are damped and

TABLE 7.1: Orders of transfer operators, vertex-centered case

prolongation	m_p^{poly}	m_p^{low}	m_p^{high}
(bi-, tri-) linear	2	2	2
(bi-, tri-) cubic	4	4	4
(bi-, tri-) quintic	6	6	6
constant upwind	0	1	1
restriction	m_r^{poly}	m_r^{low}	m_r^{high}
full weighing	2	2	2
injection	0	∞	0
half weighting	0	2	0
seven point	1	2	2

they show up on the coarse grid as smooth components with very small amplitudes. This is, again, in accordance with the basic two-(multi-)grid idea that rough error components should be treated on the fine grid by the relaxation procedure and that their spurious contribution on the coarse grid (in terms of additionally induced smooth error components produced by the restriction operator) should be controlled. The heuristic explanation for why $m_p^{\text{low}}, m_p^{\text{high}} > 0$ is preferable is again obvious. The computed correction on the coarse grid for smooth error components should be transferred to the fine grid as accurately as possible ($m_p^{\text{low}} > 0$), whereas the spuriously introduced rough error components should be kept to a minimum ($m_p^{\text{high}} > 0$).

Apart from these heuristic considerations, the high-frequency orders should satisfy the following condition [7, 30]:

$$m_r^{\text{high}} + m_p^{\text{high}} \geq M. \quad (7.2)$$

It is shown in [30] that (7.2) is a necessary condition for the eigenvalues of $\tilde{K}_h^{2h}(\theta)$ (i.e., the eigenvalues of the Fourier representation of the coarse-grid correction operator K_h^{2h} , see (6.1)) to remain bounded as $|\theta| \rightarrow 0$ and thus a necessary condition for the two-(multi-)grid method to be convergent as the resolution of the finest grid is increased (unless for certain special situations where the relaxation operator takes care of such problematic eigenvalues).

Combining (7.2) and Remark 7.1, condition (7.1) seems to be too restrictive in the framework of local Fourier analysis. In fact, the experiments given in [44] indicate that for the convergence of cell-centered multigrid, (7.2) is more relevant and the violation of (7.1) does not lead to severe performance

degradation.

Remark 7.2 (Boundedness of error and defect reduction norm). To ensure that the norm values $\|K_h^{2h}\|_s$ and $\|K_h^{2h}\|_d$ are bounded, one has to require

$$m_r^{\text{high}} \geq M, \quad m_p^{\text{high}} \geq 0, \quad \text{and} \quad m_p^{\text{high}} \geq M, \quad m_r^{\text{high}} \geq 0,$$

respectively [30]. Obviously, $m_r^{\text{high}} \geq M$ is violated in case of an injection of residuals ($m_r^{\text{high}} = 0$) applied to the Poisson equation ($M = 2$). As a consequence, the corresponding error reduction norm of a two-grid cycle $\|M_{2L}\|_s$ is unbounded as reported in Example 6.6. Switching to the full weighting operator ($m_r^{\text{high}} = 2$) it is possible to fix this deterioration of the error norm; compare with Remark 6.4. ►

7.2 SIMPLIFIED FOURIER k-GRID ANALYSIS

In the previous chapter, we presented the Fourier analysis for the complete k -grid operator. The so-called simplified k -grid analysis [12, 56] solely focuses on the relation between the fine-grid discretization and the discretization on the coarser grids.

For singularly perturbed problems one often observes a coarse-grid correction difficulty which is caused by an unsatisfactory approximation of certain “characteristic” very low-frequency Fourier components. Then, it is reasonable to consider the so-called *simplified two-grid* or *simplified k -grid analysis*. The influence of the smoothing method is neglected or, more precisely, an “ideal” relaxation is assumed with good smoothing properties for the high-frequency components and which leaves the low-frequency components unchanged. In this respect, the simplified analysis can be seen as a counterpart to the smoothing analysis from Chapter 5 where we have an “ideal” coarse-grid correction operator (5.25). The analysis is, furthermore, substantially simplified if

- we neglect the coupling of the harmonics (see Definitions 5.8 and 6.3) and
- we assume that the transfer operators act like identities which is, in fact, nearly correct for very low-frequency components.

Moreover, we assume standard coarsening and restrict our considerations to two-dimensional applications for ease of presentation. Then, one only has to investigate the strongly simplified “two-grid” Fourier symbol

$$\tilde{C}_h^{2h}(\boldsymbol{\theta}) = 1 - \tilde{L}_h(\boldsymbol{\theta})/\tilde{L}_{2h}(2\boldsymbol{\theta}).$$

Note that $\tilde{C}_h^{2h}(\boldsymbol{\theta})$ does *not* belong to a well-defined operator due to the above simplifications.

For problems with characteristic directions, as they occur for hyperbolic or parabolic operators [25], we define

$$\chi_2 := \lim_{\theta_1 \rightarrow 0} \tilde{C}_h^{2h}(\boldsymbol{\theta}_c) = \lim_{\theta_1 \rightarrow 0} \left(1 - \frac{\tilde{L}_h(\boldsymbol{\theta}_c)}{\tilde{L}_{2h}(2\boldsymbol{\theta}_c)} \right) \quad \text{with} \quad \boldsymbol{\theta}_c = (\theta_1, c\theta_1),$$

where $\boldsymbol{\theta}_c = (\theta_1, c\theta_1)$ denotes a frequency along the characteristic direction $\theta_2 = c\theta_1$ corresponding to the operator under consideration. The generalization for systems of q PDEs reads

$$\chi_2 := \lim_{\theta_1 \rightarrow 0} \rho \left(\tilde{C}_h^{2h}(\boldsymbol{\theta}_c) \right) = \lim_{\theta_1 \rightarrow 0} \rho \left(I_q - \left(\tilde{L}_{2h}(2\boldsymbol{\theta}_c) \right)^{-1} \tilde{L}_h(\boldsymbol{\theta}_c) \right),$$

where $I_q \in \mathbb{C}^{q \times q}$ is the identity matrix. Obviously, the simplified analysis investigates the ability of the coarse-grid discretization to approximate the fine-grid discretization for very low-frequency characteristic Fourier components.

If the simplified *two*-grid analysis applies it is, furthermore, possible to give a rough prediction $\chi_k(\gamma)$ of the convergence of a k -grid cycle as it is indicated in [12, 56]. $\chi_k(\gamma)$ can be calculated using the following recurrence equation:

$$\chi_i(\gamma) := 1 - (1 - \chi_2) \left(1 - (\chi_{(i-1)}(\gamma))^\gamma \right) \quad \text{for} \quad i = 3, \dots, k \quad (7.3)$$

with cycle index γ . For example, the simplified k -grid convergence factor for a V-cycle ($\gamma = 1$) reduces to

$$\chi_k(1) = 1 - (1 - \chi_2)^{k-1}. \quad (7.4)$$

In this monograph, we only consider V-cycles and W-cycles ($\gamma = 1, 2$) as a larger cycle index often leads to less efficient multigrid methods. Note that there might be a fast deterioration of the k -grid factors $\chi_k(\gamma)$ for an increasing number of levels and $\gamma = 1, 2$; compare with Table 7.2.

Remark 7.3 (Reliability of $\chi_k(\gamma)$). The simplified Fourier analysis can be considered as a “worst-case” analysis. The limiting k -grid factors $\chi_k(\gamma)$ can sometimes be avoided by an appropriate choice of the smoothing procedure which is neglected by the simplified analysis. For example, an application of ILU smoothing for the rotated anisotropic diffusion equation (4.6) also takes care of problematic characteristic low-frequency error components leading to fast multigrid convergence. However, the definition of robust ILU smoothers for *three*-dimensional problems involving mixed derivatives in connection with standard coarsening is much more difficult than for two dimensions. ►

Typical examples for simplified two-grid factors (and thus for the related k -grid factors from (7.3)) arise from the anisotropic diffusion equation and convection-dominated flow problems.

Example 7.1 (Rotated anisotropic diffusion equation). If the rotated anisotropic diffusion equation (4.6) with $\beta = 45^\circ$ is discretized by central differences (on Ω_h and Ω_{2h}), one obtains for the singularly perturbed cases $\varepsilon \rightarrow 0, \infty$:

$$\theta_c = (\theta_1, \pm\theta_1) \quad \text{and} \quad \chi_2 = \frac{3}{4};$$

see [67]. From Table 7.2 one has to expect a very unsatisfactory multigrid convergence. Different possibilities to overcome this difficulty are discussed in Section 4.1.6. \blacktriangleleft

TABLE 7.2: Simplified k -grid convergence factors $\chi_k(\gamma)$ (7.3) for $\chi_{2g} = 0.5, 0.75$

k	2	3	4	5	6	7
V-cycle	0.500	0.750	0.875	0.938	0.969	0.984
W-cycle	0.500	0.625	0.695	0.742	0.775	0.800
V-cycle	0.750	0.938	0.984	0.996	0.999	1.000
W-cycle	0.750	0.891	0.948	0.975	0.988	0.994

Example 7.2 (Convection diffusion equation). In [12], χ_2 is calculated for the convection diffusion equation for a general discretization of p -th order accuracy assuming a vanishing diffusion coefficient $\varepsilon \rightarrow 0$. It is given by

$$\chi_2 = 1 - 2^{-p}. \quad (7.5)$$

Table 7.2 illustrates the deterioration of the multigrid convergence with an increasing number of levels for $p = 1, 2$, i.e. $\chi_2 = 1/2, 3/4$.

One obtains some additional insight if we allow a nonvanishing ε , especially for the higher-order upwind discretization from Section 4.1.8. Details can be found in [67]. We would like to point out that also for convection-dominated Navier-Stokes-type flows the simplified two-grid factor is governed by (7.5) as it has been demonstrated in [67]. \blacktriangleleft

7.3 CELL-CENTERED MULTIGRID

Thinking of multigrid methods, one typically imagines a hierarchy of grids, where each coarse grid is a coarsening of the previous finer one. In this vertex-centered case a node of the coarse grid is also a node of the fine grid. In the

finite element context one thinks of a hierarchy of nested spaces. However, this is not the only possible situation. Cell-centered discretization like, e.g., finite volume approaches naturally lead to grid hierarchies, where nodes on coarser grids do not form a subset of the fine-grid nodes. An example is given in Figure 7.1. Here, four fine-grid cells are united to form a coarse-grid cell. Since nodes/unknowns are located in the cell centers this leads to a nonnested situation. We restrict our considerations to standard coarsening shown in Figure 7.1.

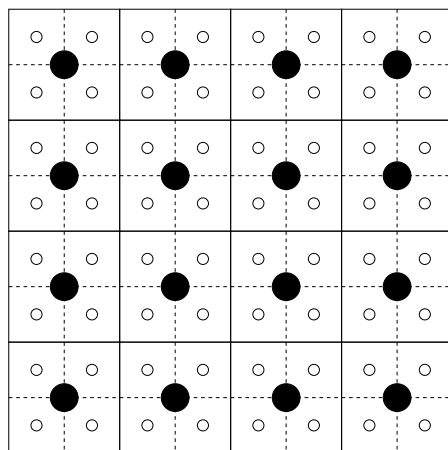


FIGURE 7.1: Coarsening in cell-centered multigrid: White circles mark the centers of fine-grid cells, black circles those of coarse-grid cells.

The development of cell-centered multigrid methods is strongly related to the question, what types of problems can be treated without the use of problem-dependent transfer operators. It is well known that interface problems, i.e., problems with jumping coefficients cannot be handled well by vertex-centered multigrid with standard prolongation operators. One has to modify the interpolation to reflect the discontinuity, see, e.g., [1, 17, 77]. In cell-centered multigrid one tries to avoid this, possibly at the expense of requiring more sophisticated smoothers [34, 35, 64].

It has been reported in literature [43, 44] that this different type of grid hierarchy does not only require different transfer operators, it also leads to quite a different behavior of the resulting multigrid methods. The interplay between relaxation and correction is different. This can require changes to the smoothers, like the introduction of a relaxation parameter. For complicated problems (especially in three dimensions) the application of more sophisticated smoothers may even be necessary, see, e.g., [34]. Results are

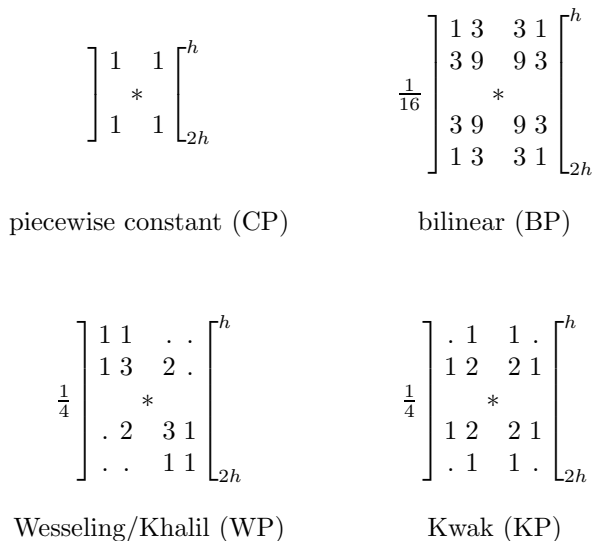


FIGURE 7.2: Four different types of interpolation operators. The * marks the position of the coarse-grid node.

also sometimes contradictory to the expectations based on experience from the vertex-centered case. In this context, the application of local Fourier two- and three-grid analysis is particularly useful. Especially the three-grid analysis can be of great benefit here for a deeper understanding of the interplay between smoothing and coarse-grid correction in the cell-centered case. Moreover, the different behavior of pre- and postsmoothing or V- and W- cycles can be investigated, which is particularly conspicuous in the cell-centered case. Several examples for such phenomena are discussed and explained with the help of Fourier three-grid analysis in [44]; compare also with Section 7.3.4. Here, we simply focus on the necessary modifications of the local Fourier analysis in the case of cell-centered grids which are mainly governed by the transfer operators.

7.3.1 Transfer operators

In literature one finds four main types of interpolation operators P_{2h}^h . These are given in Figure 7.2 in stencil notation. The notation shows the weight with which the value of the coarse-grid function at the coarse-grid node * contributes to the fine-grid function at its neighboring fine-grid nodes. Restriction operators R_h^{2h} are constructed as the adjoints of the given prolongations.

The simplest prolongation operator is the piecewise constant interpolation (CP). It has a small stencil, involving a small number of operations, but is only of first order. Here order refers to the polynomial order of the operator

defined in Section 7.1.1. A prolongation of second order is bilinear interpolation (BP). However, as can be seen from Figure 7.2 it requires a compact 16-point stencil. Prolongation operators that are also of second order, but employ sparser stencils, are the ones by Wesseling/Khalil (WP) and Kwak (KP). Note that the stencil of Wesseling/Khalil with its 10 nonzero entries is only slightly more costly than the compact 9-point-stencil of bilinear interpolation in the vertex-centered case.

The stencil by Wesseling and Khalil is based on linear interpolation on triangles, the vertices of the triangles being the nodes of the coarse grid, see, e.g., [65]. The stencil by Kwak is derived in [37] from geometric considerations.

All three second-order interpolations involve stencils that will extend outside of the problem domain, when applied to a cell in the direct vicinity of the boundary. This holds even in the simple case of Dirichlet boundary conditions. Thus the stencils must be modified along the boundary for an explicit implementation. These modifications in the case of Dirichlet and Neumann boundary conditions are detailed in [44, 65]. However, in connection with local Fourier analysis—assuming an infinite grid—it is sufficient to stay with the constant coefficient stencils from Figure 7.2.

In Figure 7.3 the respective stencils for three-dimensional applications are given, with the exception of piecewise constant prolongation (CP) which is trivial. The three-dimensional variant of WP is constructed by linear interpolation on tetrahedra, whose vertices are formed by the coarse-grid nodes, see again [65]. Kwak employs in [38] the same geometric ideas as for the two-dimensional case to derive his stencil. Note, however, that, while a fine-grid node is interpolated from its four nearest coarse-grid neighbors, this is not done in a linear fashion as in two dimensions, so the polynomial order is only 1. However, this need not necessarily imply a worse performance, as the results in [38] and the investigation of the high- and low-frequency orders indicate.

Remark 7.4 (Vertex-centered analogs of WP). WP in two dimensions can be considered as the cell-centered relative to seven-point prolongation in the vertex-centered case; see Section 3.4.6. The same statement holds for the three-dimensional variant of WP and the adjoint of (3.12). ►

7.3.2 Fourier two- and three-grid analysis

Fourier smoothing analysis is based on an ideal coarse-grid correction (5.25) and does not distinguish between cell-centered or vertex-centered coarsening, since the splitting into low- and high-frequency components remains unchanged. Hence, all considerations presented in Chapter 5 apply in the case of cell-centered grids, as well, including the corresponding parts of the accompanying software. Consequently, we only have to consider the modifications of the analysis if more levels are involved.

$$\begin{aligned}
 & \left[\frac{1}{64} \begin{pmatrix} 1 & 3 & 3 & 1 \\ 3 & 9 & 9 & 3 \\ & * & & \\ 3 & 9 & 9 & 3 \\ 1 & 3 & 3 & 1 \end{pmatrix} \begin{pmatrix} 3 & 9 & 9 & 3 \\ 9 & 27 & 27 & 9 \\ & * & & \\ 9 & 27 & 27 & 9 \\ 3 & 9 & 9 & 3 \end{pmatrix} \begin{pmatrix} 3 & 9 & 9 & 3 \\ 9 & 27 & 27 & 9 \\ & * & & \\ 9 & 27 & 27 & 9 \\ 3 & 9 & 9 & 3 \end{pmatrix} \begin{pmatrix} 1 & 3 & 3 & 1 \\ 3 & 9 & 9 & 3 \\ & * & & \\ 3 & 9 & 9 & 3 \\ 1 & 3 & 3 & 1 \end{pmatrix} \right]_{2h}^h \\
 & \text{tri-linear (TP)} \\
 & \left[\frac{1}{4} \begin{pmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ & * & & \\ \cdot & \cdot & 1 & 1 \\ \cdot & \cdot & 1 & 1 \end{pmatrix} \begin{pmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & 2 & 2 & \cdot \\ & * & & \\ \cdot & 2 & 3 & 1 \\ \cdot & \cdot & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & \cdot & \cdot \\ 1 & 3 & 2 & \cdot \\ & * & & \\ \cdot & 2 & 2 & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{pmatrix} \begin{pmatrix} 1 & 1 & \cdot & \cdot \\ 1 & 1 & \cdot & \cdot \\ & * & & \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{pmatrix} \right]_{2h}^h \\
 & \text{Wesseling/Khalil (WP)} \\
 & \left[\frac{1}{6} \begin{pmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & 1 & 1 & \cdot \\ & * & & \\ \cdot & 1 & 1 & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{pmatrix} \begin{pmatrix} \cdot & 1 & 1 & \cdot \\ 1 & 3 & 3 & 1 \\ & * & & \\ 1 & 3 & 3 & 1 \\ \cdot & 1 & 1 & \cdot \end{pmatrix} \begin{pmatrix} \cdot & 1 & 1 & \cdot \\ 1 & 3 & 3 & 1 \\ & * & & \\ 1 & 3 & 3 & 1 \\ \cdot & 1 & 1 & \cdot \end{pmatrix} \begin{pmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & 1 & 1 & \cdot \\ & * & & \\ \cdot & 1 & 1 & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{pmatrix} \right]_{2h}^h \\
 & \text{Kwak (KP)}
 \end{aligned}$$

FIGURE 7.3: Prolongation stencils in three dimensions (from left to right the planes of the three-dimensional stencil are shown from bottom to top).

Fourier two- and three-grid analysis for cell-centered and for vertex-centered multigrid differ only w.r.t. the treatment of the transfer operators. Thus, we concentrate on the calculation of the Fourier representations of R_h^{2h} , P_{2h}^h and their related orders which are crucial for an efficient coarse-grid correction. We consider a discrete elliptic boundary value problem

$$B_h u_h = f_h \quad \text{in } \Omega_h, \quad B_h u_h = g_h \quad \text{on } \partial\Omega_h,$$

resulting from a cell-centered discretization of its continuous analog. The grid is given as the intersection $\Omega_h = \Omega \cap G_h$ of the problem domain $\Omega \subset \mathbb{R}^2$ with a cell-centered infinite grid of mesh size h

$$G_h = \left\{ \mathbf{x} = \boldsymbol{\kappa}h + (h/2, \dots, h/2)^T \mid \boldsymbol{\kappa} \in \mathbb{Z}^d \right\}. \quad (7.6)$$

In contrast to the complete three-grid operator which acts on three different grids (G_h, G_{2h}, G_{4h}), the transfer operators involve only two grids, e.g., G_h and G_{2h} or G_{2h} and G_{4h} . Thus, it is sufficient to consider the transfer from

G_h to G_{2h} and reverse. The remaining transfers involved in a three-grid cycle can be handled analogously.

Consider an arbitrary restriction which is characterized by a stencil $R_h^{2h} \triangleq [r_{\kappa}]_h^{2h}$, i.e.,

$$(R_h^{2h} w_h)(\mathbf{x}) = \sum_{\kappa \in J} r_{\kappa} w_h(\mathbf{x} + \kappa h) \quad (\mathbf{x} \in G_{2h})$$

with a finite index set J and $\kappa = (\kappa_1, \dots, \kappa_d)$, $\kappa_j \in \{\pm 1/2, \pm 3/2, \pm 5/2, \dots\}$ ($j = 1, \dots, d$). Note that

$$\mathbf{x} + \kappa h \in G_h \quad \text{for } \mathbf{x} \in G_{2h}$$

due to the elements of J which reflect the cell-centered grid, compare with (7.6) and Figure 7.1. For the Fourier components this means that

$$\begin{aligned} (R_h^{2h} \varphi_h(\boldsymbol{\theta}^\alpha, \cdot))(\mathbf{x}) &= \sum_{\kappa \in J} r_{\kappa} \exp(i(\mathbf{x} + \kappa h) \boldsymbol{\theta}^\alpha / h) \\ &= \underbrace{\sum_{\kappa \in J} r_{\kappa} \exp(i \kappa \boldsymbol{\theta}^\alpha)}_{\tilde{R}_h^{2h}(\boldsymbol{\theta}^\alpha)} \varphi_{2h}(2\boldsymbol{\theta}^{00}, \mathbf{x}) \quad (\mathbf{x} \in G_{2h}) \end{aligned} \quad (7.7)$$

with Fourier symbol $\tilde{R}_h^{2h}(\boldsymbol{\theta}^\alpha)$. (7.7) illustrates the aliasing of the coupled Fourier components on G_{2h} .

The stencil elements r_{κ} for the restrictions under consideration can easily be obtained from Figure 7.2 where the stencils for the adjoint prolongations are given. For example, a proper scaling with $1/4$ yields the four stencil elements of CR in the two-dimensional case:

$$r_{(\pm 1/2, \pm 1/2)} = 1/4.$$

Then, one may calculate the Fourier symbols of CR, BR, WR, and KR using the corresponding stencil representations and (7.7). They are given in Table 7.3.

Of course, the Fourier symbols of CP, BP, WP, KP are implicitly contained in Table 7.3 as well, since for each prolongation there is an adjoint restriction.

To carry out Fourier two- and three-grid analysis for cell-centered multigrid, we simply have to insert the computed Fourier symbols $\tilde{R}_h^{2h}(\boldsymbol{\theta})$ and $\tilde{P}_{2h}^h(\boldsymbol{\theta})$ into the representations of $\tilde{M}_h^{2h}(\boldsymbol{\theta})$ and $\tilde{M}_h^{4h}(\boldsymbol{\theta})$ specified in Chapter 6.

7.3.3 Orders of transfer operators

Recall the definition of high- and low-frequency orders for transfer operators from Section 7.1.2. The high- and low-frequency orders of the transfer operators from Section 7.3.1 are given in Table 7.4. Again, the orders of the

TABLE 7.3: Fourier symbols $\widetilde{R}_h^{2h}(\boldsymbol{\theta})$ and $\widetilde{P}_{2h}^h(\boldsymbol{\theta})$ for CR, BR, WR, KR

	two-dimensional case	three-dimensional case
CR	$\prod_{j=1}^2 \cos\left(\frac{\theta_j}{2}\right)$	$\prod_{j=1}^3 \cos\left(\frac{\theta_j}{2}\right)$
BR/TR	$\prod_{j=1}^2 \cos^3\left(\frac{\theta_j}{2}\right)$	$\prod_{j=1}^3 \cos^3\left(\frac{\theta_j}{2}\right)$
WR	$\cos\left(\frac{\theta_1-\theta_2}{2}\right) \prod_{j=1}^2 \cos\left(\frac{\theta_j}{2}\right)$	$\cos\left(\frac{\theta_1-\theta_2-\theta_3}{2}\right) \prod_{j=1}^3 \cos\left(\frac{\theta_j}{2}\right)$
KR	$\frac{1}{2} \prod_{j=1}^2 \cos\left(\frac{\theta_j}{2}\right) \sum_{i=1}^2 \cos(\theta_i)$	$\frac{1}{3} \prod_{j=1}^3 \cos\left(\frac{\theta_j}{2}\right) \sum_{i=1}^3 \cos(\theta_i)$

TABLE 7.4: Orders for CR, BR, WR, KR

two-dimensional case		CR	BR	WR	KR
low-frequency order	m_r^{low}	2	2	2	2
high-frequency order	m_r^{high}	1	3	2	2
polynomial order	m_r^{poly}	1	2	2	2
three-dimensional case		CR	BR	WR	KR
low-frequency order	m_r^{low}	2	2	2	2
high-frequency order	m_r^{high}	1	3	2	1
polynomial order	m_r^{poly}	1	2	2	1

prolongation operators are implicitly given by the orders of the restriction, as the prolongations are the adjoints of the restrictions. We see here that the local Fourier analysis definition of order contains more information than the pure polynomial one. It gives us $m_r^{\text{high}} = 3$ for BP, which captures the fact that BP is more precise in the sense that it will also interpolate bilinear functions correctly. With respect to the order there is no advantage for KP compared to WP, although its stencil employs more nodes. We also want to point out that the combination (CP, CR) fulfills (7.2) for differential operators with $M = 2$, whereas (7.1) is violated. Note that not only the polynomial but also the high-frequency orders for KR (and KP) are different in the two- and three-dimensional cases.

7.3.4 Numerical experiments

To demonstrate the usefulness of Fourier three-grid analysis for cell-centered multigrid we consider the central second-order discretization of the two-dimen-

sional Poisson equation on the unit square with Dirichlet boundary conditions.

We apply a simple V(1,1)-cycle with point RB-JAC as a smoother together with standard coarsening and DCA. In Table 7.5 we compare the prediction $\rho(M_{3L})$ of the local Fourier analysis with the asymptotic convergence rate $\rho_n(3L)$ for the three-level case and a finest grid Ω_h with $h = 1/128$.

The empirical convergence rates match the prediction from the local Fourier analysis very well in the majority of cases. However, there are three runaways, where the local Fourier analysis is much too optimistic. It can be expected that these discrepancies may be reduced by performing some extra boundary relaxations. We would like to emphasize that the violation of rule (7.1) for the (CP, CR) pair does not lead to a diverging algorithm.

Note that the convergence rates themselves are not as good as expected from the vertex-centered case where one obtains factors around 0.1 for most of the popular transfer operators (for example, the vertex-centered analogs of BP/BR and WP/WR). This is not a standalone difficulty of the smoother,

TABLE 7.5: Two-dimensional Poisson equation: Predictions $\rho(M_{3L})$ versus empirical convergence rates for point RB-JAC smoothing; V(1,1), DCA, $h = 1/128$

	CP	BP	WP	KP	
CR	0.161	0.086	0.204	0.086	local Fourier analysis
BR	0.156	0.194	0.301	0.162	
WR	0.204	0.237	0.331	0.204	
KR	0.204	0.237	0.331	0.204	
CR	0.289	0.155	0.203	0.203	empirical tests
BR	0.155	0.194	0.300	0.236	
WR	0.203	0.236	0.330	0.203	
KR	0.203	0.236	0.331	0.203	

which gives excellent smoothing factors, as can be seen in Table 7.6. Instead it is a problem of the combination of smoother and coarse-grid correction. As a remedy for this Wesseling and Khalil apply in their papers a more powerful relaxation (ILU) which works very well in the cell-centered case. We, however, stay with simple point-relaxation variants. Then, the performance can be improved by changing the cycling structure from a V- to a W-cycle or more efficiently in this situation, since it is less costly and gives a better convergence rate, by the introduction of an overrelaxation parameter ω leading to ω -RB-JAC (we used $\omega = 1.2$ in this example). Similar to the classical SOR situation, this smoother then also reduces low-frequency error components more efficiently which seems to be favorable for this combination of

TABLE 7.6: Local Fourier analysis predictions and empirical convergence rates for two-dimensional Poisson problem; WP, CR, DCA, $h = 1/128$

		$\rho(S^2Q)$	$\rho(M_{2L})$	$\rho(M_{3L})$	$\rho_n(3L)$	$\rho_n(5L)$
GS-LEX	V(1,1)	0.250	0.176	0.231	0.227	0.240
RB-JAC	V(1,1)	0.063	0.148	0.200	0.203	0.218
RB-JAC	W(1,1)	0.063	0.148	0.148	0.148	0.148
ω -RB-JAC	V(1,1)	0.105	0.102	0.120	0.118	0.114

multigrid components. Apparently, the use of ILU relaxation yields a similar implication.

Table 7.6 also shows the superiority of the three-grid Fourier analysis as compared to the two-grid analysis alone. An increase from $\rho(M_{2L})$ to $\rho(M_{3L})$ typically indicates that the convergence of the multigrid method will deteriorate as it becomes “deeper.” We precisely find this behavior for the undamped relaxations, where $\rho_n(5L)$ is larger than $\rho_n(3L)$ for the first two cases, while it remains constant in the third one.

Comparing the performance of GS-LEX and RB-JAC it is interesting to note that the superiority of RB-JAC—known from the vertex-centered case—seems to vanish for certain variants of cell-centered multigrid. This has already been observed by numerical experiments for other types of pattern relaxations like zebra line or plane smoothing compared to the lexicographic variants [26, 43] but could not be explained. Using Fourier three-grid analysis this behavior can be precisely predicted.

7.4 FOURIER ANALYSIS FOR MULTIGRID PRECONDITIONED BY GMRES

We conclude our discussion of further applications of local Fourier analysis with the generalization of the Fourier k -grid analysis to multigrid as a right preconditioner for GMRES(m). For a detailed discussion with respect to the two-grid analysis we refer to [69].

The typical use of Fourier k -grid analysis in a multigrid context is that the spectral radius of the error or defect transformation matrix is obtained theoretically due to a unitary transformation of the k -grid iteration matrix into a sparse block-diagonal matrix. From this block structure one can, however, also calculate the whole spectrum of the k -grid matrix which is used in Section 7.4.2 to obtain asymptotic convergence estimates of multigrid preconditioned GMRES(m). A preconditioned Krylov subspace acceleration method

like GMRES(m) implicitly builds up a minimal residual polynomial. The determination of the polynomial coefficients is easily possible and can be done explicitly, with the help of the Fourier analysis which is shown in Section 7.4.1. Using these coefficients, we derive quantitative convergence estimates for the combined solution method.

As Krylov subspace methods are commonly described using matrix and vector notation, we consider the linear system, $\mathbf{L}\mathbf{u} = \mathbf{f}$, related to the discrete boundary value problem $L_h u_h = f_h$ with eliminated boundary conditions. Recall the usual Euclidean inner product of two vectors $\mathbf{v} = (v_1, \dots, v_N)^T$, $\mathbf{w} = (w_1, \dots, w_N)^T \in \mathbb{C}^N$ which is defined by

$$\langle \mathbf{v}, \mathbf{w} \rangle_2 := \sum_{i=1}^N v_i \overline{w}_i.$$

The induced Euclidean 2-norm is then given by $\|\mathbf{v}\|_2 := \sqrt{\langle \mathbf{v}, \mathbf{v} \rangle_2}$.

A three-grid (or more general a multigrid) cycle can be described by the matrix splitting, $\mathbf{C}\mathbf{u}^{(i)} + (\mathbf{L} - \mathbf{C})\mathbf{u}^{(i-1)} = \mathbf{f}$, where $\mathbf{u}^{(i)}$ and $\mathbf{u}^{(i-1)}$ represent a new and a previous approximation. This formulation is equivalent to

$$\mathbf{u}^{(i)} = \mathbf{u}^{(i-1)} + \mathbf{C}^{-1}(\mathbf{f} - \mathbf{L}\mathbf{u}^{(i-1)}) \quad \text{and} \quad \mathbf{r}^{(i)} = (\mathbf{I} - \mathbf{L}\mathbf{C}^{-1})\mathbf{r}^{(i-1)}$$

with the residual vectors $\mathbf{r}^{(i)}, \mathbf{r}^{(i-1)}$ and the residual transformation matrix

$$\mathbf{I} - \mathbf{L}\mathbf{C}^{-1} = \mathbf{L}\mathbf{M}\mathbf{L}^{-1}, \quad (7.8)$$

where \mathbf{M} denotes the three-grid iteration matrix. GMRES(m) searches for a new approximation $\mathbf{u}^{(i)}$ with corresponding residual $\mathbf{r}^{(i)}$ in the Krylov subspace

$$K^m(\mathbf{L}\mathbf{C}^{-1}, \mathbf{r}^{(i-m)}) := \text{span}[\mathbf{r}^{(i-m)}, (\mathbf{L}\mathbf{C}^{-1})\mathbf{r}^{(i-m)}, \dots, (\mathbf{L}\mathbf{C}^{-1})^{m-1}\mathbf{r}^{(i-m)}].$$

It selects the new solution by minimizing the residual in the discrete Euclidean 2-norm

$$\|\mathbf{r}^{(i)}\|_2 = \min\{\|P_m(\mathbf{L}\mathbf{C}^{-1})\mathbf{r}^{(i-m)}\|_2 \mid P_m \in \Pi_m\}, \quad (7.9)$$

where Π_m denotes the set of all polynomials of degree at most m with $P_m(0) = 1$. For convenience, $i \geq m$ is assumed. Since we are interested in the asymptotic convergence of multigrid preconditioned GMRES with a restart after m iterations, we focus on the residuals $\mathbf{r}^{(m)}, \mathbf{r}^{(2m)}, \dots, \mathbf{r}^{(j \cdot m)}$. Then, a “complete” iteration with iteration index j consists of m multigrid preconditioned GMRES(m) steps. The GMRES(m)-polynomial which characterizes the j -th complete iteration is denoted by $P_m^{(j)}$, leading to the following recursion for the corresponding residual

$$\mathbf{r}^{(j \cdot m)} = P_m^{(j)}(\mathbf{L}\mathbf{C}^{-1})\mathbf{r}^{((j-1) \cdot m)}. \quad (7.10)$$

As unitary transformations do not affect the convergence properties of GMRES, we consider the Fourier representations

$$\widetilde{\mathbf{M}}^{3g} := \text{bdiag} \left\{ [M^{3g}(\boldsymbol{\theta})]_{\boldsymbol{\theta} \in \Theta_{3g}^{\mathbb{P}}} \right\} \quad \text{and} \quad \widetilde{\mathbf{L}}^{3g} := \text{bdiag} \left\{ [L^{3g}(\boldsymbol{\theta})]_{\boldsymbol{\theta} \in \Theta_{3g}^{\mathbb{P}}} \right\}$$

(compare with Section 6.5) instead of the representations \mathbf{M} and \mathbf{L} with respect to the Euclidean basis.

7.4.1 Analysis based on the GMRES(m)-polynomial

To obtain a *quantitative* result for the convergence of GMRES(m) preconditioned by multigrid, we present an approach which explicitly depends on the iteration index j . Assuming a repeated application of preconditioned GMRES(m), the following function g has to be minimized in order to find the coefficients $\alpha_n^{(j)}$ ($n = 1, \dots, m$) of the j -th GMRES(m)-polynomial $P_m^{(j)}$ (see (7.8), (7.9), and (7.10)):

$$g(\alpha_1^{(j)}, \dots, \alpha_m^{(j)}) := \left\langle P_m^{(j)} \left(\mathbf{I} - \widetilde{\mathbf{L}}^{3g} \widetilde{\mathbf{M}}^{3g} (\widetilde{\mathbf{L}}^{3g})^{-1} \right) \tilde{\mathbf{r}}^{((j-1)m)}, P_m^{(j)} \left(\mathbf{I} - \widetilde{\mathbf{L}}^{3g} \widetilde{\mathbf{M}}^{3g} (\widetilde{\mathbf{L}}^{3g})^{-1} \right) \tilde{\mathbf{r}}^{((j-1)m)} \right\rangle_2.$$

The $\alpha_n^{(j)}$ ($n = 1, \dots, m$) are obtained by solving the linear system

$$\begin{aligned} \frac{\partial g}{\partial \alpha_\ell^{(j)}} = & 2 \sum_{n=1}^m \alpha_n^{(j)} \left\langle (\mathbf{I} - \widetilde{\mathbf{L}}^{3g} \widetilde{\mathbf{M}}^{3g} (\widetilde{\mathbf{L}}^{3g})^{-1})^\ell \tilde{\mathbf{r}}^{((j-1)m)}, (\mathbf{I} - \widetilde{\mathbf{L}}^{3g} \widetilde{\mathbf{M}}^{3g} (\widetilde{\mathbf{L}}^{3g})^{-1})^n \tilde{\mathbf{r}}^{((j-1)m)} \right\rangle_2 \\ & + 2 \left\langle (\mathbf{I} - \widetilde{\mathbf{L}}^{3g} \widetilde{\mathbf{M}}^{3g} (\widetilde{\mathbf{L}}^{3g})^{-1})^\ell \tilde{\mathbf{r}}^{((j-1)m)}, \tilde{\mathbf{r}}^{((j-1)m)} \right\rangle_2 = 0 \quad \text{for } \ell = 1, \dots, m. \end{aligned} \quad (7.11)$$

The solution of (7.11) can easily be computed for every iteration index j due to the sparse block structure of $(\mathbf{I} - \widetilde{\mathbf{L}}^{3g} \widetilde{\mathbf{M}}^{3g} (\widetilde{\mathbf{L}}^{3g})^{-1})^\ell$ ($\ell = 1, \dots, m$), if the previous Fourier transformed residual $\tilde{\mathbf{r}}^{((j-1)m)}$ is given. We simply prescribe a randomly chosen initial residual $\tilde{\mathbf{r}}^{(0)}$. This allows the calculation of $\alpha_\ell^{(1)}$ ($\ell = 1, \dots, m$) by (7.11) and gives $\tilde{\mathbf{r}}^{(1 \cdot m)} = P_m^{(1)} (\mathbf{I} - \widetilde{\mathbf{L}}^{3g} \widetilde{\mathbf{M}}^{3g} (\widetilde{\mathbf{L}}^{3g})^{-1}) \tilde{\mathbf{r}}^{(0)}$. Then, the computation of $\tilde{\mathbf{r}}^{(j \cdot m)}$ for $j > 1$ is straightforward due to its recursive definition, see (7.10). This leads to an average reduction factor $\rho_3^{\text{acc}}(j, m)$ for a complete iteration which can be obtained by the three-grid Fourier analysis:

$$\rho_3^{\text{acc}}(j, m) := \left[\left(\frac{\|\tilde{\mathbf{r}}^{(j \cdot m)}\|_2}{\|\tilde{\mathbf{r}}^{(0)}\|_2} \right)^{1/m} \right]^{1/j}. \quad (7.12)$$

The superscript “acc” is used as an abbreviation for “accelerated,” since the combination of multigrid and GMRES(m) can be interpreted as an acceleration of the multigrid convergence speed by an additional application of GMRES(m).

$\rho_3^{\text{acc}}(j, m)$ usually tends to a constant for $j \geq 20$. Moreover, all resulting GMRES(m) polynomials are almost identical for $j \geq 20$, i.e., $P_m^{(j)}$ becomes “stationary.” The particular choice of the initial residual $\tilde{\mathbf{r}}^{(0)}$ does not influence the average reduction factors for $j \gg 1$, which has been confirmed by systematic test calculations [69]. Thus, it is expected that $\rho_3^{\text{acc}}(m) := \rho_3^{\text{acc}}(20, m)$ matches well with numerical reference values.

Remark 7.5 (Relation to optimal weighting parameters). From the explicit calculation of the polynomial coefficients it is possible to derive a relation between multiparameter methods like multistage smoothers and an acceleration with GMRES(m). For example, m -stage point RB-JAC relaxation applied to the biharmonic equation—discussed in Section 4.1.5—is connected to multigrid preconditioned GMRES(m) in the following sense: The optimal multistage parameters constituting the m -stage polynomial (3.13) can be expressed by the coefficients of the “stationary” minimal residual polynomial. Similarly, it is possible to establish a relation between an overweighting of residuals and an acceleration with GMRES(m). More generally, it can be concluded that the Krylov subspace acceleration implicitly improves the coarse-grid correction or the relaxation procedure if one of these multigrid components clearly hampers the overall multigrid convergence. This acceleration is particularly useful as it is applicable to situations in which it is not easy to tune a multigrid method. Details are given in [67, 69]. ►

7.4.2 Analysis based on the spectrum of the residual transformation matrix

If we use the above Fourier representation $\mathbf{I} - \tilde{\mathbf{L}}^{3g} \tilde{\mathbf{M}}^{3g} (\tilde{\mathbf{L}}^{3g})^{-1}$ the corresponding spectrum can be calculated numerically. The common way (see for example, [52, 53]) to analyze the convergence of GMRES is to exploit information about the spectrum σ of the iteration matrix $\mathbf{L}\mathbf{C}^{-1}$.

Suppose that all eigenvalues of $\mathbf{I} - \tilde{\mathbf{L}}^{3g} \tilde{\mathbf{M}}^{3g} (\tilde{\mathbf{L}}^{3g})^{-1}$ are located in an ellipse $E(c, d, a)$ which excludes the origin. $(c, 0)$ denotes the center, d the focal distance, and a the major semi-axis. Note that σ is always symmetric with respect to the real axis, so we only consider ellipses which are aligned with the axes and where the ordinate of the center equals zero. Then, it is known [53] that the absolute value of the polynomial

$$t_m(z) := T_m\left(\frac{c}{d} - \frac{1}{d}z\right) / T_m\left(\frac{c}{d}\right) = T_m(\hat{z}) / T_m\left(\frac{c}{d}\right) \quad \text{with} \quad z, \hat{z} := \left(\frac{c}{d} - \frac{1}{d}z\right) \in \mathbb{C}$$

is small on the spectrum of $\mathbf{I} - \tilde{\mathbf{L}}^{3g} \tilde{\mathbf{M}}^{3g} (\tilde{\mathbf{L}}^{3g})^{-1}$. Here, T_m represents the Chebychev polynomial of degree m of the first kind, see [53].

If $\mathbf{I} - \tilde{\mathbf{L}}^{3g} \tilde{\mathbf{M}}^{3g} (\tilde{\mathbf{L}}^{3g})^{-1}$ is diagonalizable, i.e., $\mathbf{I} - \tilde{\mathbf{L}}^{3g} \tilde{\mathbf{M}}^{3g} (\tilde{\mathbf{L}}^{3g})^{-1} = \mathbf{X} \mathbf{D} \mathbf{X}^{-1}$ with diagonal matrix \mathbf{D} , then (7.9) yields

$$\begin{aligned} \|\tilde{\mathbf{r}}^{(j,m)}\|_2 &\leq \|t_m(\mathbf{I} - \tilde{\mathbf{L}}^{3g} \tilde{\mathbf{M}}^{3g} (\tilde{\mathbf{L}}^{3g})^{-1})\|_2 \|\tilde{\mathbf{r}}^{((j-1)m)}\|_2 \\ &\leq \left[\|t_m(\mathbf{I} - \tilde{\mathbf{L}}^{3g} \tilde{\mathbf{M}}^{3g} (\tilde{\mathbf{L}}^{3g})^{-1})\|_2 \right]^j \|\tilde{\mathbf{r}}^{(0)}\|_2 \\ &\leq \left[\kappa_2(\mathbf{X}) T_m\left(\frac{a}{d}\right) / T_m\left(\frac{c}{d}\right) \right]^j \|\tilde{\mathbf{r}}^{(0)}\|_2, \end{aligned}$$

where $\kappa_2(\mathbf{X})$ denotes the spectral condition number of the transformation matrix \mathbf{X} [53]. Using these inequalities, we obtain for an arbitrary complete iteration j

$$\rho_3^{\text{acc}}(j, m) \leq N_m^{\text{E}} \leq (\kappa_2(\mathbf{X}))^{1/m} T_m^{\text{E}} \quad \text{with} \quad (7.13)$$

$$N_m^{\text{E}} := (\|t_m(\mathbf{I} - \tilde{\mathbf{L}}^{3g} \tilde{\mathbf{M}}^{3g} (\tilde{\mathbf{L}}^{3g})^{-1})\|_2)^{1/m} \quad \text{and} \quad T_m^{\text{E}} := \left(T_m\left(\frac{a}{d}\right) / T_m\left(\frac{c}{d}\right) \right)^{1/m},$$

as approximations for the average reduction factors of m multigrid preconditioned GMRES(m) steps (see (7.12)).

In [54] it is stated that (7.13) is an asymptotic result and that the actual residual norm should rather behave like T_m^{E} without $\kappa_2(\mathbf{X})$. This presumption, in connection with the two-grid Fourier analysis, is validated in [69] where it is found that the heuristic estimate T_m^{E} gives a certain insight into the asymptotic accelerated convergence whereas the upper bound N_m^{E} is too pessimistic in general. However, the main focus lies on $\rho_3^{\text{acc}}(m)$.

Remark 7.6 (Determination of ellipses). We would like to point out that the eigenvalue distribution of the three-grid error transformation matrix $\tilde{\mathbf{M}}^{3g}$ provided by the accompanying software might be used to determine the ellipses described above. Note that the spectrum of $\mathbf{I} - \tilde{\mathbf{L}}^{3g} \tilde{\mathbf{M}}^{3g} (\tilde{\mathbf{L}}^{3g})^{-1}$ is simply shifted along the real axis by $(1, 0)^T$ compared to the spectrum of $\tilde{\mathbf{M}}^{3g}$. ►

Appendix A

FOURIER REPRESENTATION OF RELAXATION

Appendix A provides the Fourier representations $\tilde{S}_{\mathbf{h}}^{\text{RB}}(\boldsymbol{\theta}, \omega)$ of block RB-JAC relaxations for two- and three-dimensional problems. Each smoothing method under consideration consists of two partial steps. The corresponding Fourier symbols are denoted by $\tilde{S}_{\mathbf{h}}^{\text{R}}(\boldsymbol{\theta}, \omega) = \frac{1}{2}(r_{ij})_{i,j=1,\dots,2^d}$ and $\tilde{S}_{\mathbf{h}}^{\text{B}}(\boldsymbol{\theta}, \omega) = \frac{1}{2}(b_{ij})_{i,j=1,\dots,2^d}$; compare with Section 5.4.4.

We use notation (5.6) in order to characterize the different smoothing methods. Note that it is sufficient to specify \mathcal{I}^{P} from Definition 5.5 to obtain a unique characterization. Then, the two disjoint subsets $G_{\mathbf{h}}^{\text{R}}$ and $G_{\mathbf{h}}^{\text{B}}$ —specifying $\tilde{G}_{\mathbf{h}}$ from (5.6) for each partial step—are given by

$$G_{\mathbf{h}}^{\text{R}} = \{\mathbf{x} = \boldsymbol{\kappa}\mathbf{h} \in G_{\mathbf{h}} \mid \sum_{j \in \mathcal{I}^{\text{P}}} \kappa_j \text{ even}\}, \quad G_{\mathbf{h}}^{\text{B}} = \{\mathbf{x} = \boldsymbol{\kappa}\mathbf{h} \in G_{\mathbf{h}} \mid \sum_{j \in \mathcal{I}^{\text{P}}} \kappa_j \text{ odd}\}.$$

Furthermore, the subspace J^0 , which characterizes those grid points where the unknowns are smoothed simultaneously in a block, reads as

$$J^0 = \{\boldsymbol{\kappa} \in J \mid \kappa_j = 0 \text{ for } j \in \mathcal{I}^{\text{P}}\}.$$

Recall that for RB-JAC relaxations we have $J^- = J \setminus J^0$ and $J^+ = \emptyset$. Then it is possible to calculate the Fourier symbols A^{α} of the corresponding block JAC relaxations—which constitute the entries of $\tilde{S}_{\mathbf{h}}^{\text{R}}(\boldsymbol{\theta}, \omega)$ and $\tilde{S}_{\mathbf{h}}^{\text{B}}(\boldsymbol{\theta}, \omega)$ —as in (5.10):

$$A^{\alpha} := A(\boldsymbol{\theta}^{\alpha}, \omega) = 1 - \omega \frac{\tilde{L}_{\mathbf{h}}(\boldsymbol{\theta})}{\tilde{L}_{\mathbf{h}}^0(\boldsymbol{\theta})} \quad (\boldsymbol{\theta}^{(0\dots 0)} \in \Theta_{2\mathbf{h}})$$

with $\tilde{L}_{\mathbf{h}}(\boldsymbol{\theta}) = \sum_{\boldsymbol{\kappa} \in J} l_{\boldsymbol{\kappa}} \exp(\mathbf{i} \boldsymbol{\theta} \boldsymbol{\kappa})$ and $\tilde{L}_{\mathbf{h}}^0(\boldsymbol{\theta}) = \sum_{\boldsymbol{\kappa} \in J^0} l_{\boldsymbol{\kappa}} \exp(\mathbf{i} \boldsymbol{\theta} \boldsymbol{\kappa})$.

Summarizing, for each of the following block relaxations we simply have to specify \mathcal{I}^{P} , $\tilde{S}_{\mathbf{h}}^{\text{R}}(\boldsymbol{\theta}, \omega)$ and $\tilde{S}_{\mathbf{h}}^{\text{B}}(\boldsymbol{\theta}, \omega)$.

A.1 Two-dimensional case

x_1 -Line RB-JAC Relaxation: $\mathcal{I}^P = \{2\}$

$$\begin{aligned}\tilde{S}_h^R(\boldsymbol{\theta}, \omega) &= \frac{1}{2} \begin{pmatrix} A^{00} + 1 & 0 & 0 & A^{01} - 1 \\ 0 & A^{11} + 1 & A^{10} - 1 & 0 \\ 0 & A^{11} - 1 & A^{10} + 1 & 0 \\ A^{00} - 1 & 0 & 0 & A^{01} + 1 \end{pmatrix}, \\ \tilde{S}_h^B(\boldsymbol{\theta}, \omega) &= \frac{1}{2} \begin{pmatrix} A^{00} + 1 & 0 & 0 & -A^{01} + 1 \\ 0 & A^{11} + 1 & -A^{10} + 1 & 0 \\ 0 & -A^{11} + 1 & A^{10} + 1 & 0 \\ -A^{00} + 1 & 0 & 0 & A^{01} + 1 \end{pmatrix}.\end{aligned}$$

x_2 -Line RB-JAC Relaxation: $\mathcal{I}^P = \{1\}$

$$\begin{aligned}\tilde{S}_h^R(\boldsymbol{\theta}, \omega) &= \frac{1}{2} \begin{pmatrix} A^{00} + 1 & 0 & A^{10} - 1 & 0 \\ 0 & A^{11} + 1 & 0 & A^{01} - 1 \\ A^{00} - 1 & 0 & A^{10} + 1 & 0 \\ 0 & A^{11} - 1 & 0 & A^{01} + 1 \end{pmatrix}, \\ \tilde{S}_h^B(\boldsymbol{\theta}, \omega) &= \frac{1}{2} \begin{pmatrix} A^{00} + 1 & 0 & -A^{10} + 1 & 0 \\ 0 & A^{11} + 1 & 0 & -A^{01} + 1 \\ -A^{00} + 1 & 0 & A^{10} + 1 & 0 \\ 0 & -A^{11} + 1 & 0 & A^{01} + 1 \end{pmatrix}.\end{aligned}$$

A.2 Three-dimensional case

Due to space limitations, we only specify the nonzero entries r_{ij} and b_{ij} of $\tilde{S}_h^R(\boldsymbol{\theta}, \omega)$ and $\tilde{S}_h^B(\boldsymbol{\theta}, \omega)$, respectively, in the three-dimensional case.

x_1 -Line RB-JAC Relaxation: $\mathcal{I}^P = \{2, 3\}$

$$\begin{aligned}r_{11} &= A^{000} + 1, & r_{14} &= A^{011} - 1, & r_{22} &= A^{111} + 1, & r_{23} &= A^{100} - 1, \\ r_{32} &= A^{111} - 1, & r_{33} &= A^{100} + 1, & r_{41} &= A^{000} - 1, & r_{44} &= A^{011} + 1, \\ r_{55} &= A^{010} + 1, & r_{57} &= A^{001} - 1, & r_{66} &= A^{101} + 1, & r_{68} &= A^{110} - 1, \\ r_{75} &= A^{010} - 1, & r_{77} &= A^{001} + 1, & r_{86} &= A^{101} - 1, & r_{88} &= A^{110} + 1.\end{aligned}$$

$$\begin{aligned}
b_{11} &= A^{000} + 1, & b_{14} &= -A^{011} + 1, & b_{22} &= A^{111} + 1, & b_{23} &= -A^{100} + 1, \\
b_{32} &= -A^{111} + 1, & b_{33} &= A^{100} + 1, & b_{41} &= -A^{000} + 1, & b_{44} &= A^{011} + 1, \\
b_{55} &= A^{010} + 1, & b_{57} &= -A^{001} + 1, & b_{66} &= A^{101} + 1, & b_{68} &= -A^{110} + 1, \\
b_{75} &= -A^{010} + 1, & b_{77} &= A^{001} + 1, & b_{86} &= -A^{101} + 1, & b_{88} &= A^{110} + 1.
\end{aligned}$$

x_2 -Line RB-JAC Relaxation: $\mathcal{I}^P = \{1, 3\}$

$$\begin{aligned}
r_{11} &= A^{000} + 1, & r_{16} &= A^{101} - 1, & r_{22} &= A^{111} + 1, & r_{25} &= A^{010} - 1, \\
r_{33} &= A^{100} - 1, & r_{37} &= A^{001} + 1, & r_{44} &= A^{011} - 1, & r_{48} &= A^{110} + 1, \\
r_{52} &= A^{111} + 1, & r_{55} &= A^{010} - 1, & r_{61} &= A^{000} + 1, & r_{66} &= A^{101} - 1, \\
r_{73} &= A^{100} - 1, & r_{77} &= A^{001} + 1, & r_{84} &= A^{011} - 1, & r_{88} &= A^{110} + 1.
\end{aligned}$$

$$\begin{aligned}
b_{11} &= A^{000} + 1, & b_{16} &= -A^{101} + 1, & b_{22} &= A^{111} + 1, & b_{25} &= -A^{010} + 1, \\
b_{33} &= -A^{100} + 1, & b_{37} &= A^{001} + 1, & b_{44} &= -A^{011} + 1, & b_{48} &= A^{110} + 1, \\
b_{52} &= A^{111} + 1, & b_{55} &= -A^{010} + 1, & b_{61} &= A^{000} + 1, & b_{66} &= -A^{101} + 1, \\
b_{73} &= -A^{100} + 1, & b_{77} &= A^{001} + 1, & b_{84} &= -A^{011} + 1, & b_{88} &= A^{110} + 1.
\end{aligned}$$

x_3 -Line RB-JAC Relaxation: $\mathcal{I}^P = \{1, 2\}$

$$\begin{aligned}
r_{11} &= A^{000} + 1, & r_{18} &= A^{110} - 1, & r_{22} &= A^{111} + 1, & r_{27} &= A^{001} - 1, \\
r_{33} &= A^{100} - 1, & r_{35} &= A^{010} + 1, & r_{44} &= A^{011} - 1, & r_{46} &= A^{101} + 1, \\
r_{53} &= A^{100} + 1, & r_{55} &= A^{010} - 1, & r_{64} &= A^{011} + 1, & r_{66} &= A^{101} - 1, \\
r_{72} &= A^{111} - 1, & r_{77} &= A^{001} + 1, & r_{81} &= A^{000} - 1, & r_{88} &= A^{110} + 1.
\end{aligned}$$

$$\begin{aligned}
b_{11} &= A^{000} + 1, & b_{18} &= -A^{110} + 1, & b_{22} &= A^{111} + 1, & b_{27} &= -A^{001} + 1, \\
b_{33} &= -A^{100} + 1, & b_{35} &= A^{010} + 1, & b_{44} &= -A^{011} + 1, & b_{46} &= A^{101} + 1, \\
b_{53} &= A^{100} + 1, & b_{55} &= -A^{010} + 1, & b_{64} &= A^{011} + 1, & b_{66} &= -A^{101} + 1, \\
b_{72} &= -A^{111} + 1, & b_{77} &= A^{001} + 1, & b_{81} &= -A^{000} + 1, & b_{88} &= A^{110} + 1.
\end{aligned}$$

(x_1, x_2) -Plane RB-JAC Relaxation: $\mathcal{I}^P = \{3\}$

$$\begin{aligned}
r_{11} &= A^{000} + 1, & r_{17} &= A^{001} - 1, & r_{22} &= A^{111} + 1, & r_{28} &= A^{110} - 1, \\
r_{33} &= A^{100} - 1, & r_{36} &= A^{101} + 1, & r_{44} &= A^{011} - 1, & r_{45} &= A^{010} + 1, \\
r_{54} &= A^{011} + 1, & r_{55} &= A^{010} - 1, & r_{63} &= A^{100} + 1, & r_{66} &= A^{101} - 1, \\
r_{71} &= A^{000} - 1, & r_{77} &= A^{001} + 1, & r_{82} &= A^{111} - 1, & r_{88} &= A^{110} + 1.
\end{aligned}$$

$$\begin{aligned}
b_{11} &= A^{000} + 1, & b_{17} &= -A^{001} - 1, & b_{22} &= A^{111} + 1, & b_{28} &= -A^{110} - 1, \\
b_{33} &= -A^{100} - 1, & b_{36} &= A^{101} + 1, & b_{44} &= -A^{011} - 1, & b_{45} &= A^{010} + 1, \\
b_{54} &= A^{011} + 1, & b_{55} &= -A^{010} - 1, & b_{63} &= A^{100} + 1, & b_{66} &= -A^{101} - 1, \\
b_{71} &= -A^{000} - 1, & b_{77} &= A^{001} + 1, & b_{82} &= -A^{111} - 1, & b_{88} &= A^{110} + 1.
\end{aligned}$$

(x_1, x_3) -Plane RB-JAC Relaxation: $\mathcal{I}^P = \{2\}$

$$\begin{aligned}
r_{11} &= A^{000} + 1, & r_{15} &= A^{010} - 1, & r_{22} &= A^{111} + 1, & r_{26} &= A^{101} - 1, \\
r_{33} &= A^{100} - 1, & r_{38} &= A^{110} + 1, & r_{44} &= A^{011} - 1, & r_{47} &= A^{001} + 1, \\
r_{51} &= A^{000} + 1, & r_{55} &= A^{010} - 1, & r_{62} &= A^{111} + 1, & r_{66} &= A^{101} - 1, \\
r_{74} &= A^{011} - 1, & r_{77} &= A^{001} + 1, & r_{83} &= A^{100} - 1, & r_{88} &= A^{110} + 1.
\end{aligned}$$

$$\begin{aligned}
b_{11} &= A^{000} + 1, & b_{15} &= -A^{010} - 1, & b_{22} &= A^{111} + 1, & b_{26} &= -A^{101} - 1, \\
b_{33} &= -A^{100} - 1, & b_{38} &= A^{110} + 1, & b_{44} &= -A^{011} - 1, & b_{47} &= A^{001} + 1, \\
b_{51} &= A^{000} + 1, & b_{55} &= -A^{010} - 1, & b_{62} &= A^{111} + 1, & b_{66} &= -A^{101} - 1, \\
b_{74} &= -A^{011} - 1, & b_{77} &= A^{001} + 1, & b_{83} &= -A^{100} - 1, & b_{88} &= A^{110} + 1.
\end{aligned}$$

(x_2, x_3) -Plane RB-JAC Relaxation: $\mathcal{I}^P = \{1\}$

$$\begin{aligned}
r_{11} &= A^{000} + 1, & r_{13} &= A^{100} - 1, & r_{22} &= A^{111} + 1, & r_{24} &= A^{011} - 1, \\
r_{31} &= A^{000} - 1, & r_{33} &= A^{100} + 1, & r_{42} &= A^{111} - 1, & r_{44} &= A^{011} + 1, \\
r_{55} &= A^{010} + 1, & r_{58} &= A^{110} - 1, & r_{66} &= A^{101} + 1, & r_{67} &= A^{001} - 1, \\
r_{76} &= A^{101} - 1, & r_{77} &= A^{001} + 1, & r_{85} &= A^{010} - 1, & r_{88} &= A^{110} + 1.
\end{aligned}$$

$$\begin{aligned}
b_{11} &= A^{000} + 1, & b_{13} &= -A^{100} + 1, & b_{22} &= A^{111} + 1, & b_{24} &= -A^{011} + 1, \\
b_{31} &= -A^{000} + 1, & b_{33} &= A^{100} + 1, & b_{42} &= -A^{111} + 1, & b_{44} &= A^{011} + 1, \\
b_{55} &= A^{010} + 1, & b_{58} &= -A^{110} + 1, & b_{66} &= A^{101} + 1, & b_{67} &= -A^{001} + 1, \\
b_{76} &= -A^{101} + 1, & b_{77} &= A^{001} + 1, & b_{85} &= -A^{010} + 1, & b_{88} &= A^{110} + 1.
\end{aligned}$$

REFERENCES

- [1] Alcouffe, R.E., Brandt, A., Dendy, J.E., and Painter, J.W., The multi-grid methods for the diffusion equation with strongly discontinuous coefficients, *SIAM J. Sci. Stat. Comput.*, 2 (1981), pp. 430–454.
- [2] Börgers, C., Mehrgitterverfahren für eine Mehrstellendiskretisierung der Poissongleichung und für eine zweidimensionale singulär gestörte Aufgabe, Master's thesis (in German), University of Bonn, Germany, 1981.
- [3] Bramble, J., Ewing, R., Pasciak, J., and Shen, J., The analysis of multi-grid algorithms for cell centered finite difference methods, *Advance in Computational Mathematics*, 5 (1996), pp. 15–29.
- [4] Brandt, A., Multi-level adaptive solutions to boundary-value problems, *Math. Comp.*, 31 (1977), pp. 333–390.
- [5] Brandt, A., Multigrid solvers on parallel computers, in: *Elliptic Problem Solvers*, Schultz, M. (ed.), Academic Press, New York, 1980.
- [6] Brandt, A., Multigrid solvers for non-elliptic and singular-perturbation steady-state problems, Report, Dept. of Appl. Math. The Weizmann Inst. of Science, Rehovot, Israel (1981), unpublished.
- [7] Brandt, A., Multigrid techniques: 1984 guide with applications to fluid dynamics, GMD-Studie Nr. 85, Sankt Augustin, Germany (1984).
- [8] Brandt, A., Rigorous local mode analysis of multigrid, in: *Prelim. Proc. 4th Copper Mountain Conference on Multigrid Methods*, Copper Mountain, CO, April 1989, pp. 55–133.
- [9] Brandt, A., Rigorous quantitative analysis of multigrid, I: Constant coefficients two-level cycle with L_2 -norm, *SIAM J. Numer. Anal.*, 31 (1994), pp. 1695–1730.
- [10] Brandt, A. and Dinar, N., Multigrid solutions to elliptic flow problems, in: *Numerical Methods for Partial Differential Equations*, Parter, S. (ed.), Academic Press, New York, 1979, pp. 53–147.
- [11] Brandt, A. and Yavneh, I. On multigrid solution of high-Reynolds incompressible entering flows, *J. Comput. Phys.*, 101 (1992), pp. 151–164.
- [12] Brandt, A. and Yavneh, I. Accelerated multigrid convergence and high-Reynolds recirculating flows, *SIAM J. Sci. Comput.*, 14 (1993), pp. 607–626.

- [13] Briggs, W.L., van Henson, E., and McCormick, S.F., *A Multigrid Tutorial*, Second Edition, SIAM, Philadelphia, 2000.
- [14] Bröker, O., Grote, M.J., Mayer, C., and Reusken, A., Robust parallel smoothing for multigrid via sparse approximate inverses, *SIAM J. Sci. Comput.*, 23 (2001), pp. 1396–1417.
- [15] Catalano, L.A., Napolitano, M., and Deconinck, H., Optimal multi-stage schemes for multigrid smoothing of two-dimensional advection operators, *Comm. Appl. Num. Meth.*, 8 (1992), pp. 785–795.
- [16] Chan, T.F. and Elman, H.C., Fourier analysis of iterative methods for elliptic problems, *SIAM Review*, 31 (1989), pp. 20–49.
- [17] Dendy Jr., J.E., Black box multigrid, *J. Comp. Phys.*, 48 (1982), pp. 366–386.
- [18] Dendy Jr., J.E., Black box multigrid for nonsymmetric problems, *Appl. Math. Comput.*, 13 (1983), pp. 261–283.
- [19] Dick, E. and Linden, J., A multigrid method for steady incompressible Navier-Stokes equations based on flux difference splitting, *Int. J. Num. Meth. Fluids*, 14 (1992), pp. 1311–1323.
- [20] Dick, E. and Rienslagh, K., Multi-staging of Jacobi relaxation to improve smoothing properties of multigrid methods for steady Euler equations, *J. Comp. Appl. Math.*, 50 (1994), pp. 241–254.
- [21] Diskin, B. and Thomas, J.L., *SIAM J. Sci. Comput.*, 22 (2000), pp. 633–655.
- [22] Elman, H., Ernst, O.G., and OLeary, D.P., A multigrid method enhanced by Krylov subspace iteration for discrete Helmholtz equations, *SIAM J. Sci. Comput.*, 23 (2001), pp. 1290–1314.
- [23] Frohn-Schauf, C., Flux-Splitting-Methoden und Mehrgitterverfahren für hyperbolische Systeme mit Beispielen aus der Strömungsmechanik, Ph.D. Thesis (in German), University of Düsseldorf, Germany, 1992.
- [24] Gaspar, F.J., Lisbona, F.J., Oosterlee, C.W., and Wienands, R., Systematic comparison of coupled and distributive smoothing in multigrid for the poroelasticity system, accepted for publication in *Numerical Linear Algebra with Applications*.
- [25] Garabedian, P.R., *Partial Differential Equations*, Wiley, New York, 1964.
- [26] Gjesdal, T., Smoothing analysis of multicolour pattern schemes, *J. Comput. Appl. Math.*, 85 (1997), pp. 345–350.
- [27] Grote, M.J. and Huckle, T., Parallel preconditioning with sparse approximate inverses, *SIAM J. Sci. Comput.*, 21 (2000), pp. 838–853.

- [28] Hackbusch, W., *Multi-grid methods and applications*, Springer, Berlin, 1985.
- [29] Hemker, P.W., Fourier analysis of grid functions, prolongations, and restrictions, Report NW 98, CWI, Amsterdam, The Netherlands, November 1980.
- [30] Hemker, P.W., On the order of prolongations and restrictions in multi-grid procedures, *J. Comput. Appl. Math.*, 32 (1990), pp. 423–429.
- [31] Jameson, A., Schmidt, W., and Turkel, E., Numerical simulation of the Euler equations by finite volume methods using Runge-Kutta time stepping schemes, AIAA-81-1259 (1981).
- [32] Joppich, W. and Lorentz, R.A., High order positive, monotone and convex multigrid interpolations, *COMPEL*, Vol. 12, No. 1, March 1993
- [33] Joppich, W. and Mijalković, S., *Multigrid Methods for Process Simulation*, Springer, Vienna, 1993.
- [34] Khalil, M., Analysis of linear multigrid methods for elliptic differential equations with discontinuous and anisotropic coefficients, Ph.D. Thesis, Technical University Delft, The Netherlands, 1989.
- [35] Khalil, M. and Wesseling, P., Vertex-centered and cell-centered multigrid for interface problems, *J. Comput. Phys.*, 98 (1992), pp. 1–20.
- [36] Kuo, C.-C.J. and Levy, B.C., Two-color Fourier analysis of the multigrid method with red-black Gauss–Seidel smoothing, *Appl. Math. Comput.*, 29 (1989), pp. 69–87.
- [37] Kwak, D.Y., V-cycle multigrid for cell-centered finite differences, *SIAM J. Sci. Comput.*, 21 (1999), pp. 552–564.
- [38] Kwak, D.Y., V-cycle multigrid convergence for cell centered finite difference method, 3-D case, in: *Fourteenth International Conference on Domain Decomposition Methods*, Herrera, I., Keyes, D., Widlund, O., and Yates, R. (eds.), available online at *DDM.org*.
- [39] van Leer, B., Upwind-difference methods for aerodynamic problems governed by the Euler equations, in: *Large Scale Computations in Fluid Mechanics*, Lectures in Appl. Math. 22, II, Enquist, B., Osher, S., and Somerville, R. (eds.), AMS, Providence, RI, 1985, pp. 327–336.
- [40] van Leer, B., Lee, W., Roe, P.L., Powell, K.G., and Tai, C., Design of optimally smoothing multistage schemes for the Euler equations, *Comm. Appl. Num. Meth.*, 8 (1992), pp. 761–769.
- [41] Linden, J., Mehrgitterverfahren für das erste Randwertproblem der biharmonischen Gleichung und Anwendung auf ein inkompressibles Strömungsproblem, Ph.D Thesis (in German), Inst. Appl. Math., University of Bonn, Germany (1985).

- [42] Livne, O. and Brandt, A., Local mode analysis of multicolor and composite relaxation schemes, to appear in *Comp. Math. Appl.*
- [43] Llorente, I.M. and Melson, N.D., Behaviour of plane relaxation methods as multigrid smoothers, *Elect. Trans. Numer. Anal.*, 10 (2000), pp. 92–114.
- [44] Mohr, M. and Wienands, R., Cell-centred multigrid revisited, to appear in *Comput. and Visual. in Science*.
- [45] Nabben R., A new application for generalized M-matrices, in: *Numerical Linear Algebra*, Reichel, L., Ruttan, A., and Varga, R.S. (eds.), Walter de Gruyter, New York, 1993, pp. 179–193.
- [46] Niestegge, A. and Witsch, K., Analysis of a multigrid Stokes solver, Arbeitspapiere der GMD 355, St. Augustin, Germany.
- [47] Oosterlee, C.W., A GMRES-based plane smoother in multigrid to solve 3D anisotropic fluid flow problems, *J. Comp. Phys.*, 130 (1997), pp. 41–53.
- [48] Oosterlee, C.W. and Washio, T., Krylov subspace acceleration of non-linear multigrid with application to recirculating flows, *SIAM J. Sci. Comput.*, 21 (2000), pp. 1670–1690.
- [49] Oosterlee, C.W., Gaspar, F.J., Washio, T., and Wienands, R., Multigrid line smoothers for higher order upwind discretizations of convection-dominated problems, *J. Comput. Phys.*, 139 (1998), pp. 274–307.
- [50] Oosterlee, C.W. and Wienands, R., A genetic search for optimal multigrid components within a Fourier analysis setting, *SIAM J. Sci. Comput.*, 24(3) (2002), pp. 924–944.
- [51] Ruge, J. and Stüben, K., Algebraic multigrid (AMG), in: *Multigrid Methods, Frontiers in Applied Mathematics*, McCormick, S.F. (ed.), SIAM, Philadelphia, 1987, pp. 73–130.
- [52] Saad, Y. and Schultz, M.H., GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM J. Sci. Comput.*, 7 (1986), pp. 856–869.
- [53] Saad, Y., *Iterative Methods for Sparse Linear Systems*, PWS Publishing, Boston, 1996.
- [54] Saad, Y., Further analysis of minimum residual iterations, *Numer. Linear Algebra Appl.*, 7 (2000), pp. 67–93.
- [55] Schröder, J. and Trottenberg, U., Reduktionsverfahren für Differenzengleichungen bei Randwertaufgaben I, *Numer. Math.*, 22 (1973), pp. 37–68.

- [56] A. Schüller, Mehrgitterverfahren für Schalenprobleme, Ph.D. Thesis (in German), University of Bonn, GMD-Bericht Nr. 171, Oldenbourg Verlag, Munich, 1988.
- [57] Stevenson, R., On the validity of local mode analysis of multi-grid methods, Ph.D. Thesis, Rijks Univ. Utrecht, The Netherlands, 1990.
- [58] Stüben, K., Algebraic multigrid (AMG): An introduction with applications, Tech. Rep. 53, GMD, St. Augustin, Germany, April 1999. This report appeared as an appendix in [62].
- [59] Stüben, K. and Trottenberg, U., On the construction of fast solvers for elliptic equations, *Computational Fluid Dynamics*, Lecture Series 1982-04, von Karman Inst. for Fluid Dynamics, Rhode-Saint-Genese, Belgium, 1982.
- [60] Stüben, K. and Trottenberg, U., Multigrid methods: fundamental algorithms, model problem analysis and applications, in: *Multigrid Methods, Lecture Notes in Math. 960*, Hackbusch, W. and Trottenberg, U. (eds.), Springer, Berlin, 1982, pp. 1–176.
- [61] Thole, C.A. and Trottenberg, U., Basic smoothing procedures for the multigrid treatment of elliptic 3-D operators, *Appl. Math. Comp.*, 19 (1986), pp. 333–345.
- [62] Trottenberg, U., Oosterlee, C.W., and Schüller, A., *Multigrid*, Academic Press, New York, 2001.
- [63] Varga, R.S., *Matrix Iterative Analysis*, Englewood Cliffs, NJ: Prentice Hall, 1962.
- [64] Wesseling, P., Cell-centered multigrid for interface problems, *J. Comput. Phys.*, 79 (1988), pp. 86–91.
- [65] Wesseling, P., *An Introduction to Multigrid Methods*, John Wiley, Chichester, 1992.
- [66] Wienands, R., Multi-Stage-Jacobi Glätter in Mehrgitterverfahren zur Lösung der inkompressiblen Navier-Stokes-Gleichungen, Master Thesis (in German), University Köln, Germany, 1996.
- [67] Wienands, R., Extended local Fourier analysis for multigrid: Optimal smoothing, coarse grid correction, and preconditioning, Ph.D. Thesis, University of Cologne, Cologne, Germany, 2001.
- [68] Wienands, R. and Oosterlee, C.W. On three-grid Fourier analysis for multigrid, *SIAM J. Sci. Comput.*, 28 (2001), pp. 651–671.
- [69] Wienands, R., Oosterlee, C.W., and Washio, T., Fourier analysis of GMRES(m) preconditioned by multigrid, *SIAM J. Sci. Comput.*, 22 (2000), pp. 582–603.

- [70] Wienands, R., Gaspar, F.J., Lisbona, F.J., and Oosterlee, C.W., An efficient multigrid solver based on distributive smoothing for poroelasticity equations, to appear in *Computing*.
- [71] Wittum, G., Multigrid methods for Stokes and Navier-Stokes equations with transforming smoothers: algorithms and numerical results., *Numer. Math.*, 54 (1989), pp. 543–563.
- [72] Wittum, G., On the robustness of ILU smoothing, *SIAM J. Sci. Stat. Comput.*, 10(4) (1989), pp. 699–717.
- [73] Yavneh, I., Multigrid smoothing factors for red-black Gauss–Seidel relaxation applied to a class of elliptic operators, *SIAM J. Num. Anal.*, 32 (1995), pp. 1126–1138.
- [74] Yavneh, I., On red-black SOR smoothing in multigrid, *SIAM J. Sci. Comput.*, 17 (1996), pp. 180–192.
- [75] Yavneh, I., Coarse-grid correction for nonelliptic and singular perturbation problems, *SIAM J. Sci. Comput.*, 19 (1998), pp. 1682–1699.
- [76] Yavneh, I. and Olvovsky, E., Multigrid smoothing for symmetric nine-point stencils, *Appl. Math. Comp.*, 92 (1998), pp. 229–246.
- [77] de Zeeuw, P.M., Matrix-dependent prolongations and restrictions in a blackbox multigrid solver, *J. Comput. Appl. Math.*, 33 (1990), pp. 1–27.