

Institut für Informatik
der Technischen Universität München

Lehrstuhl für Informatik mit Schwerpunkt
Wissenschaftliches Rechnen

**Multigrid methods for anisotropic and
indefinite structured linear systems of
equations**

Rainer Fischer

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des Akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Hans Michael Gerndt

Prüfer der Dissertation: 1. Univ.-Prof. Dr. Thomas Huckle
2. Univ.-Prof. Dr. Bernd Simeon
3. Prof. Dr. Stefano Serra Capizzano,
Università dell' Insubria Como / Italien
(schriftliche Beurteilung)

Die Dissertation wurde am 10.4.2006 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 17.7.2006 angenommen.

Abstract

Structured linear systems of equations arise in a variety of applications in scientific computing. Image restoration problems and the solution of partial differential equations, which are discussed in this work, are only two examples out of many. We focus on classes of matrices which are related to generating functions: two-level Toeplitz matrices and matrices belonging to a trigonometric algebra, such as two-level circulant, tau, or DCT-III matrices. The main purpose of this thesis is the development of multigrid methods for these structured matrix classes. Generating functions are important for both the design of the methods and the accomplishment of convergence proofs.

In the first chapters of this thesis, we review well-established results on structured matrices, iterative methods, and especially multigrid methods. Special attention is given to the combination of the two fields. We describe several recent developments and add some new results on DST-III matrices and on block systems.

The first main contribution to ongoing research presented in this work is the development of multigrid methods for anisotropic structured linear systems. First, we focus on systems where anisotropy occurs along coordinate axes. The multigrid methods are either based on a suitable combination of semicoarsening and full coarsening steps or on especially designed smoothers. The use of generating functions allows us to present convergence proofs and to carry over some of the results to the more difficult case of linear systems with anisotropy in other directions.

Furthermore, we develop multigrid methods for linear systems corresponding to generating functions with whole zero curves instead of isolated zeros. Typically, these matrices arise when indefinite systems are solved with normal equations. We introduce a Galerkin-based multigrid method and present a convergence proof for the two-grid version. Since matrices become denser on each grid, this method is more a theoretical model than a practical algorithm. However, it is the basis for the development of all subsequent multigrid methods, which are constructed with rediscrretization and splitting techniques.

Eventually, we present the two above-mentioned applications of our multigrid methods. First, we solve image deblurring problems with anisotropic, spatially-invariant point spread functions. After introducing the principal techniques with noise-free examples, we combine multigrid methods with regularization techniques to restore pictures which are both blurred and affected by noise. The second application is the solution of partial differential equations. After briefly discussing anisotropic equations, we focus on the solution of PDEs whose discretization leads to matrices whose generating functions have a zero curve. The most prominent example for this type of PDEs is the Helmholtz equation with constant coefficients.

Zusammenfassung

Strukturierte lineare Gleichungssysteme ergeben sich in einer Vielzahl von Anwendungen im Bereich des Wissenschaftlichen Rechnens. Probleme im Bereich der Bildwiederherstellung sowie das Lösen von partiellen Differentialgleichungen werden in dieser Arbeit diskutiert. Wir konzentrieren uns dabei auf solche Matrixklassen, die eng mit erzeugenden Funktionen zusammen hängen: 2-level Toeplitz-matrizen sowie Matrizen, die zu einer trigonometrischen Algebra gehören, wie z.B. 2-level zirkulante, tau oder DCT-III-Matrizen. Das zentrale Thema dieser Arbeit ist die Entwicklung von Mehrgitterverfahren für diese Klassen von strukturierten Matrizen. Dabei spielen erzeugende Funktionen sowohl bei der Entwicklung der Verfahren als auch bei der Durchführung von Konvergenzbeweisen eine wichtige Rolle.

In den ersten Kapiteln werden bereits bekannte Ergebnisse über strukturierte Matrizen, iterative Verfahren und insbesondere Mehrgitterverfahren wiederholt. Vor allem wird dabei die Verbindung der beiden Themengebiete diskutiert und an einigen Stellen ergänzt, insbesondere für den Fall von DST-III-Matrizen und Blockmatrizen.

Der erste größere Beitrag dieser Arbeit zur aktuellen Forschung ist die Entwicklung von Mehrgitterverfahren für anisotrope strukturierte Systeme. Zuerst werden Methoden für Systeme beschrieben, bei denen die Anisotropie entlang von Koordinatenachsen auftritt. Diese Methoden basieren entweder auf einer geeigneten Kombination aus Semidiskretisierung und voller Diskretisierung oder auf speziell angepassten Glätttern. Mit Hilfe der erzeugenden Funktionen ist es möglich, Konvergenzbeweise zu führen und einige Ergebnisse auf den schwierigeren Fall, in dem Anisotropie in anderen Richtungen auftritt, zu übertragen.

Außerdem werden Mehrgitterverfahren für solche lineare Gleichungssysteme entwickelt, deren erzeugende Funktion anstelle isolierter Nullstellen eine ganze Nullkurve besitzt. Diese Matrizen treten typischerweise beim Lösen von indefiniten Systemen mit Hilfe von Normalengleichungen auf. Wir stellen ein Galerkin-basiertes Mehrgitterverfahren vor und präsentieren dafür einen Zweigitterbeweis. Da die Bandbreite der Matrizen auf größeren Gittern zunimmt, dient dieses Verfahren eher als theoretisches Modell denn als praktischer Algorithmus. Es stellt jedoch die Grundlage für alle folgenden Mehrgitterverfahren dar, die mit Hilfe von Rediskretisierungs- und Splitting-Techniken entwickelt werden.

Schließlich präsentieren wir zwei Anwendungen für unsere Mehrgitterverfahren. Zuerst sollen Probleme der Bildschärfverbesserung mit anisotroper, ortsunabhängiger Point-Spread-Funktion gelöst werden. Zuerst führen wir die wesentlichen Techniken anhand von Beispielen ein, in denen die Bilder nicht durch zusätzliches Rauschen gestört sind. Dann verbinden wir Mehrgitterverfahren mit Regularisierungsmethoden zur Wiederherstellung von Bildern, die sowohl durch Unschärfe als auch durch Rauschen gestört sind. Die zweite Anwendung ist die Lösung von partiellen Differentialgleichungen. Nach einer kurzen Diskussion anisotroper Glei-

chungen liegt der Schwerpunkt auf der Lösung von partiellen Differentialgleichungen, deren Diskretisierung auf erzeugende Funktionen mit Nullkurven führt. Das bedeutendste Beispiel für diesen Typ von Differentialgleichung ist die Helmholtz-Gleichung mit konstanten Koeffizienten.

Contents

1	Introduction	11
1.1	Multigrid methods, structured linear systems, and applications . .	11
1.2	Outline of the thesis	14
2	Structured linear systems of equations and generating functions	17
2.1	Toeplitz and block Toeplitz matrices	18
2.1.1	The correspondence of (multilevel) Toeplitz matrices and generating functions	18
2.1.2	Block Toeplitz matrices	20
2.1.3	Classification of Hermitian positive-definite two-level Toe- plitz matrices	22
2.2	Matrices forming an algebra	22
2.2.1	Circulant and ω -circulant matrices	23
2.2.2	Tau matrices	25
2.2.3	DCT-III and DST-III matrices	26
2.3	Solution of structured linear systems with direct methods	28
2.4	Iterative solution of Toeplitz systems	30
2.4.1	The conjugate gradient method	30
2.4.2	Convergence results and the use of preconditioners	32
2.5	Preconditioners for one-level Toeplitz matrices	33
2.5.1	Algebra preconditioners for well-conditioned Toeplitz matrices	34
2.5.2	Preconditioners for ill-conditioned Toeplitz matrices	35
2.6	Difficulties with preconditioners for multilevel Toeplitz matrices . .	36
2.6.1	Negative results for multilevel matrix algebra preconditioners	36
2.6.2	Multilevel Toeplitz preconditioners	37
3	Multigrid methods for structured linear systems	39
3.1	General multigrid methods	40
3.1.1	The principles of multigrid methods	40
3.1.2	Algebraic multigrid	45
3.1.3	Multigrid as a preconditioner	50
3.2	Exploiting matrix structure for the development of multigrid methods	53

3.2.1	Multigrid in terms of generating functions	53
3.2.2	Criteria for an efficient multigrid method	55
3.3	Multigrid for trigonometric matrix algebras	56
3.3.1	Circulant and tau matrices	56
3.3.2	DCT-III and DST-III algebras	60
3.4	Multigrid for Toeplitz systems	65
3.4.1	A method for zeros of order two	65
3.4.2	Adapting the matrix algebra method to the Toeplitz class	66
3.4.3	Using rediscrretization	67
3.5	Multigrid for block systems	67
3.5.1	A multigrid method for block Toeplitz systems	67
3.5.2	Toeplitz matrices treated as block Toeplitz matrices	69
3.6	Generating functions with multiple zeros	75
3.6.1	Treating each zero separately	75
3.6.2	Using the block approach	77
3.6.3	Multigrid methods with blocks and splitting	78
4	Multigrid methods for anisotropic systems	81
4.1	Anisotropy in terms of generating functions	82
4.1.1	Definition of Anisotropy	82
4.1.2	Problems arising from anisotropic systems	84
4.1.3	How to design multigrid methods for anisotropic systems?	86
4.2	Anisotropy along coordinate axes: Semicoarsening	87
4.2.1	A two-grid method with semicoarsening	87
4.2.2	Extension to a multigrid method	90
4.2.3	Convergence results for trigonometric matrix algebras	91
4.2.4	Convergence results for Toeplitz matrices	94
4.2.5	Numerical results	97
4.3	Anisotropy along coordinate axes: Line Smoothing	99
4.3.1	A multilevel method with line smoothing	99
4.3.2	Convergence results	100
4.3.3	Numerical results	104
4.4	Anisotropy in other directions: Semicoarsening	106
4.4.1	The heuristic: Transformation of the coordinate system	106
4.4.2	The 45° case: Two-grid and multigrid methods	108
4.4.3	Generalization to other directions	111
4.4.4	Convergence results	113
4.4.5	Numerical results	115
4.5	Anisotropy in other directions: Line smoothing	116
4.5.1	Line smoothing and block coarsening	116
4.5.2	Theoretical results	117
4.5.3	Numerical results	118

5	Generating functions with whole zero curves	119
5.1	Galerkin-based multigrid	120
5.1.1	Extending standard multigrid	120
5.1.2	Two-grid convergence results	122
5.1.3	Limitations of the Galerkin approach	124
5.2	Rediscretization and approximation of the zero curve	125
5.2.1	A two-level method based on rediscretization	125
5.2.2	Extending the idea to the multigrid case	128
5.2.3	Numerical examples	129
5.3	A new splitting technique	130
5.3.1	A two-grid method with splitting	130
5.3.2	From two-grid to multigrid	133
5.3.3	Combining splitting with the Galerkin method	134
5.3.4	Numerical results	135
5.4	Anisotropic problems with zero curves	136
5.4.1	Galerkin-based methods	136
5.4.2	Rediscretization and splitting techniques	139
5.4.3	Numerical results	140
5.5	Multigrid as a preconditioner	141
5.5.1	A Galerkin- or rediscretization-based preconditioner	141
5.5.2	Approximation with auxiliary problems	143
5.5.3	The choice of the generating functions	144
5.5.4	Combining the Galerkin approach with auxiliary problems	146
5.5.5	Numerical results	147
6	Applications	149
6.1	Anisotropic image restoration problems	149
6.1.1	The deblurring problem with boundary conditions	150
6.1.2	Multigrid for noise-free anisotropic deblurring	152
6.1.3	Test problems and numerical results	153
6.1.4	Regularization strategies in the presence of noise	156
6.1.5	Numerical tests with Tikhonov and Riley regularization	158
6.2	Partial differential equations	160
6.2.1	Multigrid methods for anisotropic PDEs	160
6.2.2	Multigrid methods for the Helmholtz equation	163
7	Conclusions and future work	169

Chapter 1

Introduction

Partial differential equations, image restoration, signal processing, integral equations - completely different problems in scientific computing lead to very large, and in many cases sparse and structured, linear systems of equations. In the early days of numerical mathematics, direct solvers with a complexity of $O(n^3)$ arithmetic operations were applied to fairly small linear systems. Today, fast iterative methods are used for solving larger and larger systems, especially those which are highly structured. Multigrid methods are among the fastest algorithms, as they make use of both sparsity and structure of the matrices. The work presented in this thesis centers around three topics that interact with each other: multigrid methods, structured linear systems of equations, and their applications.

1.1 Multigrid methods, structured linear systems, and applications

In the year 2000, J. Dongarra and F. Sullivan [43] published a list containing those ten algorithms that had the greatest influence on science and engineering in the last century: the 'Top Ten Algorithms of the 20th century'. This list includes, for example, the simplex method, the quicksort algorithm, Krylov subspace methods, and the Fast Fourier Transform. Surprisingly, it does not mention **multigrid methods**. The latter are widely accepted as being among the fastest numerical methods for the solution of partial differential equations, for integral equations, signal and image processing problems, and many other applications. Therefore, one aim of this thesis is to demonstrate that multigrid methods are equally influential on the field of scientific computing as other methods in the abovementioned top ten list. Compared to other, more classical algorithms, the development and analysis of multigrid methods is a rather young discipline. After the first groundbreaking articles had been published in the late 1970s, the number of research groups working on multigrid methods and the amount of published articles exploded in the 1980s and 1990s. Today, multigrid methods are still a field of ongoing re-

search. The main focus is on more sophisticated applications involving indefinite or convection-dominated problems. Historically, the first multigrid methods were all geometric, i.e. they were based on an underlying physical grid, which was used for a discretization of the problem. Starting in the mid 1980s, researchers became more and more interested in algebraically-oriented multigrid methods. Since these do not require a real grid, they are purely based on the structure and the properties of the matrices. As we are not only interested in partial differential equations, we will mostly follow the algebraic multigrid approach, which is the more general one.

For large classes of linear systems, especially for Hermitian positive-definite ones, multigrid methods are the fastest standalone iterative solvers. For more involved problems, however, they are not as robust as other methods. Nevertheless, multigrid methods are extremely useful for these applications, too: They serve as preconditioners for other iterative solvers. Krylov subspace methods, for example, are not just fast, but also very robust iterative solution techniques. For Hermitian positive definite linear systems, the preconditioned conjugate gradient method is one of the classical iterative solvers. The quality of the method essentially depends on the choice of the preconditioner. If multigrid methods are used as preconditioners, the excellent convergence properties of multigrid can be combined with the robustness of Krylov subspace methods. The result is a method which would truly have deserved a place in the top ten list.

Being iterative solution techniques, multigrid methods essentially demand that matrix-vector products can be computed quickly. Moreover, prolongation and restriction operators as well as coarse grid matrices must be obtained efficiently. Both requirements are met for highly **structured linear systems**. Faster implementations of matrix-vector products are available if the structure is exploited effectively. For example, if matrices are sparse, or if they are tridiagonal, pentadiagonal, etc., matrix-vector products are computed very fast. But what kind of structure do the matrices considered in this work have? All of them either have (multilevel) Toeplitz structure or they belong to a trigonometric matrix algebra such as circulant or tau matrices. Also block variants of these matrices will be considered in this thesis. If matrices with these structures are solved with the help of multigrid methods, the structure of the matrices is used to improve the quality and the efficiency of the algorithms in the following two ways:

- Matrix-vector multiplications involving these structured matrices can be performed significantly faster than with general matrices.
- The matrices correspond to generating functions (often also called symbols), which contain valuable information about the matrices, especially on their eigenvalues. Therefore, essential parts of the multigrid methods are described in terms of generating functions. This is extremely helpful for choosing adequate coarsening operators and for conducting convergence proofs.

The first improvement is quite significant for the matrix classes that are of interest here. For dense matrices, the complexity of a matrix-vector product is reduced from $O(n^2)$ to $O(n \log n)$ arithmetic operations. If, in addition, the matrices are sparse or banded, this complexity is further reduced to $O(n)$ operations. Multigrid methods for Toeplitz-like matrices or for matrices belonging to a trigonometric algebra are a very recent development. The first article on this topic was published by Fiorentino and Serra [52] in 1992, and most of the convergence results have been obtained in the last five years.

Both sparse and dense matrices appear in **applications**, which are the third main topic considered in this thesis. Structured matrices corresponding to generating functions arise in many applications. Typical examples come from the fields of ordinary and partial differential equations, integral equations, Markov chains, signal and image processing, to mention just a few. Several areas of applications are described in the book of Ng [83]. For many of these applications, multigrid methods belong to the fastest solvers. The two areas we are particularly interested in are partial differential equations (PDEs) and image restoration problems.

- Discretization of large classes of PDEs leads to sparse and structured linear systems. Especially if the PDEs have constant coefficients and if discretization is done on regular grids, the resulting matrices belong to a class that is related to generating functions. Although structured linear systems also arise from discretization of parabolic and hyperbolic PDEs, we restrict ourselves to the elliptic case here. Discretization of many boundary value problems with elliptic PDEs results in matrices of Toeplitz or trigonometric algebra type, depending on which kind of boundary condition is used. In this work, we are mainly interested in two types of elliptic problems: on the one hand Poisson-like equations, especially their anisotropic versions, and Helmholtz-related equations on the other.
- The main goal of image restoration is to obtain the original image from an observed image. An observed image can be blurred and, in addition, affected by noise. Typical examples for this type of problem are found in astronomy or in medicine. The blur results for example from looking at an image through clouds or other objects, whereas the noise comes from measurement or transmission errors. If the blur operator is spatially invariant (i.e. if it does not depend on the position within the image), it is represented by a structured matrix. Our aim then is to compute the original image x from the blurred image b by solving the linear system

$$Ax = b + \eta \quad , \quad (1.1.1)$$

where η denotes the noise. By imposing certain boundary conditions, one obtains a blurring matrix which is of two-level Toeplitz, circulant, or DCT-III type. Assuming that there is no noise, these positive definite and ill-conditioned linear systems are solved efficiently with multigrid methods.

However, since the matrices are highly ill-conditioned, the systems are unstable with respect to noise, i.e. perturbations in b . Therefore, several regularization techniques have been proposed in the literature.

1.2 Outline of the thesis

In the following, we wish to give an overview of the main parts of this thesis. Chapter 2 and large parts of Chapter 3 review important, well-established results, whereas Chapters 4 and 5 contain original results obtained in the course of this work. Chapter 6 presents applications for the latter.

Chapter 2 introduces certain classes of structured matrices which are related to generating functions. These include Toeplitz, multilevel Toeplitz, and block Toeplitz matrices on the one hand, as well as matrices belonging to trigonometric matrix algebras such as circulant and tau matrices on the other. We will emphasize the strong connection, which will be used throughout this work, between generating functions and matrices. Whereas matrices of a trigonometric algebra are efficiently solved with direct methods, this is not true for Toeplitz matrices. Therefore, we present the conjugate gradient method as an example of an iterative solver which is highly efficient if a suitable preconditioner is used. In the last two sections of this chapter, we discuss preconditioners for certain classes of Hermitian positive definite (multilevel) Toeplitz systems. It becomes clear that, especially for ill-conditioned multilevel Toeplitz systems, there are hardly any efficient preconditioners. These limitations of the preconditioned conjugate gradient method leave plenty of room for the development of multigrid methods for structured matrices. This will be the central topic in the remainder of this thesis.

Chapter 3 gives an introduction to multigrid methods, particularly to multigrid methods for structured linear systems. We start with a rather general description of the main multigrid components, smoothing and coarse grid correction, and of the structure of the iterations. Special emphasis is given to algebraic multigrid and on the convergence theory of Ruge and Stüben. Moreover, we describe the use of multigrid methods as a preconditioner for other iterative solvers. Since our main goal is to develop multigrid methods for structured matrices and generating functions, we describe how matrix structure can be exploited. Moreover, we discuss general prolongation and restriction operators in terms of generating functions as well as criteria that should be fulfilled by a good multigrid algorithm. In the next two sections, we review well-established results on multigrid methods for matrices of a trigonometric algebra and their extension to Toeplitz problems. The results for the DST-III algebra are an original contribution of this work. These methods are restricted to generating functions with single isolated zeros. The two final sections contain new results on multigrid methods for block Toeplitz systems and their use for linear systems whose generating functions have multiple zeros.

Chapter 4 presents multigrid methods for anisotropic linear systems from the

point of view of structured matrices and generating functions. We start with a description of what anisotropy means in the context of generating functions and why the multigrid methods from Chapter 3 lose their efficiency when applied to anisotropic systems. In terms of generating functions, we divide anisotropic problems into two classes, depending on whether anisotropy occurs along coordinate axes or in other directions. For each class, we develop multigrid methods based both on a combination of semicoarsening steps and full coarsening steps, and multigrid methods based on more sophisticated smoothers. Multigrid methods for the first class of problems have been developed in the context of partial differential equations. Here, however, we present them in terms of generating functions and their level curves. On the one hand, this allows us to perform convergence proofs for two-grid methods, and for multigrid methods using W-cycles. On the other hand, we can carry over most of the results for Toeplitz and circulant matrices to the second class of problems, which are more difficult to solve. After a coordinate transformation, which corresponds to a permutation of the rows and columns of the matrices, we design multigrid methods either with a semicoarsening strategy or with specialized smoothers. Due to the use of generating functions, we also retain theoretical results in this more general context.

Chapter 5 is devoted to multigrid methods for positive-definite linear systems whose generating functions have a whole curve of zeros instead of isolated zeros. These matrices appear, for example, when indefinite systems are solved with normal equations. We start with an extension of the Galerkin-based multigrid methods from Chapter 3 for generating functions with a zero curve. Although optimal convergence can be proved, this approach is computationally too expensive. Therefore, we present a multigrid method based on rediscrretization, where the zero curve is only approximated on coarser grids. Since this method preserves bandedness of the matrices on all levels, it is computationally acceptable. The only disadvantage is that the number of grids is limited, because zero curves become significantly larger on coarser grids. Therefore, we devise a splitting technique which divides the original problem into a fixed number k of subproblems on coarser grids, each of them corresponding to a generating function with isolated zeros. The splitting technique is then combined with the Galerkin approach, which means that the Galerkin-based coarsening is used on the finest levels and splitting only on coarser levels. In the next section, we discuss multigrid methods for anisotropic linear systems whose generating functions have a whole curve of zeros. These methods are constructed by combining the techniques from this chapter with the multigrid methods developed in Chapter 4. Eventually, we construct multigrid preconditioners, using a slightly different splitting technique. It is based on an approximation of the zero curve by k auxiliary problems with isolated zeros.

Chapter 6 contains examples for applications of the multigrid methods presented in Chapters 4 and 5. The first application considered in this work are image restoration problems, where the blur is of anisotropic nature. Under the assumption that the blurred images are not contaminated with any noise, the

multigrid methods from Chapter 4 are highly efficient for the restoration of such pictures. In the presence of noise, regularization is necessary. After discussing several strategies, we apply either the Tikhonov or the Riley regularization and solve the resulting linear systems with multigrid methods.

The second application are partial differential equations. Discretization of anisotropic PDEs of Poisson type with constant coefficients results in anisotropic linear systems which can be analyzed with the methods from Chapter 4. Depending on the choice of boundary conditions, one obtains matrices belonging to the Toeplitz class or to one of the trigonometric matrix algebras. When indefinite PDEs of Helmholtz type are discretized and then solved with normal equations, one obtains linear systems, the generating functions of which have a whole zero curve. Again, the boundary conditions determine the matrix class. Both the rediscrretization-based multigrid method from Section 5.2 and the splitting technique from Section 5.3 are used to solve these problems.

Chapter 7 concludes this thesis by summarizing the main results which have been obtained in the course of this work. Furthermore, we outline a number of possible extensions of our results, which could be subject of further research.

The main results of Chapter 4 are contained in the articles [55] and [56], whereas the results of Sections 5.1 and 5.2 are summarized in [57]. The rest of Chapter 5 as well as Section 6.2 will appear in an upcoming paper.

Chapter 2

Structured linear systems of equations and generating functions

This chapter gives an overview of the classes of structured matrices which are relevant for our work. All these matrices are connected to generating functions. Throughout this thesis, generating functions will be used for the development of iterative methods, especially of multigrid methods. The linear systems we are most interested in can be divided into two different classes: multilevel Toeplitz matrices and block Toeplitz matrices on the one hand, and matrices belonging to a trigonometric algebra such as multilevel circulant or τ matrices on the other. Matrices from these algebras are not only challenging problems in themselves, they are also important as preconditioners for Toeplitz systems. Moreover, the development of multigrid methods for Toeplitz systems in Chapter 3 will be based on multigrid methods for matrix algebras.

At the beginning of this chapter, we introduce both types of matrices and emphasize their strong connection to generating functions. Since in this work, we mainly focus on Hermitian positive-definite matrices, these are subdivided into several categories, depending on the corresponding generating functions. For Toeplitz matrices, the applicability of direct solution methods is rather limited. Therefore, we describe the preconditioned conjugate gradient method as an example of an efficient iterative solver. In the last two sections of this chapter, we summarize the main results which were obtained on preconditioners for Hermitian positive-definite Toeplitz systems in the last 15 years. In particular, we describe the limitations of the pcg method for ill-conditioned multilevel Toeplitz matrices. These results are the starting point for the development of multigrid methods for structured linear systems, both as standalone solvers and as preconditioners for the cg method.

2.1 Toeplitz and block Toeplitz matrices

This section introduces the class of Toeplitz matrices, which is one of the classes we are primarily interested in. We start with a description of one- and multilevel Toeplitz matrices in the context of generating functions. Since we mainly focus on two-level matrices, we consider the general multilevel case only briefly. In the second part of this section, we introduce block Toeplitz matrices, which will be helpful for the construction of certain multigrid preconditioners and solvers. In the last part, we give a more detailed description of those Toeplitz matrices we are mostly looking into: Hermitian positive-definite two-level Toeplitz matrices. These can be divided into different classes, depending on the generating function. The description of Toeplitz and multilevel Toeplitz matrices is, to a large extent, based on the presentation in [62, 27, 83], the one of block Toeplitz matrices on the presentation in [96, 79, 71].

2.1.1 The correspondence of (multilevel) Toeplitz matrices and generating functions

A matrix $T_n \in \mathbb{C}^{n \times n}$ is called Toeplitz if it is constant along its diagonals, i.e. if it is of the form

$$T_n = \begin{pmatrix} t_0 & t_{-1} & \cdots & t_{2-n} & t_{1-n} \\ t_1 & t_0 & t_{-1} & & t_{2-n} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ t_{n-2} & & t_1 & t_0 & t_{-1} \\ t_{n-1} & t_{n-2} & \cdots & t_1 & t_0 \end{pmatrix}. \quad (2.1.1)$$

T_n can be interpreted as the n -by- n principal submatrix of a singly-infinite matrix T_∞ , whose entries are given by $T_{l,m} = t_{l-m}$. This matrix is connected to the generating function

$$f(x) = \sum_{k=-\infty}^{\infty} t_k e^{-ikx} \quad . \quad (2.1.2)$$

On the other hand, we can define, for any function $f \in L^2[-\pi, \pi]$, the sequence of Toeplitz matrices $(T_n[f])_{n \in \mathbb{N}}$, whose coefficients are the Fourier coefficients

$$t_k = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) e^{-ikx} dx \quad (k \in \mathbb{Z}). \quad (2.1.3)$$

of f . There is a strong connection between a generating function f and the matrices of the sequence $(T_n[f])_{n \in \mathbb{N}}$.

Remark 1 The following properties of a Toeplitz matrix $T_n[f]$ are derived from the properties of the generating function f .

- If f is a trigonometric polynomial $f(x) = \sum_{k=-p}^q t_k e^{-ikx}$, then $T_n[f]$ is a band matrix with bandwidth $\leq p + q + 1$.
- If f is real-valued, then $T_n[f]$ is Hermitian.
- If, in addition, f is even, then $T_n[f]$ is real symmetric. This means that f can be written as a cosine series.
- Toeplitz matrices do not form an algebra. For example, the product of two Toeplitz matrices is not necessarily Toeplitz. However, they have the following properties:

$$T_n[1] = I_n, \quad T_n[\lambda f] = \lambda T_n[f] \quad (\lambda \in \mathbb{C}), \quad T_n[f + g] = T_n[f] + T_n[g] .$$

The most important relationship between generating function and Toeplitz matrices concerns the spectra of the matrices. The following Theorem summarizes the main results given by Grenander and Szegö on the bounds and the distribution of the eigenvalues of $T_n[f]$.

Theorem 1 (Grenander and Szegö, [62])

Let f be an integrable, real-valued function and $(T_n[f])_n$ the sequence of Toeplitz matrices generated by f . Let f_{\min} and f_{\max} denote the essential infimum and essential supremum of f , respectively.

1. Then, for each $n \geq 1$, the eigenvalues $\lambda_j^{(n)}$ ($0 \leq j \leq n-1$) of $T_n[f]$ have the following properties:
 - $\lambda_k \in [f_{\min}, f_{\max}]$ ($1 \leq k \leq n$)
 - If $f_{\max} > f_{\min}$, then $f_{\min} < \lambda_{\min}(T_n[f]) \leq \lambda_{\max}(T_n[f]) < f_{\max}$
 - Furthermore, for $n \rightarrow \infty$, the extreme eigenvalues of $T_n[f]$ tend to f_{\min} and f_{\max} , i.e.

$$\lim_{n \rightarrow \infty} \lambda_{\min}^{(n)} = f_{\min} \quad \text{and} \quad \lim_{n \rightarrow \infty} \lambda_{\max}^{(n)} = f_{\max} . \quad (2.1.4)$$

2. $\lambda_j^{(n)}$ are equally distributed as $f(\frac{2\pi j}{n})$, i.e.

$$\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{j=0}^{n-1} \left[g(\lambda_j^{(n)}) - g\left(f\left(\frac{2\pi j}{n}\right)\right) \right] = 0 \quad (2.1.5)$$

for any continuous function g defined on $[-\pi, \pi]$.

The first part of this theorem implies that the matrices $T_n[f]$ are Hermitian positive-definite and ill-conditioned if f is nonnegative and has zeros. A more detailed description of this fact will be given for two-level Toeplitz matrices in Chapter 2.1.3. The second part of Theorem 1 implies that for large n one can hardly distinguish between the graph of f and a plot of the eigenvalues.

So far, we have described one-level Toeplitz matrices. However, all the results hold for multilevel Toeplitz matrices as well. A two-level Toeplitz matrix of size $\mathbf{n} = n_1 \cdot n_2$ is a block Toeplitz matrix of the form

$$T_{\mathbf{n}} = \begin{pmatrix} T_0 & T_{-1} & \cdots & T_{2-n_1} & T_{1-n_1} \\ T_1 & T_0 & T_{-1} & & T_{2-n_1} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ T_{n_1-2} & & T_1 & T_0 & T_{-1} \\ T_{n_1-1} & T_{n_1-2} & \cdots & T_1 & T_0 \end{pmatrix}, \quad (2.1.6)$$

where each block T_j is itself a Toeplitz matrix of size n_2 . Two-level Toeplitz matrices are also called block-Toeplitz-Toeplitz-block (BTTB) matrices. They correspond to generating functions in two variables. (2.1.2) and (2.1.3) are replaced by

$$f(x, y) = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} t_{k,l} e^{-i(kx+ly)} \quad (2.1.7)$$

and

$$t_{k,l} = \frac{1}{(2\pi)^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} f(x, y) e^{-i(kx+ly)} dx dy \quad (k, l \in \mathbb{Z}). \quad (2.1.8)$$

A p -level Toeplitz matrix has Toeplitz structure on each level and corresponds to a p -variate generating function. It is a matrix of size $\mathbf{n} = n_1 \cdot \cdots \cdot n_p$.

Remark 2 In the multilevel case, the results of Remark 1 and Theorem 1 hold unchanged. Most importantly, the eigenvalues are bounded by f_{min} and f_{max} , and for large \mathbf{n} the minimum and maximum eigenvalues of $T_{\mathbf{n}}$ tend to f_{min} and f_{max} .

2.1.2 Block Toeplitz matrices

In a BTTB matrix $T_{\mathbf{n}}[f]$, the number of blocks n_1 and the size of the blocks n_2 are variable, because a scalar generating function $f(x, y)$ corresponds to a whole sequence of matrices $(T_{n_1 n_2}[f])_{n_1, n_2 \in \mathbb{N}}$. Now, we introduce another class of matrices which will be useful for the development of multigrid methods: the so called block Toeplitz matrices, which are for example described in [93, 96, 79]. They are connected to matrix-valued generating functions $F : [-\pi, \pi] \rightarrow \mathbb{C}^{k \times k}$. Thus, $F(x)$ defines a sequence of matrices $(T_{n_1 \cdot k}[F])_{n_1}$, which are of the form (2.1.6) on the block level. The individual blocks, however, do not necessarily have structure,

but k is a fixed and usually small number. In the following, we assume that the matrix $F(x)$ is Hermitian. The block T_j is considered to be the Fourier coefficient

$$T_j = \frac{1}{2\pi} \int_{-\pi}^{\pi} F(x) e^{-ikx} dx \quad (j \in \mathbb{Z}) \quad (2.1.9)$$

of the function $F(x)$, which is assumed to be L_2 -integrable on $[-\pi, \pi]$. (2.1.9) implies that $T_{n_1 k}[F]$ is Hermitian if the k -by- k matrix $F(x)$ is Hermitian.

The main results on the eigenvalues of $T_{n_1 k}[F]$ are extensions of the two parts of Theorem 1 for the block case ($k > 1$). They are summarized in the following two theorems, which can be found in [96, 79]. The first of them gives upper and lower bounds for the eigenvalues of $T_{n_1 k}[F]$.

Theorem 2

Let $f : [-\pi, \pi] \rightarrow \mathbb{C}^{k \times k}$ be an Hermitian function in $L_2([-\pi, \pi])$ with eigenvalue functions $\lambda_j(x)$. Let $(T_{n_1 k}[F])_{n_1}$ denote the corresponding sequence of block Toeplitz matrices, where the matrix $T_{n_1 k}[F]$ has eigenvalues $\lambda_j^{(n_1, k)}$ ($1 \leq j \leq n_1 \cdot k$). Let m_f and M_f be defined by

$$m_f = \operatorname{ess\,inf}_{x \in [-\pi, \pi]} \min_{j=1, \dots, k} (\lambda_j(f(x))) \quad , \quad m_f = \operatorname{ess\,sup}_{x \in [-\pi, \pi]} \max_{j=1, \dots, k} (\lambda_j(f(x))) \quad .$$

Then the following holds:

1. The $\lambda_j^{(n_1, k)}$ lie in the interval $[m_f, M_f]$.
2. If $\min_{j=1, \dots, k} \lambda_j(x)$ is not essentially constant in x , then all $\lambda_j^{(n_1, k)}$ lie in $(m_f, M_f]$.
3. If $\max_{j=1, \dots, k} \lambda_j(x)$ is not essentially constant in x , then all $\lambda_j^{(n_1, k)}$ lie in $[m_f, M_f)$.

The second theorem leads to an equal distribution of the eigenvalues.

Theorem 3

Let $f(x)$, $(T_{n_1 k}[F])_{n_1}$, $\lambda_j(x)$, and $\lambda_j^{(n_1, k)}$ be defined as in Theorem 2. Then the $\lambda_j^{(n_1, k)}$ are equally distributed as the $\lambda_j(x)$, i.e. for any continuous function $G(x)$ on \mathbb{R} with bounded support, we obtain

$$\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{j=1}^{n_1 k} G(\lambda_j^{(n_1, k)}) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \sum_{j=1}^k G(\lambda_j(F(x))) dx \quad . \quad (2.1.10)$$

Furthermore, we wish to introduce multilevel block Toeplitz matrices. These have multilevel Toeplitz structure on the block level with unstructured blocks of fixed size k . The results of Theorems 2 and 3 hold unchanged for these matrices. In this work, we are mainly interested in two-level block Toeplitz matrices, i.e. matrices which have BTTB structure on the block level and blocks of fixed size k .

2.1.3 Classification of Hermitian positive-definite two-level Toeplitz matrices

In Section 2.1.1, we have described the strong connection between (multilevel) Toeplitz matrices and generating functions. Theorem 1 implies that a change of sign in the function leads to indefinite matrices. There has been research on indefinite Toeplitz systems [53], but it is extremely difficult to find preconditioners which are suitable for large classes of indefinite matrices. In this work, we are mainly concerned with the solution of Hermitian positive-definite linear systems, whose underlying generating function is nonnegative. Nevertheless, there is still a huge variety of matrices that satisfy this condition. We now wish to achieve a better understanding of the different types of two-level Toeplitz systems we have to deal with in this thesis. Therefore, we divide these matrices into three different classes, depending on the zeros of their generating functions:

- (I) f is strictly positive, i.e. there exists a constant c with $f(x, y) \geq c > 0$. Due to Theorem 1 the corresponding Toeplitz matrices have a smallest eigenvalue $\lambda_{min} > c$. Therefore, this class of linear systems is the only one where matrices are well-conditioned.
- (II) f has a single isolated zero of finite order in $] -\pi, \pi]^2$. Since f is nonnegative, the order of the zero is even. By Theorem 1, the minimum eigenvalue of the Toeplitz matrices tends to zero for large n . Moreover, Serra [94] has proved that the condition number of $T_n[f]$ grows like $O(n^{2\nu})$ if the zero is of order 2ν .
- (III) f has several isolated zeros of finite order in $] -\pi, \pi]^2$.
- (IV) f has a whole zero curve in $] -\pi, \pi]^2$, i.e. infinitely many zeros which form a smooth curve. The design of multigrid methods will turn out to be considerably more difficult than for problems of class (II) or (III).

We do not solve other types of Hermitian positive definite BTTB systems here. These include systems corresponding to functions with zeros of infinite order or functions which are zero on a subset of $] -\pi, \pi]^2$ with Lebesgue measure > 0 .

2.2 Matrices forming an algebra

In addition to Toeplitz systems, we focus on other types of structured matrices, which are central to the course of this work. These include circulant and ω -circulant matrices, tau matrices, DCT-III and DST-III matrices. Each of these matrix types forms an algebra. This means that its members have more common structure than Toeplitz matrices, and that the corresponding linear systems are easier to solve. Therefore, they are not only useful as preconditioners for Toeplitz

systems. It will also be effective to develop multigrid methods first for these algebras of structured matrices, and extend them to the Toeplitz case later.

Matrices belonging to trigonometric algebras can be characterized and described from three different points of view, which all contribute to a better understanding of the matrix classes. The *structural* and *functional* characterizations are similar to the ones for the Toeplitz class, the *algebraic* characterization is new for matrix algebras. The structural description is based on the pattern of the matrices, whereas the functional characterization relies on the correspondence between matrices and generating functions. An algebraic description for Toeplitz matrices has only been possible in a very limited way. For the matrix algebras we are interested in, it is based on a diagonalization of the matrices by fast trigonometric transforms. We will use aspects of all three characterizations to describe the matrix algebras.

2.2.1 Circulant and ω -circulant matrices

Circulant matrices are described in much detail in the book by Davis [36] and in the articles [27, 83]. They arise for example when periodic boundary conditions are used in image processing or with elliptic partial differential equations. Furthermore, they are important as preconditioners for Toeplitz systems. From a structural point of view, every circulant matrix is a special Toeplitz matrix where the first element of each row is equal to the last element of its preceding row. In other words, it is of the form

$$C_n = \begin{pmatrix} c_0 & c_{n-1} & \cdots & c_2 & c_1 \\ c_1 & c_0 & c_{n-1} & & c_2 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ c_{n-2} & & c_1 & c_0 & c_{n-1} \\ c_{n-1} & c_{n-2} & \cdots & c_1 & c_0 \end{pmatrix} . \quad (2.2.1)$$

From an algebraic point of view, the circulant class contains all matrices which are diagonalized by the Discrete Fourier Transform (DFT), i.e. which have a decomposition of the form

$$C_n = (Q_n^{(circ)})^H \Lambda_n Q_n^{(circ)} . \quad (2.2.2)$$

Λ_n is the diagonal matrix containing the eigenvalues of C_n , and $Q_n^{(circ)}$ is the Fourier matrix, which is a unitary matrix with entries

$$[Q_n^{(circ)}]_{j,k} = \frac{1}{\sqrt{n}} e^{2\pi i j k / n} \quad (0 \leq j, k \leq n-1) . \quad (2.2.3)$$

The columns of $Q_n^{(circ)}$ form an eigenvector basis for all circulant matrices of size n . A direct consequence of the decomposition (2.2.2) is that the class of circulant

matrices is an algebra and, most importantly, that the product of two circulant matrices is again circulant. Furthermore, matrix-vector products involving $Q_n^{(circ)}$ and $(Q_n^{(circ)})^H$ can be computed in $O(n \log n)$ arithmetic operations with the FFT. Since linear systems involving circulant matrices are solved with three FFTs, this also requires only $O(n \log n)$ operations.

The functional characterization of circulant matrices can be given in two different ways. The first of them is similar to the functional characterization of Toeplitz matrices.

- For a given function $f(x)$, we can define a sequence of circulant matrices $(C_n[f])_{n \in \mathbb{N}}$ using the Fourier coefficients of f . The coefficients from $c_{-n/2}$ to $c_{n/2}$ are the matrix entries in the central n diagonals. The outer diagonals are filled such that the matrix has circulant structure. In other words, the circulant matrix of size n corresponding to e^{ix} is

$$Z_n = \begin{pmatrix} 0 & \cdots & & & & 1 \\ 1 & 0 & & & & \\ 0 & 1 & 0 & & & \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ & & 0 & 1 & 0 & \\ & & \cdots & 0 & 1 & 0 \end{pmatrix}, \quad (2.2.4)$$

and all other circulant matrices of size n can be interpreted as polynomials in Z_n , whose coefficients are the Fourier coefficients $c_{-n/2}$ to $c_{n/2}$ of f . Like Toeplitz matrices, circulant matrices are Hermitian if f is real-valued, and symmetric if f is even. Mostly, we are interested in sparse matrices, corresponding to generating functions which are polynomials with only a small number of nonzero coefficients.

- The second functional description of circulant matrices is obtained from the decomposition (2.2.2). It can be shown that the eigenvalues in Λ_n are a sampling of the function $f(x)$ over the points $x_j = \frac{2j\pi}{n}$ ($j = 0, \dots, n-1$). For example, the elementary matrix Z_n from (2.2.4) is obtained from Λ with $\lambda_j = e^{2ij\pi/n}$. One consequence of this functional characterization becomes important when we consider ill-conditioned matrices. If f is zero at one of the sampling points, then one of the eigenvalues of $C_n[f]$ is zero, and the matrix is singular.

ω -circulant matrices form a slightly more general matrix algebra, which includes circulant matrices as a subalgebra. ω -circulant matrices are described for example in [27, 83]. From a structural point of view, an ω -circulant matrix is a Toeplitz matrix where the first element of each row is obtained by multiplying the last element of the preceding row by $e^{i\theta}$ with $\theta \in [-\pi, \pi]$. Algebraically, an ω -circulant matrix W_n of size n is defined by the decomposition

$$W_n = \Omega_n C_n \Omega_n^H = \Omega_n ((Q_n^{(circ)})^H \Lambda_n Q_n^{(circ)}) \Omega_n^H. \quad (2.2.5)$$

Λ_n contains the eigenvalues of W_n , $Q_n^{(circ)}$ is the Fourier matrix, and $\Omega_n = \text{diag}(1, \omega^{1/n}, \dots, \omega^{(n-1)/n})$ with $\omega = e^{i\theta}$. Of course, matrix vector products involving ω -circulant matrices and the solution of linear systems with an ω -circulant matrix can be computed in $O(n \log n)$ operations.

Multilevel versions of circulant and ω -circulant matrices are obtained by forming the Kronecker product of one-level matrices. Multilevel circulant matrices correspond to multivariate generating functions. In this thesis, we are primarily interested in two-level algebras. In the circulant case, these are called block-circulant-circulant-block (BCCB) matrices. For the diagonalization of two-level matrices we only need to change the subscripts in (2.2.2) and (2.2.5). A BCCB matrix $C_{\mathbf{n}}$ is diagonalized with the two-level Fourier matrix $Q_{\mathbf{n}}^{(circ)} = Q_{n_1}^{(circ)} \otimes Q_{n_2}^{(circ)}$. For two-level ω -circulant matrices, we use, in addition to $F_{\mathbf{n}}$, the matrix $\Omega_{\mathbf{n}} = \Omega_{n_1} \otimes \Omega_{n_2}$, where the parameters ω_1 and ω_2 can be chosen with different angles θ_1 and θ_2 . Again, the eigenvalues in $\Lambda_{\mathbf{n}}$ are obtained as a sampling of $f(x, y)$ over the points $(x_j, y_k) = (\frac{2ij\pi}{n_1}, \frac{2ik\pi}{n_2})$. Again, this leads to a singular matrix if f is zero at one of the points.

2.2.2 Tau matrices

Another matrix algebra related to fast trigonometric transforms is the class of tau matrices, which are described for example in [9, 6, 51]. They are not only important as preconditioners for Toeplitz systems. The development of multigrid methods for Toeplitz matrices is essentially based on the results obtained for tau matrices. Algebraically, the class of tau matrices is obtained by applying the Discrete Sine Transform I (DST-I) to real diagonal matrices. A tau matrix τ_n is defined as the product

$$\tau_n = (Q_n^{(tau)})^H \Lambda_n Q_n^{(tau)} \quad . \quad (2.2.6)$$

Λ_n is the diagonal matrix containing the eigenvalues of τ_n , and $Q_n^{(tau)}$ is the orthogonal and symmetric DST-I matrix, which has the entries

$$[Q_n^{(tau)}]_{j,k} = \frac{2}{\sqrt{n+1}} \sin\left(\frac{\pi jk}{n+1}\right) \quad (1 \leq j, k \leq n) \quad . \quad (2.2.7)$$

Again, the columns of $Q_n^{(tau)}$ are an eigenvector basis for every tau matrix of size n . Like the FFT, the DST-I has a fast implementation which requires only $O(n \log n)$ operations. Therefore, computation of matrix-vector products and the solution of a linear system can be computed in $O(n \log n)$.

The structure of a tau matrix is closely related to a Toeplitz matrix, especially if the matrix is sparse. A tau matrix can be expressed as the difference between a symmetric Toeplitz matrix and a persymmetric Hankel matrix. Since we wish to give a more detailed description of this fact, we also need to consider the functional point of view. As tau matrices are real symmetric, generating functions are

represented as a cosine series

$$f(x) = \sum_{k=0}^{\infty} t_k \cos(kx) . \quad (2.2.8)$$

For a given generating function $f(x)$, whose Fourier coefficients are known, the tau matrix of size n is defined

$$\tau_n[f] = \begin{pmatrix} t_0 & t_1 & \cdots & t_{n-1} \\ t_1 & t_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & t_1 \\ t_{n-1} & \cdots & t_1 & t_0 \end{pmatrix} - \begin{pmatrix} t_2 & \cdots & t_{n-1} & 0 & 0 \\ \vdots & \ddots & & & 0 \\ t_{n-1} & & 0 & & t_{n-1} \\ 0 & & & \ddots & \vdots \\ 0 & 0 & t_{n-1} & \cdots & t_2 \end{pmatrix} . \quad (2.2.9)$$

Therefore, a tridiagonal τ matrix is also Toeplitz and vice versa. Another way of expressing the correspondence between matrices and generating functions is to relate the matrix

$$Y_n = \begin{pmatrix} 0 & 1 & & & \\ 1 & \ddots & \ddots & & \\ & \ddots & \ddots & 1 & \\ & & & 1 & 0 \end{pmatrix} \quad (2.2.10)$$

to the function $f(x) = 2 \cos(x)$. Then, every cosine polynomial can be expressed by a sum of powers of Y_n .

The second functional description is similar to the circulant case. The eigenvalues of a matrix $\tau_n[f]$, which appear in $\Lambda_n[f]$, are again a sampling of the generating function. Only the choice of sampling points $x_j = \frac{j\pi}{n+1}$ ($j = 1, \dots, n$) is slightly different. The elementary tau matrix Y_n from (2.2.10) has the eigenvalues $\lambda_j = 2 \cos \frac{j\pi}{n+1}$.

Like in the circulant algebra, multilevel tau matrices are constructed by forming the Kronecker product of one-level tau matrices. For the diagonalization of a two-level matrix, we use $Q_{\mathbf{n}}^{(tau)} = Q_{n_1}^{(tau)} \otimes Q_{n_2}^{(tau)}$ in (2.2.6). From a functional point of view, the eigenvalues in $\Lambda_{\mathbf{n}}$ are obtained by sampling $f(x, y)$ over the points $(x_j, y_k) = (\frac{j\pi}{n_1+1}, \frac{k\pi}{n_2+1})$ ($j, k \in \{1, \dots, n\}$).

2.2.3 DCT-III and DST-III matrices

Finally, we wish to introduce two more trigonometric matrix algebras, the ones related to the DCT-III and DST-III transforms. Matrices from these algebras are constructed when certain boundary conditions are used with partial differential equations or in image restoration applications. Matrices of the former algebra are diagonalized by a discrete cosine transform, more specifically by the DCT-III transform. This transform, which is described for example in [23, 28], also has a

fast implementation based on the FFT. The algebraic description of this matrix class is given by the diagonalization

$$R_n = Q_n^{(dct3)} \Lambda_n (Q_n^{(dct3)})^H \quad . \quad (2.2.11)$$

Λ_n is the diagonal matrix containing the eigenvalues of R_n , and $Q_n^{(dct3)}$ is the orthogonal DCT-III matrix, which has the entries

$$[Q_n^{(dct3)}]_{j,k} = \sqrt{\frac{2 - \delta_{k,1}}{n}} \cos\left(\frac{\pi(k-1)(2j-1)}{2n}\right) \quad (1 \leq j, k \leq n) \quad , \quad (2.2.12)$$

where $\delta_{1,1} = 1$ and $\delta_{k,1} = 0$ if $k \neq 1$. Again, the columns of $Q_n^{(dct3)}$ are an eigenvector basis for every DCT-III matrix of size n . From a structural point of view, a DCT-III matrix can be written as the sum of a Toeplitz matrix and a Hankel matrix. With the first functional description, this can be expressed in more detail. Since we are only interested in real symmetric DCT-III matrices, the generating function $f(x)$ is a cosine series of the form (2.2.8). The corresponding DCT-III matrix $R_n[f]$ of size n is written as the sum

$$R_n[f] = \begin{pmatrix} t_0 & t_1 & \cdots & t_{n-1} \\ t_1 & t_0 & & \vdots \\ \vdots & \ddots & \ddots & t_1 \\ t_{n-1} & \cdots & t_1 & t_0 \end{pmatrix} + \begin{pmatrix} t_1 & t_2 & \cdots & t_{n-1} & 0 \\ t_2 & & \ddots & 0 & t_{n-1} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ t_{n-1} & 0 & \ddots & \ddots & t_2 \\ 0 & t_{n-1} & \cdots & t_2 & t_1 \end{pmatrix} \quad . \quad (2.2.13)$$

The second functional point of view is again related to the eigenvalues of the matrix $R_n[f]$, which appear in $\Lambda_n[f]$. This time they are a sampling of the generating function at the points $x_j = \frac{j\pi}{n}$ ($j = 0, \dots, n-1$). As for circulant and tau matrices, there is also an elementary matrix W_n which generates the DCT-III algebra of size n , and which belongs to the generating function $f(x) = 2 \cos(x)$:

$$W_n = \begin{pmatrix} 1 & 1 & & & \\ 1 & 0 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & 0 & 1 \\ & & & 1 & 1 \end{pmatrix} \quad . \quad (2.2.14)$$

The last trigonometric algebra we want to present in this thesis contains the matrices obtained with the DST-III transform, which is a slightly different discrete sine transform. A matrix of this class

$$S_n = (Q_n^{(dst3)})^H \Lambda_n Q_n^{(dst3)} \quad (2.2.15)$$

is obtained by orthogonal transformation with $Q_n^{(dst3)}$ from the diagonal matrix Λ_n containing the eigenvalues of S_n . $Q_n^{(dst3)}$ is the orthogonal DST-II matrix, which has the entries

$$[Q_n^{(dst3)}]_{j,k} = \sqrt{\frac{2 - \delta_{k,n}}{n}} \cos\left(\frac{\pi(j-1)(2k-1)}{2n}\right) \quad (1 \leq j, k \leq n) \quad , \quad (2.2.16)$$

where $\delta_{n,n} = 1$ and $\delta_{k,n} = 0$ if $k \neq n$. Again, the columns of $Q_n^{(dst3)}$ are an eigenvector basis for every DST-III matrix of size n . From a structural point of view, the matrices in this algebra are very similar to the ones belonging to the DCT-III algebra. In (2.2.13), one has to replace the plus sign by a minus sign to obtain the matrix S_n as a Toeplitz-minus-Hankel matrix. The eigenvalues of $S_n[f]$ are obtained by sampling the generating function $f(x)$ over the points $x_j = \frac{j\pi}{n}$ ($j = 1, \dots, n$). The elementary matrix generating the DST-III algebra and corresponding to $f(x) = 2 \cos(x)$ is of the form

$$V_n = \begin{pmatrix} -1 & 1 & & & \\ 1 & 0 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & 0 & 1 \\ & & & 1 & -1 \end{pmatrix} . \quad (2.2.17)$$

Again, multilevel DCT-III and DST-III matrices are constructed by forming the Kronecker product of the corresponding one-level matrices. For the diagonalization of a two-level matrix, we use $Q_{\mathbf{n}}^{(dct3)} = Q_{n_1}^{(dct3)} \otimes Q_{n_2}^{(dct3)}$ in (2.2.11) and $Q_{\mathbf{n}}^{(dst3)} = Q_{n_1}^{(dst3)} \otimes Q_{n_1}^{(dst3)}$ in (2.2.15) . From a functional point of view, the eigenvalues in $\Lambda_{\mathbf{n}}[f]$ are obtained by sampling $f(x, y)$ over the points $(x_j, y_k) = (\frac{j\pi}{n_1+1}, \frac{k\pi}{n_2+1})$ ($j, k \in \{1, \dots, n\}$).

2.3 Solution of structured linear systems with direct methods

The first solution methods for structured linear systems of equations were direct ones. Whereas unstructured linear systems are solved in $O(n^3)$ arithmetic operations with Gaussian elimination or Cholesky factorization, the classes of matrices we have described so far can be solved much faster. Concerning efficiency, we have to distinguish between Toeplitz matrices and matrices belonging to a trigonometric algebra.

Fast direct solvers requiring only $O(n \log n)$ operations exist for all trigonometric algebras presented in Chapter 2.2. These solvers are based on fast implementations of the trigonometric transforms. The FFT, which is needed for the

solution of circulant matrices, is described in much detail in Van Loan's book [77], the other transforms can be reduced to FFTs. A linear system $A_n x = b$ is solved in the following three steps:

- Apply the fast transform to both b and the first column of A_n , which gives the vectors bt and at .
- Perform componentwise division of bt by at , which results in a vector xt .
- Compute x by application of the fast inverse transformation to xt .

Therefore, three fast transforms plus $O(n)$ operations are necessary for the solution of a linear system. For dense matrices, the same number of operations is necessary to compute a matrix-vector product $A_n x$. Here, the componentwise division in the second step is replaced by a componentwise multiplication. Thus, for dense matrices, no iterative solver which is based on matrix-vector multiplications can be faster than a direct solver. Only for banded matrices with small bandwidth, where matrix-vector multiplication is carried out in $O(n)$ operations, iterative methods can be an improvement. We will come back to this idea when we describe multigrid methods in Chapter 3.

The situation is significantly more difficult for Toeplitz matrices. Whereas matrix-vector products involving Toeplitz matrices can be computed in $O(n \log n)$ operations (as will be described in the subsequent subsection), there is no straightforward algorithm for a fast direct solution of Toeplitz systems. Here, we only give a very brief overview of the development of direct methods for Toeplitz systems. A detailed description of these methods can be found in [27, 83, 74]. Levinson [76] was the first who developed an $O(n^2)$ algorithm. His method was improved to a complexity of $3n^2$ operations by Trench [111] and Zohar [118]. These algorithms start with a 1×1 system and recursively compute the solution of the larger systems with the so called Szegő recurrence relation. In each step, the algorithm requires the invertability of a principal submatrix of T_n . In the 1980s the first superfast direct solvers with a complexity of $O(n \log^2 n)$ were developed. These methods can be divided into two classes, depending on the fundamental idea on which they are based. The methods of Bitmead and Anderson [11] and Morf [80] are based on the fast Cholesky factorization for Toeplitz matrices and on the concept of displacement rank, which was introduced in [73]. The methods of Brent et al. [18], Musicus [81], and Ammar and Gragg [1] use a generalized Schur algorithm. The main problem of all these methods is that they break down if T_n has a singular or ill-conditioned submatrix. Several look-ahead algorithms have been developed to get rid of this problem. References for these algorithms and for further research on stability issues of direct Toeplitz solvers can be found in [83].

2.4 Iterative solution of Toeplitz systems

Whereas linear systems with matrices stemming from trigonometric algebras are solved in $O(n \log n)$ operations with direct methods, no such algorithm exists for Toeplitz matrices. For the iterative solution of a linear system, however, the most time-consuming part are matrix-vector products. There are two different ways to implement matrix-vector products involving Toeplitz matrices T_n in $O(n \log n)$, which are e.g. described in [83]. The first method embeds T_n into a circulant matrix of size $2n$, and computes the matrix-vector product $T_n x$ with two FFTs of size $2n$ in $O(n \log n)$. This is done by choosing B_n such that $\begin{pmatrix} T_n & B_n \\ B_n & T_n \end{pmatrix}$ is circulant, and then computing $T_n x$ in

$$\begin{pmatrix} T_n & B_n \\ B_n & T_n \end{pmatrix} \begin{pmatrix} y \\ 0 \end{pmatrix} = \begin{pmatrix} T_n x \\ * \end{pmatrix}$$

with the FFT. The other method makes use of the fact that each Toeplitz matrix can be written as a sum of a circulant and a skew-circulant matrix. Since for both matrix types, a matrix-vector product is computed with two FFTs, a total of 4 FFTs of size n is necessary to perform the multiplication $T_n x$. For banded Toeplitz systems, a matrix-vector product is even computed in $O(n)$ operations. For multilevel Toeplitz matrices $T_{\mathbf{n}}$, computation of a matrix-vector product has the same computational cost, because multilevel versions of the fast transforms are applied.

The most expensive part in iterative methods is usually the computation of matrix-vector products. Since we are mainly interested in positive definite matrices, the conjugate gradient method is one of the fastest solvers. When no preconditioner is used, one matrix-vector product must be computed in each iteration. This implies that Toeplitz systems are solved in $O(n \log n)$ if and only if the conjugate gradient method converges after a constant number of iterations. In the following, we describe the conjugate gradient method and its convergence properties. Since the desired convergence results are only obtained with the use of preconditioners, we will state criteria on how such a preconditioner must be chosen.

2.4.1 The conjugate gradient method

Krylov subspace methods belong to the fastest iterative solvers for linear systems $A_n x = b$. For positive definite matrices, the conjugate gradient (cg) method is the most frequently used iterative solution technique. It was developed by Hestenes and Stiefel [68], and it is described in detail in [58, 59, 67, 74]. The method is based on the idea that the solution of the linear system is equivalent to the minimization of the function

$$f(x) = \frac{1}{2} x^H A_n x - b^H x, \quad (2.4.1)$$

which has the gradient $\Delta f(x) = A_n x - b$. Each iteration of the algorithm follows the same pattern. Starting with an initial guess x_0 , the following steps are carried out for $k = 0, 1, 2, \dots$, until the residual $r_k = b - A_n x_k$ is below a given tolerance TOL :

- Choose a search direction d_k .
- Solve the 1D minimization problem $\alpha_k = \min_{\alpha} f(x_k + \alpha d_k)$.
- Compute the new iterate $x_{k+1} = x_k + \alpha_k d_k$.

The simplest method following this strategy is the method of steepest descent. Since f decreases most rapidly in the direction of the negative gradient $-\Delta f(x)$, d_k is chosen to be the residual $r_k = b - A_n x_k$. However, this simple choice is not practical, because for large and ill-conditioned A_n , many search directions run in parallel, and convergence becomes prohibitively slow. Therefore, the cg method uses a projection of r_k such that all search directions d_k are A_n -conjugate, i.e. $d_j^H A_n d_k = 0$ for $j \neq k$. It starts with $d_0 = r_0$, and then obtains d_{k+1} from r_{k+1} as the result of Schmidt's orthogonalization procedure applied to r_{k+1} and $\text{span}(d_0, \dots, d_k)$. Because the d_j are pairwise conjugate, this reduces to

$$d_{k+1} = r_{k+1} + \beta_k d_k,$$

where β_k is derived from $\langle d_{k+1}, d_k \rangle = 0$:

$$\beta_{k+1} = \frac{r_{k+1}^H r_{k+1}}{r_k^H r_k}.$$

Then

$$\alpha_k = -\frac{r_k^H r_k}{d_k^H A_n d_k}$$

is computed in order to minimize $f(x)$ along d_k . Now, the conjugate gradient method for the solution of the linear system $Ax = b$ can be stated as follows:

Algorithm 1 (Conjugate Gradient Method)

make an initial guess x_0

$$d_0 = r_0 = b - Ax_0$$

for $k = 0, 1, 2, \dots$

$$\alpha_k = -\frac{r_k^H r_k}{d_k^H A_n d_k}$$

$$x_{k+1} = x_k + \alpha_k d_k$$

$$r_{k+1} = r_k - \alpha_k A_n d_k$$

if $r_{k+1} = 0$ **then** STOP

$$\beta_{k+1} = \frac{r_{k+1}^H r_{k+1}}{r_k^H r_k}$$

$$d_{k+1} = r_{k+1} + \beta_{k+1} d_k$$

end

From Algorithm 1 we see that the computational cost is dominated by the matrix-vector product. Since this can be done with FFTs, the total cost of one iteration is $O(n \log n)$ operations. In the banded case, the complexity reduces to $O(n)$. Thus, we need to carry out a convergence analysis in order to find out under which circumstances the cg method converges after $O(1)$ iterations.

2.4.2 Convergence results and the use of preconditioners

A very general convergence analysis of the cg method is carried out in [5]. It only requires that A is positive definite and that the condition number $\kappa(A) = \frac{\lambda_{max}}{\lambda_{min}}$ is known. The main result is stated in the following theorem.

Theorem 4 (Axelsson and Barker, [5])

Let A be an Hermitian positive definite matrix with condition number $\kappa(A)$, and let x_{min} be the exact solution of the linear system $Ax = b$. Then, the error of the k -th iterate x_k satisfies

$$\|x_{min} - x_k\|_A \leq 2\|x_{min} - x_0\|_A \left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1} \right)^k . \quad (2.4.2)$$

For Toeplitz matrices, this estimate can be written in terms of generating functions. Application of Theorem 1 to (2.4.2) leads to

$$\|x_{min} - x_k\|_A \leq 2 \left(\frac{\sqrt{f_{max}} - \sqrt{f_{min}}}{\sqrt{f_{max}} + \sqrt{f_{min}}} \right)^k . \quad (2.4.3)$$

If there is more information available about the distribution of the eigenvalues than just the values of λ_{min} and λ_{max} , one obtains better bounds for the error than the one from Theorem 4. A much stronger result holds under the assumption that the eigenvalues of A are clustered around 1. That means all eigenvalues are in the interval $[1 - \epsilon, 1 + \epsilon]$, except for a few outlying ones. The following theorem shows that in this case, the cg method converges very fast.

Theorem 5 (Kailath and Sayed, [74])

Let A be an Hermitian positive definite matrix whose eigenvalues λ_j satisfy the condition

$$\begin{aligned} 0 < \delta \leq \lambda_1 \leq \dots \leq \lambda_i \leq 1 - \epsilon \leq \lambda_{i+1} \leq \dots \leq \lambda_{n-j} \\ \leq 1 + \epsilon \leq \lambda_{n-j+1} \leq \dots \leq \lambda_n . \end{aligned} \quad (2.4.4)$$

Then we have

$$\|x_{min} - x_k\|_A \leq 2\|x_{min} - x_0\|_A \left(\frac{1 + \epsilon}{\delta} \right)^i \epsilon^{k-i-j} \quad (k \geq i + j) . \quad (2.4.5)$$

In the following, these two theorems shall be applied to Toeplitz matrices. According to Theorem 1, the condition number of a Toeplitz matrix depends on the generating function. Thus, for strictly positive f , the condition number is low, and Theorem 4 gives a satisfactory bound for the convergence rate. However, the clustering property of the eigenvalues, which is a prerequisite for the application of Theorem 5, does not hold for most Toeplitz matrices at all. By Theorem 1, the eigenvalues of $T_n[f]$ are equally distributed as $f(\frac{2\pi j}{n})$. Therefore, the original linear system $T_n x = b$ must be replaced by another Hermitian positive definite system with the same solution x and with better spectral properties. This is done by choosing a preconditioner P_n and formally multiplying its inverse to both sides of the equation:

$$P_n^{-1} T_n x = P_n^{-1} b .$$

The goal is to choose P_n such that the spectrum of $P_n^{-1} T_n$ is clustered, or the condition number $\kappa(P_n^{-1} T_n)$ of the preconditioned matrix is close to 1. The following two properties characterize a good preconditioner.

- Preconditioners with clustered spectrum at 1 are often called *superlinear* and the cg method applied to $P_n^{-1} T_n$ is said to converge superlinearly. Superlinearity holds if the eigenvalues of $P_n^{-1} T_n - I_n$ have a proper cluster at 0, i.e. if for any $\epsilon > 0$, the number of eigenvalues $> \epsilon$, denoted $\gamma(\epsilon)$, is $O(1)$. If $\gamma(\epsilon) = o(n)$ the cluster is denoted general, and the preconditioner *sublinear*.
- Independent of the clustering property, a preconditioner is called *optimal* if the eigenvalues of T_n and P_n are spectrally equivalent. This property holds if and only if all eigenvalues of $P_n^{-1} T_n$ lie in the interval $[\alpha, \beta]$ with $0 < \alpha$ and $\beta < \infty$.

If a preconditioner satisfies both superlinearity and optimality, the cg method applied to the preconditioned system is expected to have excellent convergence properties. In the remainder of this chapter, we summarize the most important preconditioners which were developed for both well-conditioned and ill-conditioned Toeplitz systems. Special emphasis is put on superlinearity and optimality.

2.5 Preconditioners for one-level Toeplitz matrices

As we have pointed out in the previous section, the preconditioner P_n must be a matrix which can be inverted in $O(n \log n)$ operations, and which leads to a clustered spectrum of $P_n^{-1} T_n$. The first requirement is met by all matrix algebras, and it turns out that for a wide range of positive definite one-level Toeplitz systems, many different algebra preconditioners also meet the second requirement. These are especially useful for Toeplitz systems which are well-conditioned. Therefore, we begin this chapter with a description of algebra preconditioners for well-conditioned Toeplitz matrices. In particular, we emphasize circulant preconditioners. In the second part, we present preconditioners for ill-conditioned Toeplitz

systems. These are either modified algebra preconditioners or Toeplitz matrices themselves.

2.5.1 Algebra preconditioners for well-conditioned Toeplitz matrices

Preconditioners taken from a trigonometric matrix algebra are particularly interesting for Toeplitz matrices, because they are inverted in $O(n \log n)$ with the FFT. Circulant preconditioners are not just the oldest class of algebra preconditioners for Toeplitz systems T_n , but still one of the most popular for matrices of type **(I)** from Section 2.1.3, i.e. for matrices with strictly positive generating function. A comprehensive presentation of circulant preconditioners is for example given in [27, 83]. The first circulant preconditioner $c_S(T_n)$ was developed by G. Strang [107]. $c_S(T_n)$ is constructed by copying the central diagonals of T_n and reflecting them to complete the circulant structure. For an n -by- n Hermitian Toeplitz matrix T_n the diagonals s_j of Strang's preconditioner $c_S(T_n) = [s_{k-l}]_{0 \leq k, l < n}$ are defined by

$$s_j = \begin{cases} t_j & 0 \leq j \leq \lfloor n/2 \rfloor, \\ t_{j-n} & \lfloor n/2 \rfloor < j < n, \\ s_{n+j} & 0 < -j < n. \end{cases} \quad (2.5.1)$$

It was shown by R. Chan [20] that $c_S(T_n)$ minimizes $\|C_n - T_n\|_1$ and $\|C_n - T_n\|_\infty$ over all Hermitian circulant matrices C_n . Moreover, Chan proved that the preconditioner is superlinear if T_n corresponds to a strictly positive generating function in the Wiener class. T. Chan [34] developed a circulant preconditioner $c_F(T_n)$ which minimizes

$$\|C_n - T_n\|_F \quad (2.5.2)$$

over all circulant matrices C_n , where $\|\cdot\|_F$ denotes the Frobenius norm. The diagonals of $c_F(T_n)$ are defined by

$$c_j = \begin{cases} \frac{(n-j)t_j + jt_{j-n}}{n} & 0 \leq j \leq n-1, \\ c_{n+j} & 0 < -j < n-1. \end{cases} \quad (2.5.3)$$

In [32] superlinearity of $c_F(T_n)$ is proved for 2π -periodic continuous functions which are strictly positive. Furthermore, the preconditioner is positive definite if T_n is positive definite [26]. For the Strang preconditioner, this property only holds for large n . The so-called superoptimal preconditioner $c_T(T_n)$ of Tyrtysnikov [113] is the minimizer of

$$\|I_n - C_n^{-1}T_n\|_F \quad (2.5.4)$$

over all non-singular circulant matrices C_n . It is given by

$$c_T(T_n) = c_F(T_n T_n^H) c_F(T_n)^{-1} \quad (2.5.5)$$

The superoptimal preconditioner is superlinear for positive functions in the Wiener class, and preserves positive definiteness. The same properties are proved for a preconditioner developed by Huckle [70], which minimizes

$$\|I_n - C_n^{-1/2} T_n C_n^{-1/2}\|_F . \quad (2.5.6)$$

A unifying approach for the construction of circulant preconditioners is presented in [31]. Most circulant preconditioners can be constructed using convolution products of well-known kernels with the generating function f . For a given kernel $K_n(x)$ defined on $[0, 2\pi]$, C_n is the circulant matrix with eigenvalues

$$\lambda_j(C_n) = (K_n * f)\left(\frac{2\pi j}{n}\right) , \quad 0 \leq j < n . \quad (2.5.7)$$

In [31], superlinearity is proved for all preconditioners constructed from (2.5.7) with $K_n * f$ tending to f , uniformly on $[-\pi, \pi]$. The classical preconditioners of Strang, T. Chan, and Huckle can all be obtained with this approach by using Dirichlet and Fejér kernels.

Not only circulant preconditioners have been developed for Toeplitz matrices. Several other algebras can be used for the construction of superlinear preconditioners. Two different tau preconditioners are presented by Bini and Di Benedetto [8]. The so-called natural preconditioner is simply $\tau[f]$, whereas the optimal preconditioner minimizes $\|\tau - T_n\|_F$ over all tau matrices τ . In [97], a proof of superlinearity is given for both preconditioners under the assumption that T_n belongs to a continuous, 2π -periodic functions $f > 0$. Bini and Favati [10] proved superlinearity for their optimal Hartley preconditioner. R. Chan et. al. [25] obtained the same result for their preconditioner based on the cosine transform.

2.5.2 Preconditioners for ill-conditioned Toeplitz matrices

Toeplitz matrices whose generating functions have zeros are ill-conditioned. The preconditioners from Section 2.5.1 cannot simply be applied to these matrices. Tyrtyshnikov [114] proved that both the Strang and the T. Chan preconditioner do not converge superlinearly when applied to an ill-conditioned matrix. He showed that $c_F(T_n)^{-1}T_n$ and $c_T(T_n)^{-1}T_n$ have $O(n^{\nu/(\nu+\mu)})$ and $O(n^{\nu/(\nu+1)})$ eigenvalues outside $[1 - \epsilon, 1 + \epsilon]$, respectively. ν denotes the order of the zeros of f and μ the degree of smoothness. In general, it is very difficult to find good circulant preconditioners, but Potts and Steidl [88] presented a superlinear ω -circulant preconditioner $W_{\omega,n}(f)$. They chose a number $\xi_n \in [-\pi, \pi]$ with $f(2\pi j/n + \xi_n) > 0$ for $0 \leq j < n$. The matrix containing the eigenvalues is defined

$$\Lambda_n = \text{diag}(f(\xi_n), f(2\pi/n + \xi_n), \dots, f(2(n-1)\pi/n + \xi_n)) , \quad (2.5.8)$$

and ω is chosen $\omega = e^{-i\xi_n}$. In [88], the authors proved that the eigenvalues of $W_{\omega,n}(f)^{-1}T_n(f)$ are clustered around 1, and therefore the preconditioner is superlinear.

Several Toeplitz preconditioners with small bandwidth have been proposed for ill-conditioned Toeplitz systems. The first one was presented by R. Chan [21]. It is based on the idea that f is approximated by a trigonometric polynomial of fixed degree matching the zeros of f . If f has a single zero x_0 in $] - \pi, \pi]$ of order 2ν , then the polynomial $(2 - 2 \cos(x - x_0))^\nu$ is chosen. If f has several zeros, the polynomial is the product $\prod_{j=1}^k (2 - 2 \cos(x - x_j))^{\nu_j}$. The condition number is bounded, but the eigenvalues of the preconditioned system are not clustered. Thus, the preconditioner is optimal, but not superlinear, and therefore the constant $O(1)$ in the number of iterations is expected to be rather high. Chan and Tang [29] developed a preconditioner which has slightly more bandwidth, but minimizes the relative error $\|(f - p)/f\|_\infty$ in addition to matching the zeros of f . The minimization is done with a modified Remez algorithm, which can become computationally too expensive. Based on this idea, Serra [92] proposed a superlinear band Toeplitz preconditioner. Instead of the Remez algorithm, his method is based on interpolating Chebyshev polynomials, which can be done in $O(n \log n)$. Noutsos and Vassalos [87] presented a preconditioner which is the product of three band Toeplitz matrices. The preconditioned system is optimal in the sense that its condition number is bounded. Serra also proposed a tau preconditioner [97] which applies the Sherman-Morrison-Woodbury formula to the preconditioned system and reaches superlinear behavior for large classes of matrices.

2.6 Difficulties with preconditioners for multilevel Toeplitz matrices

In the previous section, several different preconditioners have been presented for one-level Toeplitz matrices. Even in the ill-conditioned case, where f has a finite number of zeros, superlinear convergence could be achieved for some of them. Obviously, one hopes to carry over many of these results to multilevel Toeplitz matrices, but unfortunately this is not possible for most of the preconditioners studied so far. First, we present negative results concerning superlinear and optimal convergence. Then, in the last part of this section, we describe what preconditioners have been developed and used under these restrictions.

2.6.1 Negative results for multilevel matrix algebra preconditioners

For the solution of Toeplitz systems with the PCG method, the development of superlinear or at least optimal preconditioners is the main field of research activity. Recall from Chapter 2.4.2 that a preconditioner is called superlinear if $P_n^{-1}T_n$ has clustered eigenvalues around 1, and optimal if the condition number of $P_n^{-1}T_n$ is bounded. Preconditioners from trigonometric matrix algebras were most popular

for one-level Toeplitz matrices, because they are inverted in $O(n \log n)$ operations with fast trigonometric transforms. For d -level Toeplitz systems with $d > 1$, however, they cannot be applied with the same efficiency. The first negative result was obtained by Serra and Tyrtysnikov [104], who state that no circulant and circulant-like preconditioner can be superlinear. The result is extended in [105, 100], showing that no preconditioner from a trigonometric algebra is superlinear, not even if the generating function f is strictly positive. It is shown that $\gamma_{\mathbf{n}}(\epsilon)$, the number of eigenvalues of $P_{\mathbf{n}}^{-1}T_{\mathbf{n}}$ lying outside $[1 - \epsilon, 1 + \epsilon]$, is bounded from below by

$$\gamma_{\mathbf{n}}(\epsilon) \geq c(\epsilon) \mathbf{n} \sum_{k=1}^d \frac{1}{n_k} , \quad (2.6.1)$$

where n_k is the k -th partial dimension of $T_{\mathbf{n}}$, i.e. $\mathbf{n} = n_1 \cdot \dots \cdot n_d$. This is not at all satisfactory if d becomes larger. From [95] we know that this bound is sharp,

i.e. that for example multilevel circulant preconditioners have $O(\mathbf{n} \sum_{k=1}^d \frac{1}{n_k})$ outliers.

For $d = 2$, this implies that the number of outliers is $O(\sqrt{\mathbf{n}})$, which can still be acceptable if the condition number can be bounded independent of \mathbf{n} .

If superlinearity cannot be reached for multilevel preconditioners from algebras, we wish to guarantee at least optimality, i.e. spectral equivalence of $T_{\mathbf{n}}$ and $P_{\mathbf{n}}$. Following [84], $T_{\mathbf{n}}$ and $P_{\mathbf{n}}$ are called spectrally equivalent if all eigenvalues of $P_{\mathbf{n}}^{-1}T_{\mathbf{n}}$ are contained in the interval $[\alpha, \beta]$, independent of \mathbf{n} with $0 < \alpha \leq \beta < \infty$, and essentially spectrally equivalent if there are only a constant number of outliers $> \beta$. In [84] and [85], it is proved that for most important classes of multilevel Toeplitz matrices, preconditioners from trigonometric algebras satisfy none of the two properties.

2.6.2 Multilevel Toeplitz preconditioners

The negative results of the previous subsection state that no superlinear preconditioner from a trigonometric algebra can be found for BTTB systems, and that, in the ill-conditioned case, not even an optimal algebra preconditioner exists. Accepting these facts, one can nevertheless choose a multilevel circulant or ω -circulant preconditioner. In the two-dimensional case, the results are still acceptable for many linear systems. If the generating function is strictly positive, it is, for some preconditioners, possible to show that

- the eigenvalues of the preconditioned system $P_{\mathbf{n}}^{-1}T_{\mathbf{n}}$ are inside an interval $[c_1, c_2]$, independent of \mathbf{n}
- $P_{\mathbf{n}}^{-1}T_{\mathbf{n}}$ has a general cluster at 1, i.e. there are only $o(\mathbf{n})$ eigenvalues outside $[1 - \epsilon, 1 + \epsilon]$

For example, these two properties can be proved for the two-level version of the T. Chan preconditioner [35] and for the natural two-level tau preconditioner [7]

under the condition that $f(x, y) > 0$. For ill-conditioned BTTB systems, the convergence properties of algebra preconditioners deteriorate further. For example, the two-level-tau preconditioner from [7] leads to a preconditioned system with $O((n_1 + n_2) \log(\kappa(T_{\mathbf{n}})))$ outliers.

Therefore, a different strategy is pursued for the development of multilevel preconditioners: BTTB matrices themselves are used as preconditioners. Serra [91] developed the first optimal preconditioner for ill-conditioned BTTB systems corresponding to a function $f(x, y)$ with zeros. He chose $P_{\mathbf{n}}$ such that the corresponding function $p(x, y)$ is a trigonometric polynomial of minimum degree satisfying $\frac{f(x, y)}{p(x, y)} > 0$ for all x and y . In [82], similar preconditioners are constructed, matching the zeros of $f(x, y)$. If f is a nonnegative piecewise continuous real-valued function with zeros x_1, \dots, x_k of orders $2\nu_1, \dots, 2\nu_k$, then the BTTB matrix $P_{\mathbf{n}}$ corresponds to the function

$$p(x, y) = \prod_{j=1}^k (4 - 2 \cos(x - x_j) - 2 \cos(y - y_j))^{\nu_j}.$$

This preconditioner also ensures optimal convergence of the pcg method. In [86] the authors propose a new BTTB preconditioner which does not require the explicit notion of the generating function, but uses an approximation instead. Although it is not optimal in general, it leads to very fast convergence for many example problems.

All these BTTB preconditioners suffer from the fact that another BTTB system must be inverted. However, they are very useful for the preconditioning of dense BTTB matrices, because the problem of solving a full BTTB system is reduced to the problem of solving a banded one. This is the point where multigrid methods become highly efficient. It will turn out that with a multigrid solver, an ill-conditioned positive definite BTTB system can be solved in $O(\mathbf{n} \log(\mathbf{n}))$. If the matrix is banded, it is even possible to reach an overall cost of $O(\mathbf{n})$ arithmetic operations.

Chapter 3

Multigrid methods for structured linear systems

Multigrid methods are universally accepted as the fastest numerical methods for the solution of elliptic partial differential equations. Moreover, they are successfully applied to problems in image processing, to other types of PDEs, to integral equations, and to many other applications. Starting with the articles of Brandt [13] and Hackbusch [64], multigrid methods became increasingly popular. Discretization of elliptic PDEs, for example with finite differences, mostly results in ill-conditioned linear systems of equations. Especially if these systems are positive definite, multigrid methods are the fastest iterative solvers.

For structured linear systems of equations, multigrid methods offer fascinating possibilities. At the end of Chapter 2, we have illustrated the difficulties preconditioned Krylov subspace methods face when they are used for the solution of ill-conditioned multilevel Toeplitz systems. In recent years, multigrid methods have been applied to (multilevel) Toeplitz matrices and to matrices which are elements of a trigonometric algebra. Generating functions are the most important tool in the construction of multigrid methods for these specific matrix classes. All essential parts of a multigrid method such as prolongation, restriction, and computation of the coarse grid matrices are designed with the help of generating functions.

After giving a general introduction to multigrid methods with special emphasis on algebraic multigrid, we will describe how the structure of the matrices and their correspondence to generating functions are exploited for the development of multigrid methods. Then, we will review recent results on multigrid methods for matrices belonging to a trigonometric algebra or to the Toeplitz class. In Section 3.3.2, we are going to extend some of these results to the DST-III algebra. In Section 3.5, we will introduce multigrid methods for the solution of block Toeplitz systems, and describe how Toeplitz matrices are interpreted as block Toeplitz matrices. Some new results on eigenvectors and eigenvalues allow a more detailed

analysis of the methods. Eventually, we will use the results on block Toeplitz matrices for the development of different multigrid methods for functions with multiple zeros.

3.1 General multigrid methods

In this section, we give an introduction to multigrid methods, which are not restricted to a specific matrix class and which do not use generating functions. These rather general methods will constitute the framework for the development of more specific multigrid methods that can be used for certain classes of structured matrices. We begin this section with a presentation of multigrid methods, which is strongly influenced by the books of Trottenberg et. al. [112] and Briggs et. al. [19]. After describing the two main ingredients of multigrid methods, smoothing and coarse grid correction, we present two-grid and multigrid algorithms in a rather general form. In the second part of this section, we focus on the algebraic multigrid method (AMG) developed in [14, 17, 90]. Since the convergence theory of Ruge and Stüben [90] is used throughout this thesis, we lay special emphasis on their results. Finally, we describe how multigrid methods can be used as preconditioners for other iterative solvers such as Krylov subspace methods.

3.1.1 The principles of multigrid methods

The high efficiency of multigrid methods is only obtained when their two main components, smoothing and coarse grid correction, complement each other well. Behind each of these components, there is one fundamental principle. We start with a description of the smoothing principle and the coarse grid principle, and explain why each of them alone is not suitable for the construction of a standalone solver. Then, we present a generic two-grid method, which combines the two principles and hence becomes an efficient solver. Since this method is too expensive for practical calculations, smoothing and coarse grid correction are applied recursively, leading to a multigrid method.

Two-grid methods: Smoothing and coarse grid correction

The smoothing principle is derived from stationary iterative methods, which can be used for the solution of a linear system $A_h x = b$ starting with the initial guess x_0 . In PDE applications, the matrix A_h is assumed to be the result of a discretization on the grid Ω_h . Typical examples are the damped Richardson, Jacobi, or Gauss-Seidel method. One iteration of a stationary method is of the form

$$x^{(\nu+1)} = x^{(\nu)} - \omega \cdot M \cdot (A_h x^{(\nu)} - b) \quad (3.1.1)$$

with damping parameter ω . The damped Richardson method is obtained by choosing M to be the identity matrix, the damped Jacobi method by choosing

$M = \text{diag}(A_h)^{-1}$, and the damped Gauss-Seidel method with lexicographic ordering by choosing M to be the inverse of the lower triangular part of A_h . (3.1.1) can be rewritten

$$x^{(\nu+1)} = S \cdot x^{(\nu)} + \omega M \cdot b \ , \quad (3.1.2)$$

where

$$S = I - \omega M A_h \quad (3.1.3)$$

is called the iteration matrix. From the initial error $e^{(0)} = x - x^{(0)}$, the error after ν steps is computed with $e^{(\nu)} = S^\nu e^{(0)}$. Since for ill-conditioned matrices A_h , the spectral radius of S tends to 1, convergence of the three stationary methods is rather slow. However, if one of the stationary methods is applied to an elliptic problem, almost all oscillatory parts of $e^{(0)}$ are removed after only a few iterations, and the error becomes smooth. Further iterations only reduce the error very slowly, because stationary methods are not suitable for removing smooth error components. This property can be summarized in the following form (see [112]).

Smoothing principle: Many classical iterative methods such as the damped Richardson, Jacobi, or Gauss-Seidel method have a strong smoothing effect on the error, when they are applied to discrete elliptic problems.

The second principle is obtained from the following observation: A smooth error is well approximated on a coarser grid, i.e. without much loss of information, although a significantly smaller number of grid points is used. In most cases, the number of grid points is halved in each dimension, leading to a coarse grid Ω_{2h} . Other choices such as a reduction by a factor 4 in each direction or coarsening in only one direction are favorable in specific examples. Then, on the coarser grid, the error is more oscillatory, which means that it can be treated with the stationary method more efficiently. This property is summarized in the following form (see [112]).

Coarse grid principle: A smooth error term is well approximated on a coarse grid. All computations on the coarse grid are significantly less expensive, because there are much fewer grid points than on the fine grid.

It is important to note that the coarse grid principle only holds if the error is smooth. Oscillatory components are not visible on a coarser grid, because they coincide with smooth components due to aliasing of frequencies.

Now we combine the two principles and construct a fast algorithm for the solution of linear systems $A_h x = b$ obtained from discretization of a PDE on a fine grid Ω_h . After a few iterations with a stationary method, resulting in the approximate solution \hat{x} , the error $e_h = x - \hat{x}$ is smooth. Of course, we cannot compute the error, but we can compute the residual $r_h = b - A_h \hat{x}$. With r_h , we obtain another equation involving the matrix A_h , the *residual equation*

$$A_h e_h = r_h \ . \quad (3.1.4)$$

The aim is to find an approximation for e_h with the residual equation, and therefore a new approximation \hat{x} for the solution x . Such an approximation is found by computing r_h and restricting it to $r_{2h} = R \cdot r_h$ on the coarse grid Ω_{2h} with the *restriction matrix* R . Then r_{2h} is the coarse grid version of r_h , i.e. an approximation on the coarse grid with only half as many grid points. In the next step, the coarse grid version of the residual equation $A_{2h}e_{2h} = r_{2h}$ is solved exactly. The correction e_{2h} is interpolated back to the fine grid with the *prolongation matrix* P , and the result $\bar{e}_h = P \cdot e_{2h}$ is used to compute the new \hat{x} . After this coarse grid correction has been computed, further smoothing iterations can be applied. Altogether, each iteration step of a two-grid method consists of presmoothing, coarse grid correction, and postsmoothing. Following the presentation in [112], we can formulate a generic two-grid algorithm.

Algorithm 2 (Generic two-grid method)

For the solution of $A_h x = b$, start with an initial guess $x^{(0)}$. With the following iteration, which is denoted $TGM(x^{(k)}, A_h, b, \nu_1, \nu_2)$, compute $x^{(k+1)}$ from $x^{(k)}$, until the error is below a given tolerance:

1. **Presmoothing:**

Perform ν_1 iterations of a stationary method of the form (3.1.2), i.e.

$$\bar{x}^{(k)} = SMOOTH^{\nu_1}(x^{(k)}, A_h, b)$$

2. **Coarse grid correction:**

- Compute the residual: $r_h = b - A_h \bar{x}^{(k)}$
- Restrict the residual: $r_{2h} = R \cdot r_h$
- Solve exactly on Ω_{2h} : $A_{2h}e_{2h} = r_{2h}$
- Interpolate the correction: $\bar{e}_h = P \cdot e_{2h}$
- Add the coarse grid correction: $\hat{x}^{(k)} = \bar{x}^{(k)} + \bar{e}_h$

3. **Postsmoothing:**

Perform ν_2 iterations of a stationary method of the form (3.1.2), i.e.

$$x^{(k+1)} = SMOOTH^{\nu_2}(\hat{x}^{(k)}, A_h, b)$$

This algorithm is a generic procedure. Its components have to be chosen for the problem at hand. In the following, we wish to describe how efficient two-grid methods are defined for elliptic problems. We use the discrete 2D Poisson equation as a model problem to illustrate the main ideas.

Example 1 (Discrete 2D Poisson equation) The Poisson equation in two dimensions on the unit square $\Omega = [0, 1]^2$ with Dirichlet boundary conditions is given by

$$\begin{aligned} -\Delta u(x, y) &= g(x, y) && \text{for } (x, y) \in \Omega \\ u(x, y) &= 0 && \text{for } (x, y) \in \partial\Omega \end{aligned} \quad (3.1.5)$$

Finite-difference discretization with a 5-point stencil on the grid Ω_h yields a linear system of the form $A_h x = b$ with

$$A_h = I \otimes A^{(1D)} + A^{(1D)} \otimes I$$

and $A^{(1D)} = \text{tridiag}(-1, 2, -1)$.

The following individual components of a two-grid solver have to be specified.

- The stationary method must be chosen such that it serves as a smoother, i.e. such that a small number of iterations significantly reduces the oscillatory components of the error. For elliptic problems, the damped Jacobi and Gauss-Seidel methods have good smoothing properties. In many cases, the red-black version of the Gauss-Seidel method, which uses a different ordering of the grid points, leads to better smoothing. Following [112], a measure for the quality of the smoother is the *smoothing factor*, which is defined to be the worst factor by which high frequency error components are reduced in each iteration of the smoother. For the model problem of Example 1, the smoothing factor is $\frac{2}{3}$ for damped Jacobi with optimal $\omega = \frac{4}{5}$, $\frac{1}{2}$ for lexicographic Gauss-Seidel with $\omega = 1$, and $\frac{1}{4}$ for red-black Gauss-Seidel with $\omega = 1$. In many applications, more sophisticated smoothers such as an ILU-factorization [78] or the SPAI algorithm [63] are used.
- The intergrid transfer operators, i.e. restriction and prolongation, can be chosen from a huge variety of matrices. A standard choice which is used in many multigrid methods is the full weighting operator. In 1D, it is given by the matrix

$$R^{(1D)} = \frac{1}{4} \cdot \begin{pmatrix} 1 & 2 & 1 & & & & \\ & 1 & 2 & 1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & & & 1 & 2 & 1 \\ & & & & & & & & & 1 & 2 & 1 \end{pmatrix}, \quad (3.1.6)$$

and in 2D, by the matrix $R^{(2D)} = R^{(1D)} \otimes R^{(1D)}$. A typical choice for prolongation is the bilinear interpolation. It is described by the matrix $P^{(1D)} = 2 \cdot (R^{(1D)})^T$ in 1D and by the matrix $P^{(2D)} = 4 \cdot (R^{(2D)})^T$ in 2D. These restriction and prolongation operators satisfy the *variational property*

$$R = c \cdot P \quad (3.1.7)$$

with a constant $c \in \mathbb{R}$.

- Eventually, the coarse grid matrix A_{2h} can be obtained in two different ways, either by rediscrctization on the coarse grid or by algebraic computation. Depending on this choice A_{2h} is called a
 - **natural coarse grid operator:** The problem is discretized again on the coarser grid. This means A_{2h} is obtained with the same discretization technique as A_h , just with fewer grid points. This approach is chosen in most geometric multigrid methods, where matrices are obtained from discretization on real grids.
 - **Galerkin coarse grid operator:** A_{2h} is obtained purely in algebraic terms as the product

$$A_{2h} = R \cdot A_h \cdot P \quad . \quad (3.1.8)$$

Since this approach can still be applied when there is no actual grid, it is chosen for all algebraic multigrid methods.

Multigrid methods and convergence

The two-grid method cannot be used in most practical applications, because the size of the matrix A_{2h} is still too large for an exact solution of the residual equation. Nevertheless, two-grid methods are the basis both for the practical development of multigrid methods and for a theoretical investigation concerning convergence of the methods. A multigrid method is obtained by recursive application of the two-grid idea. Instead of being computed exactly, the solution e_{2h} of the residual equation is approximated by a small number γ of two-grid iteration steps. More formally, it is computed as follows.

- If A_{2h} is already small enough, solve $A_{2h}e_{2h} = r_{2h}$ directly as in Algorithm 2.
- If the size of A_{2h} is too big for a direct solution, approximate e_{2h} with k two-grid iterations, i.e.

$$e_{2h} = TGM^\gamma(0, A_{2h}, r_{2h}, \nu_1, \nu_2) \quad .$$

The approximate solution leads to a residual equation on a third grid Ω_{4h} , which can either be solved exactly or be approximated by a further level of two-grid iterations. This procedure of recursive two-grid application continues until the linear systems are small enough for being solved exactly. This multigrid method generates a hierarchy of grids $\Omega_{2h}, \Omega_{4h}, \Omega_{8h}$, etc. and the coarse grid matrices A_{2h}, A_{4h}, A_{8h} , etc. Different choices of γ result in different types of multigrid cycles. Figure 3.1 shows, for a four-grid method, the structure of one multigrid cycle with $\gamma = 1$ and $\gamma = 2$. Filled circles denote smoothing and empty circles denote exact solution of the residual equation. For $\gamma = 1$, the algorithm is called V-cycle, for $\gamma = 2$, W-cycle.

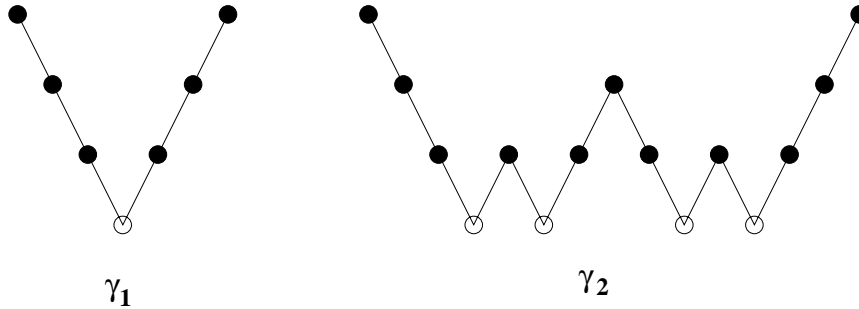


Figure 3.1: One four-grid iteration step for $\gamma = 1$ and $\gamma = 2$

So far, we have only described the multigrid algorithm and not yet said anything about the computational cost. The total cost depends on two factors, the cost per iteration and the number of iterations the multigrid method needs to converge. The cost per iteration can be determined easily. We assume that so many grids are used that the cost of the exact solution of the residual equation on the coarsest grid is negligible. Then, in each iteration, the cost is dominated by the matrix-vector products needed for the smoothers and for computation of the residuals on the finest grid. It can be shown (see e.g. [112, 19]) that both the V-cycle iteration and the W-cycle iteration are asymptotically of the same complexity as a matrix-vector product involving A_h .

In multigrid history, several approaches towards a convergence theory have been presented. The main goal is to show that a multigrid method converges optimally, i.e. that it converges after a constant number of iterations independent of the matrix size. One approach is the classical qualitative multigrid theory of Hackbusch [65, 66], which is based on certain regularity conditions for elliptic partial differential equations. The results are obtained for natural coarse grid operators. A more practical approach is the local Fourier analysis, which is developed in [13, 15] and described in detail in [112]. It is more a tool to obtain quantitative bounds for the smoothing and coarse grid factors than a theory. It is based on the idea that a discrete operator with nonconstant coefficients is linearized and replaced, locally, by an operator with constant coefficients. A different type of convergence theory has been developed for the algebraic multigrid method, which will be described in more detail in the subsequent section.

3.1.2 Algebraic multigrid

In contrast to the geometrically oriented multigrid methods described in the previous section, algebraic multigrid (AMG) methods do not require a real grid. All components of the multigrid method are constructed in a purely algebraic way. Descriptions of this method, which was introduced in [14, 17], can be found in

[90, 112, 19]. After presenting the fundamental concepts of AMG, we focus on the convergence theory presented in [90].

Multigrid without real grids

Formally, an AMG method for the solution of a linear system $Ax = b$ consists of the same fundamental components as its geometric counterpart: grids, smoothers, prolongation and restriction operators, coarse grid matrices, and solvers on the coarsest level. In the absence of a physical grid, a grid in the algebraic sense consists of the indices $1, \dots, n$ of the x_j . On this grid, all the other components are defined as in the geometric case. However, there is one fundamental conceptual difference between the two types of multigrid methods. In geometrically oriented methods, the grid hierarchy is usually given by the geometry, and the coarsening strategy is quite a straightforward one, but much effort is spent on the choice of an adequate smoother. It is important that the error is physically smooth, i.e. that it has a low spatial frequency, and this is not always easy to achieve. In AMG methods, on the other hand, a standard smoother is fixed at the beginning, which is usually the damped Jacobi or the Gauss-Seidel method. Then, much more effort is made in choosing an efficient coarsening strategy. Moreover, the concept of smoothness is quite different in AMG. Before we can start with a more detailed description of AMG methods, we need to introduce a bit of notation on inner products and norms. In addition to the Euclidean inner product $\langle u, v \rangle$, the following inner products are defined for the system matrix A :

$$\begin{aligned} \langle u, v \rangle_0 &= \langle \text{diag}(A)u, v \rangle, \quad \langle u, v \rangle_1 = \langle Au, v \rangle, \\ \langle u, v \rangle_2 &= \langle \text{diag}(A)^{-1}Au, Av \rangle \end{aligned} \tag{3.1.9}$$

The respective norms, which are derived from these inner products, are denoted $\|\cdot\|_i$, $i = 0, 1, 2$.

At first glance, the two-grid method is of the same form as the generic method of Algorithm 2. In the following, we describe how its individual components are defined. Let us start with some comments on smoothing. As in geometric multigrid, we start with a small number of iterations with a standard stationary method. The main difference to geometric multigrid is that we do not observe whether the error is smooth or not. Instead, we define it. Put informally, an error is called algebraically smooth if it is not reduced effectively by the smoother. Hence, smoothness depends on the choice of the smoother. Mathematically, this means that e is smooth if $\|Se\|_1 \approx \|e\|_1$ with the smoothing matrix S defined in (3.1.3). It can be shown that a smooth error varies slowly in the direction of strong connections between grid points. If the grid point j strongly depends on the grid point k , the value x_j can be interpolated well from the value x_k . The notion of strong influence and strong dependence is the principal motivation behind the original coloring algorithm for the construction of the coarse grid. Since we do not use this algorithm in our work, we refer the reader to [19, 112] for a more

detailed description. The prolongation operator is chosen such that the values x_k at the fine grid points are obtained as a weighted sum of the values at the coarse grid points. The restriction matrix is chosen $R = P^H$, which implies that the variational property is satisfied. Since the AMG is a purely algebraic method, the coarse grid matrix is computed using the Galerkin condition

$$A_C = R \cdot A \cdot R^H \quad .$$

Finally, a postsmoother is applied to smooth the error obtained from coarse grid correction.

In the following, we present a two-grid and a multigrid version of the AMG method. The notation chosen here will be used throughout this thesis.

Algorithm 3 (AMG, Two-grid algorithm)

The following algorithm defines one iteration of the two-grid version of the AMG method.

$$\begin{array}{l|l} x^{(k+1)} = TGM(x^{(k)}, A, b, \nu_1, \nu_2) & \\ \hline 1 & \bar{x} = SMOOTH^{\nu_1}(x^{(k)}, A, b) \\ 2 & r = A \cdot \bar{x} - b \\ 3 & r_C = R \cdot r \\ 4 & A_C = R \cdot A \cdot R^H \\ 5 & \text{solve } A_C \cdot y = r_C \\ 6 & \hat{x} = \bar{x} - R^H \cdot y \\ 7 & x^{(k+1)} = SMOOTH^{\nu_2}(\hat{x}, A, b) \end{array}$$

In matrix notation, one iteration of the two-grid method can be written

$$TG = S^{\nu_2}(I - R^H(RAR^H)^{-1}RA)S^{\nu_1} \quad . \quad (3.1.10)$$

This two-grid method will mostly be used as a starting point for a theoretical analysis and for convergence proofs. Almost all proofs for structured matrices have first been devised for the two-grid method before being extended to the multigrid method. In most practical applications, the size of A_C is too big for an exact solution of the system $A_C \cdot y = r_C$. Therefore, the two-grid idea is applied recursively, as in the geometric case.

Algorithm 4 (AMG, Multigrid algorithm)

The following algorithm shows the structure of the multigrid version of the AMG method. For the solution of the linear system $A_1 x_1 = b_1$, starting from an initial guess $x^{(0)}$, one V-cycle iteration computes $x^{(k+1)}$ from $x^{(k)}$ with

$$x^{(k+1)} = MGM(levs, x^{(k)}, A_1, b_1, \nu_1, \nu_2),$$

where *levs* denotes the number of grids which are used.

$$x_j^{(out)} = MGM(l, x_j^{(in)}, A_j, b_j, \nu_1, \nu_2)$$

IF ($l = 1$) **THEN** solve $A_j x_j^{(out)} = b_j$
ELSE **1** $\bar{x}_j = SMOOTH^{\nu_1}(x^{(in)}, A_j, b_j)$
2 $r_j = A_j \cdot \bar{x}_j - b_j$
3 $b_{j+1} = R_j \cdot r_j$
4 $A_{j+1} = R_j \cdot A_j \cdot R_j^H$
5 $x_{j+1}^{(out)} = MGM(l-1, 0, A_{j+1}, b_{j+1}, \nu_1, \nu_2)$
6 $\hat{x}_j = \bar{x}_j - R_j^H \cdot x_{j+1}^{(out)}$
7 $x_j^{(out)} = SMOOTH^{\nu_2}(\hat{x}_j, A_j, b_j)$

In matrix notation, one iteration of the V-cycle is given by the matrix $MG_{levs,1}$, where

$$MG_{l,j} = S^{\nu_2}(I - R_j^H(I - MG_{l-1,j+1})A_{j+1}^{-1}R_j A_j)S^{\nu_1} \quad (3.1.11)$$

for $l > 1$, and $MG_{1,j}$ is the matrix with all zeros. If a W-cycle is used instead of a V-cycle, line **5** of the algorithm has to be replaced by

$$\begin{array}{l} \mathbf{5a} \\ \mathbf{5b} \end{array} \left| \begin{array}{l} x_{j+1}^{(int)} = MGM(l-1, 0, A_{j+1}, b_{j+1}, \nu_1, \nu_2) \\ x_{j+1}^{(out)} = MGM(l-1, x_{j+1}^{(int)}, A_{j+1}, b_{j+1}, \nu_1, \nu_2) \end{array} \right.$$

and (3.1.11) by

$$MG_{l,j} = S^{\nu_2}(I - R_j^H(I - MG_{l-1,j+1}^2)A_{j+1}^{-1}R_j A_j)S^{\nu_1} . \quad (3.1.12)$$

The convergence theory of Ruge and Stüben

As pointed out in Section 3.1.1, it is very difficult to derive a multigrid convergence theory. For algebraically oriented multigrid methods, such a theory has been developed by Ruge and Stüben [90]. Their convergence theory can not only be applied to multigrid methods with the quite specific restriction and coarse grid operators described above. It holds for all two-grid and multigrid methods which have the form of Algorithms 3 and 4. In other words, it is required that both the variational condition ($P_j = c \cdot R_j^H$) and the Galerkin condition ($A_{j+1} = R_j \cdot A_j \cdot R_j^H$) hold. In order to state the main theorem from [90], we need to define, on each level, the exact coarse grid correction in matrix notation:

$$CGC_j = I - R_j^H \cdot A_{j+1}^{-1} \cdot R_j \cdot A_j . \quad (3.1.13)$$

The following theorem contains the most general form of the convergence result. Conditions which are easier to verify in practical proofs will be derived in two corollaries.

Theorem 6 (AMG convergence, [90])

Let $A \in \mathbb{C}^{n \times n}$ be an Hermitian positive definite matrix and $b \in \mathbb{C}^n$. Suppose MGM is the V-cycle multigrid method defined in Algorithm 4, in matrix terms described by $MG_{levs,1}$ with $levs$ being the number of levels used in the multigrid method. Moreover, let $A_1 = A$, and let A_2, \dots, A_{levs} be the coarse grid matrices of size $n_2 > \dots > n_{levs}$, let R_1, \dots, R_{levs-1} be the restriction matrices, and S_1, \dots, S_{levs-1} the smoothing matrices. If there exists a $\delta > 0$ such that

$$\|S_j x\|_1^2 \leq \|x\|_1^2 - \delta \|CGC_j x\|_1^2 \quad \forall x \in \mathbb{C}^{n_j} \quad (3.1.14)$$

holds on each level j with δ independent of j , then $\delta > 0$ and

$$\|MG_{levs,1}\|_1 \leq \sqrt{1 - \delta} < 1. \quad (3.1.15)$$

Condition (3.1.14) is difficult to prove, because it contains both properties of the smoother and of the coarse grid correction. Therefore, it is split in two or three conditions which can be proved separately. In the first corollary, conditions for optimal convergence of the two-grid method are stated. In the two-grid case, the condition on the coarse grid correction can be further simplified.

Corollary 1 (Two-grid convergence, [90])

Let $A \in \mathbb{C}^{n \times n}$ be an Hermitian positive definite matrix and $b \in \mathbb{C}^n$. Suppose TGM is the two-grid method defined in Algorithm 3, in matrix terms described by TG . Assume that the smoother S satisfies the presmoothing property and the postsmoothing property, i.e. there exist two positive constants α_{pre} and α_{post} such that

$$\|S^{\nu_1} x\|_1^2 \leq \|x\|_1^2 - \alpha_{pre} \|S^{\nu_1} x\|_2^2, \quad \forall x \in \mathbb{C}^n, \quad (3.1.16)$$

$$\|S^{\nu_2} x\|_1^2 \leq \|x\|_1^2 - \alpha_{post} \|x\|_2^2, \quad \forall x \in \mathbb{C}^n. \quad (3.1.17)$$

Moreover, assume that the correcting condition is satisfied, i.e. there exists $\beta > 0$ such that

$$\min_{y \in \mathbb{R}^{n_C}} \|x - R^H y\|_0^2 \leq \beta \|x\|_1^2, \quad \forall x \in \mathbb{C}^n. \quad (3.1.18)$$

Then the two-grid method converges optimally. More precisely, $\beta > \alpha_{post}$, and the convergence factor of the two-grid method $\|TG\|_1$ is bounded by

$$\|TG\|_1 \leq \sqrt{\frac{1 - \alpha_{post}/\beta}{1 + \alpha_{pre}/\beta}}. \quad (3.1.19)$$

The second corollary gives conditions for the multigrid method. Whereas the smoothing conditions are the same as in Corollary 1, the correcting condition is more difficult to satisfy.

Corollary 2 (Multigrid convergence, [90])

Let $A \in \mathbb{C}^{n \times n}$ be an Hermitian positive definite matrix and $b \in \mathbb{C}^n$. Suppose MGM is the V-cycle multigrid method defined in Algorithm 4, in matrix terms described by $MG_{levs,1}$. Let the matrices A_j, R_j, S_j be defined as in Theorem 6. Assume that, for all $j \in \{1, \dots, levs - 1\}$, there exist $\alpha_j^{(pre)}, \alpha_j^{(post)} > 0$ such that

$$\|S_j^{\nu_1} x\|_1^2 \leq \|x\|_1^2 - \alpha_j^{(pre)} \|S^{\nu_1} x\|_2^2, \quad \forall x \in \mathbb{C}^{n_j}, \quad (3.1.20)$$

$$\|S_j^{\nu_2} x\|_1^2 \leq \|x\|_1^2 - \alpha_j^{(post)} \|x\|_2^2, \quad \forall x \in \mathbb{C}^{n_j}. \quad (3.1.21)$$

Furthermore, assume that the correcting condition is satisfied on each level, i.e. for all $j \in \{1, \dots, levs - 1\}$, there exists $\beta_j > 0$ such that

$$\|CGC_j x\|_1^2 \leq \beta_j \|x\|_1^2, \quad \forall x \in \mathbb{C}^{n_j}. \quad (3.1.22)$$

Then the V-cycle multigrid method converges optimally. More precisely, $\beta_j > \alpha_j^{(post)}$, and the convergence factor of the multigrid method $\|MG_{levs,1}\|_1$ is bounded by

$$\|MG_{levs,1}\|_1 \leq \sqrt{\frac{1 - \delta_{post}}{1 + \delta_{pre}}} \quad (3.1.23)$$

with $\delta_{pre} = \min_{1 \leq j \leq levs} \frac{\alpha_j^{(pre)}}{\beta_j}$ and $\delta_{post} = \min_{1 \leq j \leq levs} \frac{\alpha_j^{(post)}}{\beta_j}$.

The advantage of using the two corollaries instead of Theorem 6 is that the smoothing properties and the coarse grid correction property can be studied independently. The smoothing conditions depend only on the choice of the S_j and on the number of smoothing iterations ν_1 and ν_2 , whereas the correcting condition depends only on the choice of the restriction matrices R_j .

Remark 3 Theorem 6 still holds if $diag(A)$ in (3.1.9) is replaced by any Hermitian positive definite matrix Y . This fact is mentioned in [90] and much employed in [103, 3]. We will make use of this degree of freedom in several convergence proofs throughout this thesis.

Remark 4 Both corollaries also hold if only one type of smoothing is applied, i.e. either presmoothing or postsmoothing. In Corollary 1 and 2 this implies that either $\alpha_j^{(pre)}$ or $\alpha_j^{(post)}$ is zero for all j . In this work we will describe multigrid methods which use only postsmoothing and methods where both types of smoothing are applied.

3.1.3 Multigrid as a preconditioner

Multigrid methods are not only efficient solvers for linear systems of equations. They are also suitable as preconditioners for Krylov subspace methods. In this

section, we describe how the multigrid methods presented so far can be used as preconditioners. Furthermore, we present purely additive multigrid methods which are efficiently employed as preconditioners, but not as standalone solvers.

CG accelerated multigrid

The use of multigrid methods as a preconditioner for Krylov subspace methods is often referred to as *accelerating multigrid methods by Krylov subspace methods*. Since the matrices we are interested in in this work are positive definite, the accelerator will be the conjugate gradient method described in Chapter 2.4.1. A description of Krylov subspace methods with multigrid preconditioners can be found in [112, 61]. Formally, the problem is to find a preconditioner \tilde{A} for the solution of a linear system $Ax = b$ with a Krylov subspace method. If \tilde{A} is an approximation for A^{-1} , the solver applied to the preconditioned system

$$\tilde{A}Ax = \tilde{A}b \tag{3.1.24}$$

converges significantly faster. In each iteration of the pcg method, a linear system involving the matrix \tilde{A} must be solved. If multigrid is used as a preconditioner, one V-cycle or W-cycle iteration of the multigrid method is performed with initial guess zero. In most examples, this gives much better results than the standard one-level preconditioners such as Jacobi, Gauss-Seidel, or ILU. Another possible way of using multigrid as a preconditioner is known as *multigrid by iterant recombination*. From the first approximations $x^{(1)}, \dots, x^{(k)}$ an improved approximation is obtained by a residual minimization technique.

Having introduced the use of multigrid as a preconditioner, we ask the following question for practical applications: How can we decide which is more efficient, multigrid as a standalone solver or the accelerated version? There is no general rule, but the following guidelines can be given:

- If a standalone multigrid algorithm is very efficient for a class of problems, it is not useful to accelerate it with Krylov subspace methods. The additional computational effort is not worth to be made.
- Multigrid as a preconditioner for the cg method leads to a more robust algorithm. This is especially important for more involved problems, which contain either convection dominance or strong anisotropies, or even nonlinearities. In these cases, it is very difficult to choose the multigrid components in such a way as to obtain a robust standalone solver.
- Sometimes the combination of smoothing and coarse grid correction quickly reduces all components of the error, except for a few specific components. These are responsible for a slow overall convergence of the multigrid solver. This is where the pcg method is most effective, eliminating a small number of outliers with a very small number of iterations.

Domain decomposition methods

The multigrid preconditioners discussed so far are all *multiplicative*. This means that the residual is updated after each smoothing and coarse grid correction step of the preconditioner. *Additive* multigrid preconditioners, on the other hand, perform smoothing on different grids at once and add several corrections to the residual at the same time. This property makes them highly parallelizable, but leads to much slower convergence in the sequential case. Therefore, additive multigrid methods cannot be used as standalone solvers. They are nevertheless very useful as parallel preconditioners. Some of the most famous additive multilevel preconditioners fall into the category of domain decomposition methods. These are described in much detail in the book of Smith, Björstätt, and Gropp [106]. Domain decomposition methods are usually applied to linear systems arising from discretizing PDEs in the domain Ω . In this context, domain decomposition refers to subdividing the whole problem into smaller problems. Their solutions are then used to construct a preconditioner for the whole system. Combination of the domain decomposition idea for overlapping domains on the one hand and additive multigrid methods on the other leads to additive multilevel Schwarz methods. These consist of smoothers and coarse grid correction operators. On each grid the smoothers are applied to several, possibly overlapping subdomains. The coarse grid correction is computed with exact inversion of the coarse grid matrix in the two-grid case, whereas in the multigrid case the two-grid idea is applied recursively. In the following, we briefly introduce three additive multilevel Schwarz methods, which are described for example in [106].

The *multilevel diagonal scaling* preconditioner [46] is the Schwarz method which uses minimal size subdomains consisting of only one node. Thus the smoothing part is equivalent to the Jacobi smoother. The restriction matrix R is chosen for the problem at hand and the prolongation is given by R^H . Then the two-grid version of the diagonal scaling preconditioner is of the form

$$\tilde{A} = \text{diag}(A)^{-1} + R^H A_C^{-1} R \quad , \quad (3.1.25)$$

where the coarse grid matrix A_C is computed either with the Galerkin approach or obtained from rediscrretization on the coarse grid. A multilevel version of (3.1.25) is obtained by applying the two-grid idea recursively to approximate A_C . In this thesis, we will use the multilevel diagonal scaling preconditioner to illustrate that our multigrid methods for structured linear systems are not only suitable as standalone solvers, but also as preconditioners for Krylov subspace methods. Another famous Schwarz method is the *BPX* or *multilevel nodal basis* preconditioner [12]. It is very similar to the diagonal scaling preconditioner. Instead of the Jacobi smoother, which contains the inverse of the main diagonal of A , it uses a diagonal matrix whose entries are the size h_j of the elements of a finite element discretization on the respective grid. This is an approximation to the main diagonal of A which works very well for discrete elliptic PDEs. The *hierarchical basis* preconditioner

[117] can be derived from both the BPX and the diagonal scaling preconditioner. The main difference concerns coefficients which appear on the finest grid and also on some of the coarse grids. Instead of using these coefficients on each grid for smoothing, they are only treated once, on the coarsest grid on which they appear. The coarse grid correction is the same for all three methods.

Convergence results for additive Schwarz methods are presented in [44, 45, 116] and in the book [106]. These results are obtained in an abstract finite-element-based framework known as the Schwarz framework. A is interpreted as an operator a in a Hilbert space V , and both smoothing and coarse grid correction are considered to be subspace corrections computed in subspaces $V_j \subset V$. The condition number of the preconditioned system is essentially given by the product of three parameters. The first parameter states whether the subspaces V_j provide a stable splitting of V , the second parameter serves as a measure of the orthogonality of the subspaces, and the third parameter predicts how good the operators a_j in the subspaces approximate a . Convergence of the three Schwarz methods described above is proved in [46, 12, 48] by estimating the three parameters, assuming that a finite element discretization is used for the underlying PDEs. The diagonal scaling preconditioner is the one with the best convergence properties of the three.

3.2 Exploiting matrix structure for the development of multigrid methods

In the first part of this chapter, we have pointed out that multigrid methods belong to the most efficient and most popular solution techniques for linear systems of equations. On the other hand, we have outlined at the end of Chapter 2 that classical iterative solvers, such as the pcg method, face serious limitations when applied to ill-conditioned two-level Toeplitz systems. Therefore, beginning in the mid 1990s, multigrid methods have been developed for certain classes of structured matrices. In the following, we wish to give an introduction to these methods, which are essentially based on the strong connection between certain types of structured matrices and generating functions. Furthermore, we state criteria which allow us to judge whether a multigrid method, defined in terms of generating functions, is suitable for the solution of the corresponding structured linear systems.

3.2.1 Multigrid in terms of generating functions

One- and multilevel Toeplitz matrices as well as matrices belonging to a trigonometric algebra have been introduced in Chapter 2. In Section 2.1.3, a classification of these matrices has been given according to the zeros of the corresponding generating functions. For matrices of class **(I)**, i.e. for strictly positive generating functions, iterative solvers such as the pcg method are highly efficient. For all other matrix classes, they cannot be applied with the same efficiency. Especially

in the multilevel case, convergence deteriorates significantly (see Chapter 2.6). Therefore, multigrid methods have been developed which are especially designed for structured matrices. One main goal is to apply these methods to a greater variety of Toeplitz matrices than the pcg method, especially to matrices of class **(II)**, **(III)**, and **(IV)**. The methods developed so far can be used for matrices of class **(II)** and, with some restrictions, to matrices of class **(III)**, i.e. for functions with isolated zeros. Moreover, one hopes to obtain faster methods for Toeplitz matrices than the ones already available. For banded matrices belonging to the Toeplitz class or to a trigonometric algebra, the aim is to find an $O(n)$ solver. Multigrid methods for tau and Toeplitz systems were first developed by Fiorentino and Serra [52, 54]. Further work has been done in [99, 108, 24, 109, 72]. For recent results on circulant and tau matrices we refer to [103, 3], and for DCT-III matrices to [28, 33].

Multigrid methods for structured matrices make heavy use of the correspondence between matrices and generating functions. They are designed in an AMG-like fashion, because they rely more on the matrix structure than on an actual geometry. Therefore, all multigrid methods presented in this work are of the same form as Algorithm 3 and 4. All convergence proofs are based on the theory of Ruge and Stüben [90], which was summarized in Theorem 6 and its two corollaries. Following the classical AMG approach, the smoother is chosen to be a rather simple one. Most of the methods developed so far use the damped Jacobi method, some of them even the damped Richardson method. Prolongation and restriction as well as computation of the coarse grid matrices are described in terms of generating functions. The given matrix $A_1 = A_{\mathbf{n}}$ corresponds to the generating function $f_1 = f$. If the coarse grid matrices A_2, A_3, A_4 , etc. are still in the same matrix class, the corresponding generating functions is denoted f_2, f_3, f_4 , etc., respectively. The restriction matrices R_1, R_2, R_3 , etc. are of the form

$$R_j = B_j \cdot E_j \quad . \quad (3.2.1)$$

B_j is a matrix in the same class as A_j , corresponding to a function b_j , which is defined to deal with the zeros of f_j . More precisely, since the matrices A_j are coarse grid representations of A_1 , the functions f_j should be coarse grid representations of f with the same number of zeros of the same order. E_j is an elementary restriction matrix of the respective matrix class. The matrix A_{j+1} on the next coarser level is computed with the Galerkin approach

$$A_{j+1} = R_j^H A_j R_j = E_j^H (B_j^H A_j B_j) E_j \quad . \quad (3.2.2)$$

Depending on the respective matrix class, (3.2.2) will be translated to generating functions. This correspondence between matrices and functions enables us to develop multigrid methods solely from looking at the zeros of the corresponding generating functions. Moreover, rigorous convergence proofs can be given for many problems where proofs would not be possible without generating functions. Later, we will also discuss the use of multigrid as a preconditioner for the pcg method.

Since multigrid methods for structured matrices are constructed in terms of generating functions, the requirements on the choice of projection/restriction matrices and on computation of coarse grid matrices are stated in terms of generating functions first. But then, what special properties is a good multigrid method for structured linear systems expected to have? Following the presentation in [3], we state three main criteria for the assessment of a multigrid method. If they are all satisfied, we can expect our method to work fast and efficiently.

3.2.2 Criteria for an efficient multigrid method

In the following, we describe three properties of multigrid methods which are desirable for the solution of structured linear systems $A_{\mathbf{n}}x = b$ with $A_{\mathbf{n}}$ corresponding to a generating function $f(\mathbf{x})$. They are stated in [3, 38] and concern algebraic, computational, and convergence-related issues. If all three criteria are satisfied, the multigrid method is optimal in the sense of Axelsson and Neytcheva [4]. This means the problem of solving a linear system with matrix $A_{\mathbf{n}}$ has the same asymptotic cost as one matrix-vector multiplication involving $A_{\mathbf{n}}$. In subsequent chapters, we try to satisfy as many of the criteria as possible when we design our own multigrid methods.

- **Algebraic criterion:**

All coarse grid matrices obtained in the multigrid procedure should be in the same matrix class as $A_1 = A_{\mathbf{n}}$. For example, if A_1 is a two-level tau matrix, then the matrices A_2 , A_3 , and so on must also be two-level tau matrices, just of smaller size. This criterion is a purely algebraic one. It does not state anything about convergence or optimality. However, it is important for a multigrid method to preserve the relationship between matrices and generating functions on coarser levels. Only if the algebraic criterion is satisfied, we can define a multigrid method which fully exploits the matrix structure.

- **Computational criterion:**

The computational cost of each iteration should be optimal. More precisely, the cost of one multigrid iteration is of the same order as the cost of one matrix-vector product involving $A_{\mathbf{n}}$. This means that for banded matrices, the total cost of one multigrid iteration should be $O(\mathbf{n})$, and for dense matrices $O(\mathbf{n} \log(\mathbf{n}))$. In more detail, the computational criterion states that the following operations should be carried out in at most $O(\mathbf{n})$ flops for banded matrices and in at most $O(\mathbf{n} \log(\mathbf{n}))$ flops for full matrices: matrix-vector products involving A_i, R_i, R_i^H , computation of coarse grid matrices A_i , smoothing operations, exact solution of the linear system on the coarsest grid.

- **Convergence criterion:** In each multigrid iteration, the error reduction must be uniformly bounded by a constant smaller than 1, independent of

the matrix size \mathbf{n} . This guarantees that the number of multigrid iterations to reach convergence is bounded by a constant, independent of \mathbf{n} . This criterion is verified by proving that Corollaries 1 and 2 and therefore the convergence theorem of Ruge and Stüben is satisfied.

3.3 Multigrid for trigonometric matrix algebras

In this section, we wish to give an overview of multigrid methods for trigonometric matrix algebras. All these methods have been developed for matrices corresponding to generating functions with isolated zeros of finite order. The methods benefit from the fact that in a matrix algebra the product of two matrices is still in the algebra. Since Toeplitz matrices do not share this property, there will be additional difficulties with respect to this fact. In the first part of this section, we focus on circulant and tau matrices. A description of the main components of the multigrid algorithms is followed by a proof that the criteria stated in Section 3.2.2 are indeed satisfied. The second part of this section is devoted to the DCT-III and DST-III algebras. First, we review a well-known multigrid method for the DCT-III algebra, and analyze it from the point of view of the three criteria. Then, we carry over the main ideas to the related DST-III algebra, for which similar convergence and optimality results are obtained.

3.3.1 Circulant and tau matrices

Fiorentino and Serra developed the first multigrid methods for the structured matrix classes which shall be analyzed in this work. In [52, 54], they present a method for one- and two-level tau matrices. In [100], the second author gave a two-grid convergence proof for multilevel tau matrices. For circulant matrices, a similar multigrid algorithm was presented in [102], and two-grid convergence for one- and multilevel circulant matrices was proved in [103]. In [3], a V-cycle multigrid convergence proof for circulant and tau matrices is given under slightly stronger conditions. It is extended to the multilevel case in [2]. All these methods are based on the algebraically oriented multigrid approach, i.e. they are of the same form as Algorithms 3 and 4. In the following, we describe how the individual components of the multigrid methods are chosen and how the conditions stated in Corollaries 1 and 2 are satisfied. We use the same notation as in Algorithms 3 and 4.

Let us start with the multigrid method for one-level circulant and tau matrices, and then generalize it to the multilevel case. As for most of their multigrid algorithms, the authors choose the damped Richardson method as the smoother S_j on all levels $1 \leq j \leq \text{levs}$. To compute the coarse-grid correction, a restriction matrix R_j according to (3.2.1) must be defined on each level. This means B_j and E_j are chosen such that A_{j+1} is about half the size of A_j , and that the zeros of the coarse-grid function f_{j+1} correspond to the zeros of f_j . Computation of A_{j+1}

in (3.2.2) is translated to generating functions in two steps. The product

$$\hat{A}_j = B_j^H A_j B_j \quad (3.3.1)$$

in (3.2.2) is in the same algebra as A_j , and therefore corresponds to the product of functions

$$\hat{f}_j(x) = f_j(x) \cdot b_j^2(x) \quad (3.3.2)$$

The algebraic criterion is only satisfied if E_j is chosen such that the matrix A_{j+1} is in the same algebra as A_j , and has about half the size of A_j . To preserve the circulant structure on the coarser level, the size of A_{j+1} is $n_{j+1} = \frac{n_j}{2}$, whereas for tau matrices it is $n_{j+1} = \frac{n-1}{2}$. $E_j^{(circ)}$ is an n_{j+1} -by- n_j matrix which is given by

$$(E_j^{(circ)})^H = \begin{pmatrix} 1 & 0 & & & & & & & \\ & 1 & 0 & & & & & & \\ & & \ddots & \ddots & & & & & \\ & & & & \ddots & \ddots & & & \\ & & & & & & 1 & 0 & \\ & & & & & & & & \end{pmatrix}, \quad (3.3.3)$$

whereas $E_j^{(tau)}$ is of the form

$$(E_j^{(tau)})^H = \begin{pmatrix} 0 & 1 & 0 & & & & & & \\ & 0 & 1 & 0 & & & & & \\ & & \ddots & \ddots & \ddots & & & & \\ & & & & & & 0 & 1 & 0 \\ & & & & & & & & \end{pmatrix}. \quad (3.3.4)$$

If we use the notation E_j , this means that the matrix belongs to either of the matrix algebras. With this choice of E_j , $A_{j+1} = E_j^H \hat{A}_j E_j$ is obtained by picking every second row and every second column of \hat{A}_j . In terms of generating functions, this means f_{j+1} is obtained by picking every second Fourier coefficient of \hat{f}_j . For circulant matrices, this is achieved with

$$\begin{aligned} f_{j+1}(x) &= \frac{1}{2} \left(\hat{f}_j\left(\frac{x}{2}\right) + \hat{f}_j\left(\frac{x}{2} + \pi\right) \right) \\ &= \frac{1}{2} \left(f_j\left(\frac{x}{2}\right) b_j^2\left(\frac{x}{2}\right) + f_j\left(\frac{x}{2} + \pi\right) b_j^2\left(\frac{x}{2} + \pi\right) \right). \end{aligned} \quad (3.3.5)$$

For tau matrices, $x/2 + \pi$ is replaced with $\pi - x/2$. It should be mentioned that application of E_j to the orthogonal transform matrices $Q_{n_j}^{(circ)}$ or $Q_{n_j}^{(tau)}$ produces two transform matrices $Q_{n_{j+1}}^{(circ)}$ or $Q_{n_{j+1}}^{(tau)}$ of smaller size. More precisely, we obtain

$$(E_j^{(circ)})^H \cdot Q_{n_j}^{(circ)} = \left(Q_{n_{j+1}}^{(circ)} | Q_{n_{j+1}}^{(circ)} \right) \quad (3.3.6)$$

for circulant matrices and

$$(E_j^{(tau)})^H \cdot Q_{n_j}^{(tau)} = \left(Q_{n_{j+1}}^{(tau)} | 0_{n_{j+1}} | D_{n_{j+1}} Q_{n_{j+1}}^{(tau)} \right) \quad (3.3.7)$$

for tau matrices, where $D_{n_{j+1}} = \text{diag}(d_0, \dots, d_{n_{j+1}-1})$ with $d_k = (-1)^k$.

At the very heart of the algorithm is the choice of B_j corresponding to a function $b_j(x)$. The first idea was the following: If $f_j(x)$ has a zero at x_0 of order ν_0 , then $b_j(x)$ must be strictly positive at x_0 , and it must have a zero of order ν_0 at $\pi + x_0$ in the circulant case and at $\pi - x_0$ in the tau case. Then, it can be shown that f_{j+1} has a zero of order ν_0 at $2x_0$. The zero of b_j is called a mirror point. This choice of b_j has been refined and described in more detail in [102] and [99]. b_j is chosen to satisfy the following conditions for circulant and tau matrices, respectively. Let us assume that f_j has a single isolated zero x_0 in $] -\pi, \pi]$ in the circulant case or a pair of zeros $-x_0, x_0$ in the tau case. Then, for circulant matrices, we choose

$$b_j(x) \sim |x - (\pi + x_0)|^{2\lceil\beta/2\rceil} \quad \text{over }] -\pi, \pi] , \quad (3.3.8)$$

where

$$\beta = \underset{k}{\text{argmin}} \left(\lim_{x \rightarrow x_0} \frac{(x - x_0)^{2k}}{f_j(x)} < \infty \right) \quad (3.3.9)$$

and

$$0 < b_j^2(x) + b_j^2(\pi + x) . \quad (3.3.10)$$

(3.3.8) and (3.3.9) can be generalized to the condition

$$\limsup_{x \rightarrow x_0} \frac{b_j^2(\pi + x)}{f_j(x)} < \infty , \quad (3.3.11)$$

which gives the minimum order the zero of $b_j(x)$ must have at the mirror point. Condition (3.3.10) states that $b_j(x)$ must be nonzero at x_0 . A possible choice of b_j satisfying the conditions is

$$b_j(x) = (2 - 2 \cos(x - (\pi + x_0)))^{\beta/2} . \quad (3.3.12)$$

For tau matrices, the conditions are very similar. (3.3.8) is replaced by

$$b_j(x) \sim |x - (\pi - x_0)|^{2\lceil\beta/2\rceil} \quad \text{over }]0, \pi] , \quad (3.3.13)$$

whereas (3.3.9) holds unchanged, leading to the more general

$$\limsup_{x \rightarrow x_0} \frac{b_j^2(\pi - x)}{f_j(x)} < \infty . \quad (3.3.14)$$

(3.3.10) is changed to

$$0 < b_j^2(x) + b_j^2(\pi - x) . \quad (3.3.15)$$

Since b_j must be an even function, it can, for example, be chosen

$$b_j(x) = (\cos(x_0) + \cos(x))^{\beta} . \quad (3.3.16)$$

For d -level circulant and tau matrices, the elementary restriction matrices are chosen to be the Kronecker products of (3.3.3) and (3.3.4), i.e.

$$E_j = E_{j_1} \otimes \cdots \otimes E_{j_d} . \quad (3.3.17)$$

Restriction in terms of generating functions is done with the d -dimensional analogs of (3.3.2) and (3.3.5). In the two-level case, these are

$$\hat{f}_j(x, y) = f_j(x, y) \cdot b_j^2(x, y) \quad (3.3.18)$$

and

$$f_{j+1}(x, y) = \frac{1}{4} \left(\hat{f}_j\left(\frac{x}{2}, \frac{y}{2}\right) + \hat{f}_j\left(\frac{x}{2} + \pi, \frac{y}{2} + \frac{x}{2}, \frac{y}{2} + \pi\right) + \hat{f}_j\left(\frac{x}{2} + \pi, \frac{y}{2} + \pi\right) \right) \quad (3.3.19)$$

or the tau analogs. Frequently, we use the notation $\mathbf{x} = (x, y)$ or $\mathbf{x}_0 = (x_0, y_0)$.

The function $b_j(\mathbf{x})$ must satisfy the d -level version of conditions (3.3.8)-(3.3.16). For each zero of f_j , there is not just one, but $2^d - 1$ mirror points, where b_j must be zero. For example in two dimensions, a zero of $f(\mathbf{x})$ at (\mathbf{x}_0) requires $b(\mathbf{x})$ to be zero at the three mirror points, which are contained in the set

$$M(\mathbf{x}_0) = \{(\pi + x_0, y_0), (x_0, \pi + y_0), (\pi + x_0, \pi + y_0)\} \quad (3.3.20)$$

for two-level circulant matrices and

$$M(\mathbf{x}_0) = \{(\pi - x_0, y_0), (x_0, \pi - y_0), (\pi - x_0, \pi - y_0)\} \quad (3.3.21)$$

for two-level tau matrices. The multilevel conditions on b_j are the following for circulant and tau matrices:

$$\limsup_{\mathbf{x} \rightarrow \mathbf{x}_0} \frac{b_j^2(\mathbf{y})}{f_j(\mathbf{x})} < \infty \text{ for } \mathbf{y} \in M(\mathbf{x}) \quad (3.3.22)$$

$$\sum_{\mathbf{y} \in M(\mathbf{x}) \cup \{\mathbf{x}\}} b_j^2(\mathbf{y}) > 0 . \quad (3.3.23)$$

With these definitions of E_j and B_j , the following theoretical results have been obtained in [3] and [2]:

- The algebraic condition is satisfied, i.e. on each level the matrix A_j is the circulant or tau matrix generated by f_j . Moreover, a zero of f at \mathbf{x}_0 implies a zero of f_2 at $2\mathbf{x}_0$.
- If $A_{\mathbf{n}}$ is banded, i.e. if f is a trigonometric polynomial of small degree, then the b_j can be chosen such that all coarse grid functions f_j are also polynomials of small degree. Thus, there exists a constant C such that each matrix-vector product involving A_j is computed in less than $C \cdot n_j$ operations. The computational condition holds.

- The two-grid version of the Ruge-Stüben-Theorem holds, i.e. conditions (3.1.16)-(3.1.18) in Corollary 1 are proved to hold if (3.3.22) and (3.3.23) are satisfied. The convergence rate $\sqrt{\frac{1-\alpha_{post}/\beta}{1+\alpha_{pre}/\beta}}$ is independent of the recursion level. This property, which is called level independency, is necessary, but not sufficient for optimal convergence of a V-cycle multigrid method. A simple example for non-optimal convergence is given in [3]. The tau matrix corresponding to the function

$$f(x) = (2 - 2 \cos(x))^2$$

is solved with a multigrid method, where on each level, the prolongation matrix is related to

$$b(x) = 2 + 2 \cos(x) .$$

Then, (3.3.14) and (3.3.15) are satisfied, but the V-cycle multigrid method does not converge optimally.

The above conditions suffice to ensure two-grid convergence and also convergence of the W-cycle, but they are not strong enough to guarantee optimal multigrid convergence in the V-cycle. This can be achieved by using functions b_j with zeros of higher order at the mirror points, and therefore slightly denser matrices B_j . In [3, 2], the following condition is proposed instead:

$$\limsup_{\mathbf{x} \rightarrow \mathbf{x}_0} \left\| \frac{b_j(\mathbf{y})}{f_j(\mathbf{x})} \right\| < \infty \text{ for } \mathbf{y} \in M(\mathbf{x}) . \quad (3.3.24)$$

With (3.3.24) and (3.3.23), the multigrid correcting condition (3.1.22), and therefore Corollary 2, is proved. Hence, also the convergence criterion is satisfied and the multigrid method is optimal in the sense of Axelsson and Neytcheva.

3.3.2 DCT-III and DST-III algebras

Multigrid methods for DCT-III matrices have been introduced in [33, 28]. In these articles, convergence has been proved for the two-grid method, but in the numerical experiments, also the V-cycle multigrid method shows optimal convergence. We review the main results and then carry them over to the related DST-III algebra.

DCT-III matrices

Again, the smoother is chosen to be the damped Richardson or Jacobi iteration. We start with the one-level DCT-III algebra and present an extension to the two-level algebra later. Having in mind the three criteria for an efficient multigrid method, the restriction matrices $R_j = B_j \cdot E_j$ are defined as follows. In order to

satisfy the algebraic requirement, the elementary restriction matrix of size n_j -by- n_{j+1} with $n_{j+1} = n_j/2$ is chosen to be

$$(E_j^{(dct3)})^H = \begin{pmatrix} 1 & 1 & & & \\ & & 1 & 1 & \\ & & & & \ddots & \ddots \\ & & & & & & 1 & 1 \end{pmatrix}. \quad (3.3.25)$$

Again, $(E_j^{(dct3)})^H$ represents the spectral link between the space of frequencies on the fine and on the coarse grid. In [28], it is shown that applying $(E_j^{(dct3)})^H$ to the matrix $Q_{n_j}^{(dct3)}$ leads to the DCT-III equivalent of equations (3.3.6) and (3.3.7):

$$(E_j^{(dct3)})^H \cdot Q_{n_j}^{(dct3)} = \left(Q_{n_{j+1}}^{(dct3)} D_{j+1}^{(1)} | Q_{n_{j+1}}^{(dct3)} D_{j+1}^{(2)} \Pi_{j+1} \right) \quad (3.3.26)$$

with

$$D_{j+1}^{(1)} = \text{diag}_{l=1, \dots, n_{j+1}} \left[\sqrt{2} \cos \left(\frac{(l-1)\pi}{4n_{j+1}} \right) \right], \quad D_{j+1}^{(2)} = \text{diag}_{l=1, \dots, n_{j+1}} \left[-\sqrt{2} \sin \left(\frac{(l-1)\pi}{4n_{j+1}} \right) \right],$$

The permutation matrix Π_{j+1} corresponds to the permutation

$$\pi_{j+1} : (1, 2, \dots, n_{j+1}) \longrightarrow (1, n_{j+1}, n_{j+1} - 1, \dots, 2).$$

With (3.3.26), it is shown that

$$\begin{aligned} A_{j+1} &= R_j^H A_j R_j = (E_j^{(dct3)})^H B_j^H A_j B_j E_j^{(dct3)} \\ &= (E_j^{(dct3)})^H Q_{n_j}^{(dct3)} \Delta_{n_j} (f_j b_j^2) (Q_{n_j}^{(dct3)})^H E_j^{(dct3)} \\ &= Q_{n_{j+1}}^{(dct3)} [D_{j+1}^{(1)}, D_{j+1}^{(2)} \Pi_{j+1}] \Delta_{n_j} (f_j b_j^2) \begin{pmatrix} D_{j+1}^{(1)} \\ \Pi_{j+1}^H D_{j+1}^{(2)} \end{pmatrix} (Q_{n_{j+1}}^{(dct3)})^H \end{aligned} \quad (3.3.27)$$

with $\Delta_{n_j}(f) = \text{diag}_{0 \leq j \leq n_j-1} [f(j\pi/n_j)]$. Hence, A_{j+1} corresponds to the function $f_{j+1}(x)$, which is obtained with

$$f_{j+1}(x) = 2[\cos^2(\frac{x}{4})f_j(\frac{x}{2})b_j(\frac{x}{2}) + \sin^2(\frac{x}{4})f_j(\pi - \frac{x}{2})b_j(\pi - \frac{x}{2})]. \quad (3.3.28)$$

The choice of the functions $b_j(x)$ is done in a similar way as in (3.3.13)-(3.3.15). If f_j has a unique zero $x_0 \in [0, \pi]$, then b_j must satisfy (3.3.15) and, in addition,

$$b_j(x) = [2 - 2 \cos(x - (\pi - x_0))]^{[\beta/2]} \sim |x - (\pi - x_0)|^{2[\beta/2]} \text{ over }]0, \pi], \quad (3.3.29)$$

where

$$\beta \geq \underset{k}{\text{argmin}} \left(\lim_{x \rightarrow x_0} \frac{\sin^2(x/2)}{\cos^2(x/2)} \cdot \frac{|x - x_0|^{2k}}{f_j(x)} < \infty \right). \quad (3.3.30)$$

Remark 5 With conditions (3.3.29) and (3.3.30), the following properties of the two-grid method are shown in [33, 28]:

- If x_0 is a zero of f_j , then $2x_0$ is a zero of f_{j+1} which is of the same order as x_0 . The only exception is $x_0 = \pi$, where the order of $2x_0$ is the order of x_0 plus 2. However, a zero of f at $x_0 = \pi$ leads to a zero of f_2 at the origin. Therefore, this additional increase of the order happens only on the finest grid.
- If f is a trigonometric polynomial of fixed degree, then f_2 (and therefore all f_j) are polynomials of fixed degree, only depending on the orders of the zeros. Thus, the computational complexity is not increased on coarser levels.
- The postsmoothing condition (3.1.17) and the correcting condition (3.1.18) are satisfied, and the two-grid method converges optimally.

These properties are necessary conditions for optimality in the sense of Axelsson and Neytcheva. However, V-cycle convergence has not yet been proved.

For two-level DCT-III matrices, the elementary restriction matrix is the Kronecker product of the matrices $(E_j^{(dct3)})^H$ from (3.3.31), i.e.

$$(E_{\mathbf{j}}^{(dct3)})^H = (E_{j_1}^{(dct3)})^H \otimes (E_{j_2}^{(dct3)})^H \quad (3.3.31)$$

With the two-dimensional analogue of $Q_{n_j}^{(dct3)}$, being defined as

$$Q_{\mathbf{n}_j}^{(dct3)} = Q_{n_{j_1}}^{(dct3)} \otimes Q_{n_{j_2}}^{(dct3)}, \quad (3.3.32)$$

(3.3.27) and (3.3.26) are directly carried over to the two-dimensional case with Kronecker products. In terms of generating functions, (3.3.28) is replaced by

$$\begin{aligned} f_{j+1}(x, y) = & 4[\cos^2\left(\frac{x}{4}\right)\cos^2\left(\frac{y}{4}\right)\hat{f}_j\left(\frac{x}{2}, \frac{y}{2}\right) + \sin^2\left(\frac{x}{4}\right)\cos^2\left(\frac{y}{4}\right)\hat{f}_j\left(\pi - \frac{x}{2}, \frac{y}{2}\right) \\ & + \cos^2\left(\frac{x}{4}\right)\sin^2\left(\frac{y}{4}\right)\hat{f}_j\left(\frac{x}{2}, \pi - \frac{y}{2}\right) + \sin^2\left(\frac{x}{4}\right)\sin^2\left(\frac{y}{4}\right)\hat{f}_j\left(\pi - \frac{x}{2}, \pi - \frac{y}{2}\right)] , \end{aligned} \quad (3.3.33)$$

with $\hat{f}_j(x, y) = f_j(x, y) \cdot b_j^2(x, y)$. $b_j(x, y)$ is chosen in analogy to (3.3.29) and (3.3.30):

$$b_j(\mathbf{x}) \sim \prod_{\hat{\mathbf{x}}_0 \in M(\mathbf{x})} \|\mathbf{x} - \hat{\mathbf{x}}_0\|_1^{2\lceil \beta/2 \rceil} \quad \text{over }]0, \pi]^2 \quad (3.3.34)$$

with

$$\beta \geq \operatorname{argmin}_k \left(\lim_{x \rightarrow x_0} \frac{\sin^2(x/2)}{\cos^2(x/2)} \cdot \frac{|x - x_0|^{2k}}{f_j(x)} + \lim_{y \rightarrow y_0} \frac{\sin^2(y/2)}{\cos^2(y/2)} \cdot \frac{|y - y_0|^{2k}}{f_j(y)} < \infty \right). \quad (3.3.35)$$

If b_j satisfies these two conditions, the same facts as in Remark 5 hold.

DST-III matrices

In the following, we define the components of a multigrid method for the related DST-III algebra. In the one-level case, we choose $n_{j+1} = n_j/2$ and

$$E_j^{(dst3)} = E_j^{(dct3)} \quad . \quad (3.3.36)$$

with $E_j^{(dst3)}$ from (3.3.25). The DST-III equivalent of (3.3.26) is

$$(E_j^{(dst3)})^H \cdot Q_{n_j}^{(dst3)} = \left(Q_{n_{j+1}}^{(dst3)} \tilde{D}_{j+1}^{(1)} | Q_{n_{j+1}}^{(dst3)} \tilde{\Pi}_{j+1}^{(2)} \tilde{D}_{j+1}^{(2)} \tilde{\Pi}_{j+1}^{(1)} \right) \quad (3.3.37)$$

with

$$\tilde{D}_{j+1}^{(1)} = \text{diag}_{l=1, \dots, n_{j+1}} \left[\sqrt{2} \cos \left(\frac{l\pi}{4n_{j+1}} \right) \right], \quad \tilde{D}_{j+1}^{(2)} = \text{diag}_{l=1, \dots, n_{j+1}} \left[\sqrt{2} \sin \left(\frac{(l-1)\pi}{4n_{j+1}} \right) \right],$$

The permutation matrix $\tilde{\Pi}_{j+1}^{(1)}$ corresponds to the permutation

$$\pi_{j+1}^{(1)} : (1, 2, \dots, n_{j+1}) \longrightarrow (n_{j+1}, n_{j+1} - 1, \dots, 1),$$

whereas $\tilde{\Pi}_{j+1}^{(2)}$ corresponds to

$$\pi_{j+1}^{(2)} : (1, 2, \dots, n_{j+1}) \longrightarrow (2, 3, \dots, n_{j+1}, 1).$$

Hence, we can compute

$$A_{j+1} = Q_{n_{j+1}}^{(dst3)} [\tilde{D}_{j+1}^{(1)}, \tilde{\Pi}_{j+1}^{(2)} \tilde{D}_{j+1}^{(2)} \tilde{\Pi}_{j+1}^{(1)}] \Delta_{n_j} (f_j b_j^2) \left(\begin{array}{c} \tilde{D}_{j+1}^{(1)} \\ (\tilde{\Pi}_{j+1}^{(1)})^H \tilde{D}_{j+1}^{(2)} (\tilde{\Pi}_{j+1}^{(2)})^H \end{array} \right) (Q_{n_{j+1}}^{(dct3)})^H, \quad (3.3.38)$$

which implies that $f_{j+1}(x, y)$ is computed with (3.3.28). We choose $b_j(x)$, which corresponds to the DST-III matrices B_j , such that (3.3.15), (3.3.29), and (3.3.30) are satisfied. Since the multigrid method is similar as in the DCT-III case, the following properties can be stated.

Theorem 7

Let $S_{\mathbf{n}}[f]$ be the DST-III matrix of size \mathbf{n} corresponding to a trigonometric polynomial f with an isolated zero. If $R_j = B_j \cdot E_j^{(dst3)}$ with $E_j^{(dst3)}$ and $b_j(x)$ is chosen as above, the following holds:

1. If $x_0 \neq \pi$ is a zero of $f_j(x)$, then $2x_0$ is a zero of $f_{j+1}(x)$ of the same order.
If $x_0 = \pi$, then the order is increased by 2.
2. If $f(x)$ is a trigonometric polynomial of fixed degree, then this is also true for all $f_j(x)$.

3. For the two-grid method, the postsmoothing condition (3.1.17) and the correcting condition (3.1.18) are satisfied, and the two-grid method converges.

Proof: The proof of 1. and 2. is the same as in the articles [33, 28]. For the postsmoothing condition, we show that

$$(S_n[f])_{j,j} = \sum_{k=0}^{n-1} f(k\pi/n)(Q_n^{(dst)})_{j,k}^2 > 0 \quad (0 \leq j \leq n-1).$$

Hence, $\hat{a} = \min_j (S_n[f])_{j,j} > 0$. Then, (3.1.17) follows as in [28]. The proof of the correcting condition is similar to the DCT-III case. Following the proof in [28], (3.1.18) is proved if the following inequality holds:

$$I - R^H [RR^H]^{-1} R \leq \frac{\gamma}{\hat{a}} S_n[f]. \quad (3.3.39)$$

We carry out a block diagonalization of (3.3.39) by multiplying both sides with the two-dimensional Fourier matrix $Q_n^{(circ)}$ from the right and with $(Q_n^{(dst3)})^H$ from the left. On the right hand side, this results in a diagonal matrix containing the eigenvalues of $S_n[f]$, i.e. the values of f at the grid points. On the left-hand side, we can permute

$$(Q_n^{(dst3)})^H I - R^H [RR^H]^{-1} R Q_n^{(circ)}$$

into a 2-by-2 block diagonal matrix. With this decomposition, we follow the proof of [28]. Due to the continuity of $f(x)$ and $b(x)$, we prove (3.3.39) by showing that

$$I_2 - \frac{1}{\|b[x]\|_2^2} b[x](b[x])^T \leq \frac{\gamma}{\hat{a}} \text{diag}(f[x]) \quad (3.3.40)$$

with $b[x] = (\cos(x/2)b(x), \sin(x/2)b(\pi-x))^T$ and $f[x] = (f(x), f(\pi-x))^T$. This is true if all entries of the 2-by-2 matrix

$$R(x) = \text{diag}^{-1/2}(f[x]) \left(I_2 - \frac{1}{\|b[x]\|_2^2} b[x](b[x])^T \right) \text{diag}^{-1/2}(f[x]) \quad (3.3.41)$$

are uniformly bounded. For $j \neq k$, we have

$$[R(x)]_{j,k} = \frac{\cos(x/2) \sin(x/2)}{\cos^2(x/2)b^2(x) + \sin^2(x/2)b^2(\pi-x)} \frac{b(x)b(\pi-x)}{\sqrt{f(x)f(\pi-x)}}.$$

Hence, the matrix $R(x)$ has almost the same entries as in [28], and all entries are uniformly bounded. Thus, the correcting condition holds. \blacksquare

In the two-dimensional case, the matrices $E_j^{(dst3)}$ and $Q_{\mathbf{n}_j}^{(dst3)}$ are the Kronecker products of the one-level matrices, similar to their DCT-III counterparts in (3.3.31) and (3.3.32). $f_{j+1}(\mathbf{x})$ is computed with (3.3.33) and $b_j(\mathbf{x})$ is chosen such that (3.3.34) and (3.3.35) are satisfied.

Remark 6 With E_j and b_j defined as above, the same results as in Theorem 7 can be proved for the two-level case. The proof of the smoothing and correcting condition are carried out with tensor arguments.

3.4 Multigrid for Toeplitz systems

Unlike matrices from trigonometric algebras, Toeplitz matrices do not have the nice property that the product of two Toeplitz matrices is again Toeplitz. Therefore, it is significantly more difficult to develop multigrid methods for Toeplitz matrices, and especially to carry out convergence proofs. However, in recent years, several articles have been published on multigrid methods for Toeplitz systems. Based on the first methods proposed by Fiorentino and Serra [52, 54], the methods are all very similar to the ones for trigonometric matrix algebras presented in Section 3.3. In this section, we give an overview of the well-known results on multigrid methods for Toeplitz systems. We start with convergence proofs for functions with zeros of order 2. Then, we present more general convergence results, which were obtained for an adaptation of the methods presented for tau matrices. Finally, we describe a slightly different method, which is based on rediscrretization.

3.4.1 A method for zeros of order two

In [108, 24], R. Chan et. al. gave the first multigrid convergence proof for Toeplitz systems corresponding to functions with a single isolated zero of order 2 in $]-\pi, \pi[$. In [109], their results are extended to BTTB systems. These articles are based on the multigrid methods developed in [52, 54], which are stated not only for tau, but also for Toeplitz matrices. The smoother is a simple one, either the damped Richardson or the damped Jacobi method. Under the assumption that the zero of $f(x, y)$ is located at the origin, i.e. that $f(x, y)$ satisfies the condition

$$\min_{(x,y) \in [-\pi, \pi]^2} \frac{f(x, y)}{2 - \cos(x) - \cos(y)} > 0, \quad (3.4.1)$$

the restriction matrix is again chosen to be $R = B \cdot E$. B is the BTTB matrix corresponding to

$$b(x, y) = 2 - \cos(x) - \cos(y), \quad (3.4.2)$$

and E is the elementary restriction matrix defined in (3.3.4). Convergence of the two-grid method is proved by verifying the conditions of Corollary 1. Only postsmoothing is used in this method. The smoothing condition for Toeplitz matrices is already proved in [90], the correcting condition, and therefore two-grid convergence, is proved in [109].

Under the assumption that the size of the matrix $T_{n_1 n_2}$ is of the form $n_1 = 2^{k_1} - 1$ and $n_2 = 2^{k_2} - 1$, all matrices on coarser grids are again BTTB. For this case, level-independency of the multigrid method has been proved. Following the presentation in [112], this implies optimal convergence of the W-cycle, but not of the V-cycle. The numerical results, however, suggest that also a multigrid method with V-cycles converges optimally. In Chapter 4, we will use the proof technique from [109] to obtain convergence results for anisotropic BTTB systems.

3.4.2 Adapting the matrix algebra method to the Toeplitz class

In the original articles of Fiorentino and Serra [52, 54] as well as in [99, 3], multigrid methods for Toeplitz matrices are tightly connected to the methods for tau matrices described in the previous section. For tau matrices, computation of the coarse grid matrices with (3.2.2) was directly translated to generating functions in (3.3.2) and (3.3.5). In the Toeplitz class, this analogy can be used as a heuristic, but there is no one-to-one correspondence between matrices and functions. Even if A_j is Toeplitz, neither \hat{A}_j nor A_{j+1} is Toeplitz in general. In the one-level Toeplitz class, \hat{A}_j and A_{j+1} are Toeplitz plus low-rank matrices, but for BTTB matrices, low-rank means $O(\sqrt{\mathbf{n}})$. Nevertheless, the prolongation Fiorentino and Serra use for Toeplitz matrices is very similar to the prolongation they use for tau matrices. More precisely, E_j is the same as in (3.3.4), and B_j is either the Toeplitz or the tau matrix corresponding to the function $b_j(\mathbf{x})$, which is chosen as in the tau case. If $B_j = \tau(b_j)$ with b_j satisfying (3.3.22) and (3.3.23), Serra [99] proved conditions (3.1.17) and (3.1.18) of the Ruge-Stüben-theorem and therefore optimal two-grid convergence for f having a zero of finite order, also for multilevel Toeplitz matrices. Implicitly, level independency of the multigrid method has been proved in [99]. Following the presentation in [112], this implies optimal convergence of the W-cycle, but not of the V-cycle. In [3], several examples have been given to demonstrate that level independency does not imply optimal multigrid convergence. Optimal V-cycle convergence is much harder to prove, because Toeplitz matrices do not have algebra structure. Therefore, such a proof has not yet been given. However, the numerical results presented in [99, 3] suggest that the convergence of the V-cycle method is indeed optimal. Recently, there have been two further proposals for the choice of the restriction matrices. For one dimensional Toeplitz matrices, they enforce the Toeplitz structure on coarser grids by setting some rows in E_j to zero. The first cutting matrix was presented in [99], the second one in [3]:

- $E_j[t]$ is obtained from E_j by deleting the first t and the last t rows, where $t = \deg(b_j) - 1$ with $\deg(b_j)$ denoting the degree of the trigonometric polynomial $b_j(x)$.
- $E_j\{t\}$ is obtained from E_j by deleting only about half as many rows as for $E_j[t]$. If t is even, the first $t/2$ and the last $t/2$ rows of E_j are deleted with t as above.

It is shown that the second proposal preserves the Toeplitz structure, cutting the lowest amount of information. If f has only one isolated zero of order at most two, then $t = 0$ and no extra cutting has to be applied. Thus E_j , $E_j[t]$, and $E_j\{t\}$ are equivalent. For multilevel Toeplitz matrices, the cutting matrices are obtained as tensor products of the one-level cutting matrices.

3.4.3 Using rediscrretization

All multigrid methods for Toeplitz matrices described so far use coarse grid operators which are computed with the Galerkin approach, i.e. as a product of the form $T_C = R^H T_n R$. The main disadvantage of this technique is that in general, Toeplitz structure is lost on coarser levels. Application of additional cutting, as described in Chapter 3.4.2, enforces the Toeplitz structure of T_C , but cuts information by deleting columns of R , and therefore rows and columns of T_C . In [72], Huckle and Staudacher propose a different kind of multigrid method, which uses natural coarse grid operators instead. This means the coarse grid matrix T_C is a Toeplitz matrix of smaller size, generated by the original function f .

If f has a single zero in $]-\pi, \pi]$, the algorithm proposed in [72] starts with a diagonal scaling of T_n which corresponds to a shift of the zero to the origin. Prolongation and restriction are done in a similar way as in [24, 109], but computation of the coarse grid matrix is done differently. A natural coarse grid operator is used, i.e. an operator obtained from rediscrretization of the problem on the coarser grid. The advantage of this technique is that the coarse grid matrix is again Toeplitz, independent of its size. Since the zero remains at the origin on coarser levels, the same prolongation and restriction matrices can be applied on each level. Furthermore, the coarse grid matrices on all levels correspond to the function f , they are just of smaller size. The numerical results presented in [72] show that for large classes of linear systems, the method based on rediscrretization converges as fast as the Galerkin-based methods. Since no Galerkin strategy is used, it is not possible to give convergence proofs based on the theory of Ruge and Stüben.

3.5 Multigrid for block systems

The multigrid methods discussed so far are all designed for one-level or multilevel matrices of the Toeplitz class or of a trigonometric matrix algebra. In this section, we introduce multigrid methods for block Toeplitz matrices, where the blocks are usually small, but unstructured. First, we present an idea of Huckle and Staudacher [71] for the development of a multigrid method for block matrices. Then, we describe, in more detail, how Toeplitz matrices can be treated as block Toeplitz matrices. This technique, which is also applied to trigonometric matrix algebras, will be useful for generating functions with multiple zeros in the next section.

3.5.1 A multigrid method for block Toeplitz systems

Block Toeplitz matrices were introduced in Section 2.1.2. They have Toeplitz structure on the block level and small, unstructured blocks of fixed size k . The corresponding generating functions are k -by- k matrix functions $F(x)$ in one variable (see [93, 96, 79]). The same type of structured matrix exists in two dimen-

sions. We call it *block BTTB matrix* in the following. These matrices have BTTB structure on the block level and unstructured blocks of size l . The corresponding functions are l -by- l matrix functions $F(x, y)$ in two variables.

Huckle and Staudacher [71] developed multigrid methods for one-level block Toeplitz matrices corresponding to *simple* generating matrix functions. They call a matrix function $F(x)$ simple if

- it is symmetric and nonnegative definite,
- all its eigenvalue functions have a single isolated zero in $]-\pi, \pi]$ of order at most 2.

First, the authors use the Galerkin approach to compute the coarse grid matrix. The restriction matrix R is defined as $R = B \cdot E$, where B is chosen to be the block Toeplitz matrix generated by

$$B(x) = \text{diag}(b_1(x), b_2(x), \dots, b_k(x)) \ , \quad (3.5.1)$$

and E^H is the elementary restriction matrix

$$E^H = \begin{pmatrix} I_k & 0_k & 0_k & & & & \\ & 0_k & I_k & 0_k & & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & & & 0_k & I_k \end{pmatrix} \quad (3.5.2)$$

with I_k being the k -by- k identity matrix and 0_k the k -by- k zero matrix. The coarse grid matrix A_C is computed with

$$A_C = E^H \cdot B \cdot A \cdot B^H \cdot E \ ,$$

which, up to a perturbation of low-rank, corresponds to

$$F_2(x) = \frac{1}{2} \cdot \left[B\left(\frac{x}{2}\right)^H F\left(\frac{x}{2}\right) B\left(\frac{x}{2}\right) + B\left(\frac{x}{2} + \pi\right)^H F\left(\frac{x}{2} + \pi\right) B\left(\frac{x}{2} + \pi\right) \right] \ . \quad (3.5.3)$$

The choice of the functions $b_j(x)$ depends on the zeros of the eigenvalue functions $\lambda_j(x)$ of $F(x)$. Under the assumption that either all singularities of $F(x)$ are located at 0 or all singularities are located at π , all $b_j(x)$ are the same, i.e. $B(x) = b(x) \cdot I_k$ with $b(x) = 1 + \cos(x)$ or $b(x) = 1 - \cos(x)$, respectively.

In [71], the authors also propose a different choice of the coarse grid matrix A_C . Instead of computing A_C with the Galerkin method, they employ a natural coarse grid matrix, i.e. a matrix obtained from rediscrretization on the coarser grid. This is the same idea as in [72], which has the advantage that the block Toeplitz structure is retained on coarser levels. However, there are no convergence proof for this method.

3.5.2 Toeplitz matrices treated as block Toeplitz matrices

In [71], there is a brief description how Toeplitz matrices can be interpreted as block Toeplitz matrices and then solved with the multigrid methods described in the previous subsection. Here we wish to extend these results, laying special emphasis on the eigenvalues and eigenvectors of the generating matrix functions and on the treatment of two-dimensional problems.

In the next section, interpreting Toeplitz matrices as block Toeplitz matrices will become a helpful tool for the multigrid solution of Toeplitz systems with several zeros in $]-\pi, \pi]$. Before being presented in a more formal context, this technique shall be illustrated for the simplest case, i.e. for block size $k = 2$.

Motivation

Let $T_n[f]$ be the Toeplitz matrix

$$T_n[f] = \begin{pmatrix} t_0 & t_{-1} & | & t_{-2} & t_{-3} & | & \dots & \dots & | & t_{2-n} & t_{1-n} \\ t_1 & t_0 & | & t_{-1} & t_{-2} & | & \dots & \dots & | & t_{3-n} & t_{2-n} \\ \hline & & & & & & \dots & \dots & & & \\ t_2 & t_1 & | & \ddots & & \vdots & & & \vdots & \vdots & \vdots \\ t_3 & t_2 & | & & \ddots & \vdots & & & \vdots & \vdots & \vdots \\ \hline \vdots & \vdots & \vdots & & & \vdots & \ddots & & | & t_{-2} & t_{-3} \\ \vdots & \vdots & \vdots & & & \vdots & & \ddots & | & t_{-1} & t_{-2} \\ \hline & & & \dots & \dots & & & & & & \\ t_{n-2} & t_{n-3} & | & \dots & \dots & | & t_2 & t_1 & | & t_0 & t_{-1} \\ t_{n-1} & t_{n-2} & | & \dots & \dots & | & t_3 & t_2 & | & t_1 & t_0 \end{pmatrix} \quad (3.5.4)$$

corresponding to the generating function $f(x)$. Interpretation of $T_n[f]$ as a block Toeplitz matrix (see [71]) leads to the generating matrix function

$$\begin{aligned} F(x) &= \dots + e^{-ix} \begin{pmatrix} t_2 & t_1 \\ t_3 & t_2 \end{pmatrix} + \begin{pmatrix} t_0 & t_{-1} \\ t_1 & t_0 \end{pmatrix} + e^{ix} \begin{pmatrix} t_{-2} & t_{-3} \\ t_{-1} & t_{-2} \end{pmatrix} + \dots \\ &= \frac{1}{2} \begin{pmatrix} f(x/2) + f(x/2 + \pi) & e^{ix/2}(f(x/2) - f(x/2 + \pi)) \\ e^{-ix/2}(f(x/2) - f(x/2 + \pi)) & f(x/2) + f(x/2 + \pi) \end{pmatrix}. \end{aligned} \quad (3.5.5)$$

The eigenvalues of the 2-by-2 matrix function $F(x)$ are easily computed:

$$\lambda_0(x) = f\left(\frac{x}{2}\right), \quad \lambda_1(x) = f\left(\frac{x}{2} + \pi\right). \quad (3.5.6)$$

In the following, we present a one-dimensional and a two-dimensional example function, which will be used to illustrate the main ideas.

Example 2 For a positive integer k ,

$$f(x) = 1 - \cos(kx) \quad (3.5.7)$$

has k zeros in $[-\pi, \pi[$. If k is even, f is zero at both 0 and π . A two-dimensional example is the function

$$g(x) = 2 - \cos(kx) - \cos(ky), \quad (3.5.8)$$

which has k^2 zeros in $[-\pi, \pi]^2$. The corresponding structured matrices are denoted $T_n[f]$ and $T_n[g]$.

For the function $f(x)$ from Example 2 with $k = 2$ this means that

$$\lambda_0(x) = \lambda_1(x) = 1 - \cos(x),$$

i.e. both eigenvalue functions only have a zero at the origin. Since $f(x/2) - f(x/2 + \pi) \equiv 0$ in this case, $F(x)$ is a diagonal matrix. This example suggests another interpretation of the block Toeplitz approach. The two functions in the diagonal of $F(x)$ can be interpreted as generating functions of two Toeplitz matrices of size $n/2$ which are the diagonal blocks of the matrix $\tilde{T}_n = \begin{pmatrix} T_{n/2}[\lambda_0(x)] & 0 \\ 0 & T_{n/2}[\lambda_1(x)] \end{pmatrix}$. $\tilde{T}_n = T_n[f](perm, perm)$ is obtained by permuting rows and columns of $T_n[f]$ with the vector

$$perm = [1, 3, 5, \dots, n-1, 2, 4, 6, \dots, n]^T.$$

Functions of the form $f(x) = 1 - \cos(kx)$ are solved in a similar way by using blocks of size k . Then, all eigenvalue functions are of the form $\lambda_j(x) = 1 - \cos(x)$.

In two dimensions, a BTTB matrix can be interpreted as a block BTTB matrix. The procedure illustrated in (3.5.4) is carried out in two dimensions. This leads to small blocks of size k^2 with $k \in \{2, 3, 4, \dots\}$. For $k = 2$, we obtain the analogue of the first line in (3.5.5)

$$G(x, y) = \sum_{l=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} e^{i(lx+my)} \cdot T_{lm}, \quad (3.5.9)$$

where the matrices $T_{l,m}$ are of the form

$$T_{l,m} = \begin{pmatrix} t_{2m,2l} & t_{2m,2l-1} & t_{2m-1,2l} & t_{2m-1,2l-1} \\ t_{2m,2l+1} & t_{2m,2l} & t_{2m-1,2l+1} & t_{2m-1,2l} \\ t_{2m+1,2l} & t_{2m+1,2l-1} & t_{2m,2l} & t_{2m,2l-1} \\ t_{2m+1,2l+1} & t_{2m+1,2l} & t_{2m,2l+1} & t_{2m,2l} \end{pmatrix}. \quad (3.5.10)$$

The generating matrix function $G(x, y)$ is then a k^2 -by- k^2 function, which is the analogue of the second line in (3.5.5). For $k = 2$, it is an Hermitian BTTB matrix of the form

$$G(x, y) = \begin{pmatrix} G_0(x, y) & G_1^H(x, y) \\ G_1(x, y) & G_0(x, y) \end{pmatrix}. \quad (3.5.11)$$

where

$$G_0(x, y) = \begin{pmatrix} g_{00} + g_{\pi 0} + g_{0\pi} + g_{\pi\pi} & e^{i\frac{x}{2}}(g_{00} - g_{\pi 0} + g_{0\pi} - g_{\pi\pi}) \\ e^{-i\frac{x}{2}}(g_{00} - g_{\pi 0} + g_{0\pi} - g_{\pi\pi}) & g_{00} + g_{\pi 0} + g_{0\pi} + g_{\pi\pi} \end{pmatrix} \quad (3.5.12)$$

and

$$G_1(x, y) = \begin{pmatrix} e^{-i\frac{y}{2}}(g_{00} + g_{\pi 0} - g_{0\pi} - g_{\pi\pi}) & e^{i(\frac{x}{2} - \frac{y}{2})}(g_{00} - g_{\pi 0} - g_{0\pi} + g_{\pi\pi}) \\ e^{-i(\frac{x}{2} + \frac{y}{2})}(g_{00} - g_{\pi 0} - g_{0\pi} + g_{\pi\pi}) & e^{-i\frac{y}{2}}(g_{00} + g_{\pi 0} - g_{0\pi} - g_{\pi\pi}) \end{pmatrix} \quad (3.5.13)$$

with the abbreviations

$$g_{00} = g\left(\frac{x}{2}, \frac{y}{2}\right), \quad g_{\pi 0} = g\left(\frac{x}{2} + \pi, \frac{y}{2}\right), \quad g_{0\pi} = g\left(\frac{x}{2}, \frac{y}{2} + \pi\right), \quad g_{\pi\pi} = g\left(\frac{x}{2} + \pi, \frac{y}{2} + \pi\right).$$

The eigenvalue functions of $G(x, y)$ are

$$\begin{aligned} \lambda_0(x, y) &= f\left(\frac{x}{2}, \frac{y}{2}\right), \quad \lambda_1(x, y) = f\left(\frac{x}{2} + \pi, \frac{y}{2}\right), \\ \lambda_2(x, y) &= f\left(\frac{x}{2}, \frac{y}{2} + \pi\right), \quad \lambda_3(x, y) = f\left(\frac{x}{2} + \pi, \frac{y}{2} + \pi\right). \end{aligned} \quad (3.5.14)$$

For the example function $g(x, y)$ from Example 2, $G(x, y)$ becomes a 4-by-4 diagonal matrix with eigenvalues $\lambda_j = 2 - \cos(x) - \cos(y)$. Again, this can be interpreted as a permutation of $\tilde{T}_{\mathbf{n}} = T_{\mathbf{n}}[f](perm2, perm2)$, where $perm2$ is the two-dimensional analogue of $perm$. If g from Example 2 with $k = 2$ is interpreted as a block BTTB matrix, the generating matrix function is the diagonal matrix

$$G(x, y) = (2 - \cos(x) - \cos(y)) \cdot I_4 \quad .$$

Eigenvalues of the generating matrix functions

We have outlined how one- and two-dimensional Toeplitz systems can be treated as block Toeplitz and block BTTB systems, respectively. This allows multigrid methods to be applied also to problems with multiple zeros and especially with zeros at the mirror points of another zero. As a motivation for the following analysis we have illustrated that this approach leads to considerable improvement in the special case of block size 2. In order to apply this approach to more general problems, we wish to examine the generating matrix functions as well as their eigenvalues and eigenvectors for arbitrary block sizes. First, we introduce the following bit of notation.

Definition 1

Let $T_n[f]$ be a Toeplitz matrix corresponding to the generating function $f(x)$, and let $T_n[f]$ be treated as a block Toeplitz matrix with blocks of size k . Then we define, for $0 \leq l \leq k - 1$:

$$f_l := f\left(\frac{x}{k} + \frac{l}{k} \cdot 2\pi\right) \quad . \quad (3.5.15)$$

Furthermore, let $T_n[f]$ be the BTTB matrix corresponding to $f(x, y)$, and let $T_n[f]$ be treated as a block BTTB matrix with blocks of size k^2 . Then we define, for $0 \leq l, m \leq k - 1$:

$$f_{l,m} := f\left(\frac{x}{k} + \frac{m}{k} \cdot 2\pi, \frac{y}{k} + \frac{l}{k} \cdot 2\pi\right). \quad (3.5.16)$$

With this notation, we can analyze the generating matrix functions more formally. For the development of multigrid methods, we are mainly interested in their eigenvalues and eigenvectors. In one dimension, the main result is stated in the following theorem.

Theorem 8

Let $T_n[f]$ be a Toeplitz matrix corresponding to the generating function $f(x)$, and let $T_n[f]$ be treated as a block Toeplitz matrix with blocks of size k . Then, the generating function $F(x)$ is a k -by- k matrix with the following properties:

1. $F(x)$ is an Hermitian Toeplitz matrix with first column

$$F_0(x) = \frac{1}{k} \begin{pmatrix} \sum_{j=0}^{k-1} f_j \\ e^{-ix/k} \sum_{j=0}^{k-1} e^{-2ij\pi/k} f_j \\ \vdots \\ e^{-i(k-1)x/k} \sum_{j=0}^{k-1} e^{-2ij(k-1)\pi/k} f_j \end{pmatrix}. \quad (3.5.17)$$

2. The eigenvalues of $F(x)$ are

$$\lambda_0 = f_0, \dots, \lambda_1 = f_1, \dots, \lambda_{k-1} = f_{k-1}. \quad (3.5.18)$$

3. The eigenvector v_j corresponding to λ_j is

$$v_j = \begin{pmatrix} e^{2ij \frac{k-1}{k} \pi} e^{i \frac{k-1}{k} x} \\ \vdots \\ e^{2ij \frac{2}{k} \pi} e^{i \frac{2}{k} x} \\ e^{2ij \frac{1}{k} \pi} e^{i \frac{1}{k} x} \\ 1 \end{pmatrix}. \quad (3.5.19)$$

Proof: The result in 1. is a generalization of (3.5.4) and (3.5.5). It is obtained from the Fourier representation of $f(x)$.

2. and 3. are proved by direct calculation. For each eigenvalue λ_j , we can verify that

$$F(x)v_j = \lambda_j v_j,$$

which completes the proof. \blacksquare

In two dimensions, similar results are obtained, computations are performed with tensor arguments. The analogue of Theorem 8 can be proved in the same way.

Theorem 9

Let $T_n[f]$ be a BTTB matrix corresponding to the generating function $f(x, y)$, and let $T_n[f]$ be treated as a block BTTB matrix with blocks of size k^2 . Then, the generating function $F(x, y)$ is a k^2 -by- k^2 matrix with the following properties:

1. $F(x, y)$ is an Hermitian BTTB matrix with Hermitian blocks. For $0 \leq l, m < k$, the element $F_{l,m}$ (i.e. the m -th element in the l -th block), is given by

$$F_{l,m} = e^{-imx/k} e^{-ily/k} \sum_{r=0}^{k-1} \sum_{s=0}^{k-1} e^{-2irl\pi/k} e^{-2ism\pi/k} f_{r,s}. \quad (3.5.20)$$

2. The eigenvalues of $F(x, y)$ are

$$\begin{aligned} \lambda_{0,0} &= f_{0,0}, \dots, \lambda_{0,1} = f_{0,1}, \dots, \lambda_{0,k-1} = f_{0,k-1} \\ \lambda_{1,0} &= f_{1,0}, \dots, \lambda_{1,1} = f_{1,1}, \dots, \lambda_{1,k-1} = f_{1,k-1} \\ &\vdots \\ \lambda_{k-1,0} &= f_{k-1,0}, \dots, \lambda_{k-1,1} = f_{k-1,1}, \dots, \lambda_{k-1,k-1} = f_{k-1,k-1}. \end{aligned} \quad (3.5.21)$$

3. The eigenvector $v_{l,m}$ corresponding to $\lambda_{l,m}$ is

$$v_{l,m} = v_l \otimes v_m \quad (3.5.22)$$

with v_l and v_m from (3.5.19).

Consequences of the results on eigenvalues

The eigenvalues of the matrix-valued functions $F(x)$ or $F(x, y)$ are crucial for the development of multigrid methods. Since Theorems 8 and 9 yield very simple formulas for these eigenvalues, we can now state some immediate consequences. The first corollary ensures that all zeros of $f(x)$ which are located at $x = \frac{l}{k} \cdot 2\pi$ ($l \in \{0, \dots, k-1\}$) lead to zeros of the eigenvalue functions located at integer multiples of 2π .

Corollary 3

Let $T_n[f]$ be a Toeplitz matrix corresponding to the nonnegative generating function $f(x)$, and let $T_n[f]$ be treated as a block Toeplitz matrix with blocks of size k .

1. If $f(x)$ has periodicity p , then all functions $f_j(x)$ have periodicity $k \cdot p$.

2. Suppose $f(x) = 0$ implies that $\exists l \in \{0, \dots, k-1\}$ such that $x = \frac{l}{k} \cdot 2\pi$. Then, for each eigenvalue function $\lambda_j(x)$, the following holds:

- $\lambda_j(x)$ is nonnegative
- $\lambda_j(x) = 0$ implies $\exists m \in \mathbb{Z}$ such that $x = m \cdot 2\pi$.

3. In particular, suppose $f(x)$ has a zero at $x = \frac{l}{k} \cdot 2\pi$ ($l \in \{0, \dots, k-1\}$). Then the j -th eigenvalue function $\lambda_j(x)$ of $F(x)$ has a zero at $x = (l - j) \cdot 2\pi$.

Proof:

- The first result follows from the definition of the f_j in (3.5.15). Whereas x/k is responsible for the periodicity of f_j , $2j\pi/k$ represents the shift of the zeros.
- Since $f(x)$ is nonnegative,

$$\lambda_j(x) = f_j(x) = f\left(\frac{x}{k} + \frac{j}{k} \cdot 2\pi\right) \tag{3.5.23}$$

guarantees that all $\lambda_j(x)$ are also nonnegative.

- Let us now suppose $f_j(x) = 0$. With (3.5.23) this is equivalent to

$$f\left(\frac{x}{k} + \frac{j}{k} \cdot 2\pi\right) = 0.$$

Since by assumption the zeros of $f(x)$ are located at $x = \frac{l}{k} \cdot 2\pi$, the following equation must hold:

$$\frac{x}{k} + \frac{j}{k} \cdot 2\pi = \frac{l}{k} \cdot 2\pi.$$

This is equivalent to

$$x = (l - j) \cdot 2\pi,$$

proving the rest of 2. and 3. ■

This Corollary ensures that zeros of the eigenvalue functions only occur at integer multiples of 2π . However, it does not state whether the λ_j are zero for all of these points, which would be very helpful for the construction of a prolongation matrix. The following corollary gives one positive and one negative answer to this question.

Corollary 4

Let $T_n[f]$ be a Toeplitz matrix corresponding to the 2π -periodic nonnegative generating function $f(x)$, and let $T_n[f]$ be treated as a block Toeplitz matrix with blocks of size k .

1. If, for all $j \in \{0, \dots, k-1\}$, $f(\frac{j}{k} \cdot 2\pi) = 0$, then all functions $\lambda_j(x)$ become zero for all $x = l \cdot 2\pi$ ($l \in \mathbb{Z}$). If, in addition, $f(x) > 0$ for all other x , then the λ_j are strictly positive for $x \neq l \cdot 2\pi$ ($l \in \mathbb{Z}$).

2. If $\exists j \in \{0, \dots, k-1\}$ such that $f(\frac{j}{k}) > 0$, then for each λ_j the following holds: $\exists m \in \mathbb{Z}$ such that $f_j(m \cdot 2\pi) > 0$.

Proof: The results are immediate consequences of Definition 1 and of the third part of Corollary 3, which is applied to each zero of $f(x)$. ■

Remark 7 The results of Corollaries 3 and 4 hold unchanged in the two-level case.

- The periodicity of the $\lambda_{r,s}$ is k times the periodicity of f both in x - and in y -direction.
- A zero of $f(x, y)$ at $(\frac{m}{k} \cdot 2\pi, \frac{l}{k} \cdot 2\pi)$ leads to a zero of the eigenvalue function $\lambda_{r,s}$ at $((m-s) \cdot 2\pi, (l-r) \cdot 2\pi)$.
- If, for all $r, s \in \{0, \dots, k-1\}$, $f(\frac{m}{k} \cdot 2\pi, \frac{l}{k} \cdot 2\pi) = 0$, then all eigenvalue functions are zero at $(m \cdot 2\pi, l \cdot 2\pi)$ for all $l, m \in \mathbb{Z}$.

3.6 Generating functions with multiple zeros

So far, we have considered linear systems where the corresponding generating function $f(x)$ has a single isolated zero x_0 of finite order in the interval $[-\pi, \pi[$. If $f(x)$ has several zeros in $[-\pi, \pi[$, it is more complicated to define prolongation matrices. Especially in the case where $f(x)$ has another zero at $\pi - x_0$, the convergence theory described in the previous sections does not hold anymore, and the multigrid methods fail in most numerical experiments. If a two-dimensional function $f(x, y)$ with a zero (x_0, y_0) vanishes at one of the mirror points from (3.3.20) or (3.3.21), standard multigrid fails for the same reason. However, generating functions with multiple zeros arise in many applications. In this work, they become important for the solution of certain two-level Toeplitz or trigonometric algebra systems which are anisotropic.

In this section, we develop multigrid methods especially designed for linear systems whose generating functions have multiple zeros. Starting from suggestions in [99, 3], we discuss generalizations of the multigrid methods present so far, where each zero is treated individually. Then, we make use of the block interpretation of Toeplitz matrices from Section 3.5.2, laying special emphasis on the eigenvalues of the generating matrix functions. The multigrid methods obtained from this approach are eventually combined with the former ones.

3.6.1 Treating each zero separately

In [99, 3], the authors mention that their methods can be easily extended to linear systems corresponding to functions with several zeros. Let us assume that $f(x)$ has l zeros located at x_j ($1 \leq j \leq l$). If none of the zeros are located at distance

π , only the prolongation function b has to be modified. This is the first strategy we wish to discuss. It has been suggested in [99, 3].

- (1) The multigrid methods from Sections 3.3.1 and 3.4.2 are used with b from (3.3.16) being replaced by the product

$$b(x) = (1 + \cos(x - x_1))^{q_1} \cdots (1 + \cos(x - x_l))^{q_l} \quad (3.6.1)$$

in the Toeplitz or circulant case and by

$$b(x) = (\cos(x_1) + \cos(x))^{2q_1} \cdots (\cos(x_l) + \cos(x))^{2q_l} \quad (3.6.2)$$

in the tau case. For a two-dimensional function $f(x, y)$, the prolongation function $b(x, y)$ is defined analogously.

This technique works well if f has a small number of zeros, and if these are not located at distance π . For a larger number of zeros, this approach has the disadvantage that matrices become denser on coarser levels, and therefore the multigrid method loses its efficiency. Both multigrid solvers and multigrid preconditioners are constructed with this method.

In the following, we present a different technique, which also treats each zero of f individually. It can be used for both one- and multilevel structured matrices. First, we describe the construction of a multigrid preconditioner which is based on the diagonal scaling method described in Section 3.1.3. Then we present a modified version of the V-cycle which allows more than one coarse grid correction per iteration.

- (2) For each zero x_k , separate restriction and coarse grid matrices are computed. We construct a diagonal scaling preconditioner P for the linear system $Ax = b$ as the sum of l coarse grid corrections. The two-grid version is of the form

$$P = I + R_1 A_{C_1}^{-1} R_1^H + \cdots + R_l A_{C_l}^{-1} R_l^H \quad (3.6.3)$$

with $A_{C_j} = R_j^H A R_j$. The restriction matrices are constructed from the l functions

$$b^{(j)}(x) = (1 + \cos(x - x_j))^{q_j} \quad (3.6.4)$$

or

$$b^{(j)}(x) = (\cos x_j + \cos(x))^{2q_j} \quad (3.6.5)$$

This two-grid preconditioner is extended to a multigrid preconditioner by applying the two-grid idea recursively to the A_{C_j} , i.e. by replacing $A_{C_j}^{-1}$ with $I + \tilde{R}_j A_{CC_j}^{-1} \tilde{R}_j^H$, where A_{CC_j} is the representation of the matrix A_{C_j} on the next coarser grid. If multigrid is used as a solver, we modify the standard V-cycle such that several coarse grid corrections are used in every multigrid iteration, one for each zero. The coarse grid matrices are computed in the

same way as was done for the preconditioner, i.e. with b_j from (3.6.4) or (3.6.5). Then, one iteration of the multigrid method consists of k coarse grid corrections with smoothing between each of them. We will follow this idea in much more detail in Chapter 5, where the structure of one iteration is shown in Figure 5.5.

Since for each zero, a coarse grid correction is computed in each iteration, this technique is also limited to a rather small number of zeros. Unlike **(1)**, **(2)** can be used for arbitrary zeros, even if they are located at distance π .

3.6.2 Using the block approach

The two multigrid strategies presented in the previous subsection are very useful for generating functions with a small number of zeros, distributed irregularly in $[-\pi, \pi]^d$. The opposite case is a large number of zeros, distributed regularly, or even equidistantly in $[-\pi, \pi]^d$. Typical functions are the ones from Example 2. For these types of linear systems, we propose different multigrid techniques which are based on the block interpretation of structured matrices introduced in Section 3.5. The multigrid methods for block Toeplitz matrices described in [24, 71] efficiently solve systems like the ones from Example 2. We have described the approach from [71] in Section 3.5 and given an analysis of this method in terms of generating matrix functions and its eigenvalues. In the following, we wish to apply this method to larger classes of matrices whose generating functions have multiple zeros which are distributed equidistantly or at least somehow regularly. To simplify notation, we start with the one-level Toeplitz case. The trigonometric algebra classes, which can be treated similarly, will be described later. Having the results from Corollaries 3 and 4 in mind, we propose a third strategy for the multigrid solution of linear systems whose generating function has several zeros.

- (3)** Instead of computing several coarse grid corrections, one for each zero of f , use a single coarse grid correction in each iteration and construct a multigrid method following these ideas:
- Treat the Toeplitz matrix $T_n[f]$ as a block Toeplitz matrix with small blocks of size k , which corresponds to a k -by- k generating matrix function $F(x)$.
 - If all zeros of f are located at points $x = l \cdot 2\pi/k$, choose k as the smallest natural number such that the eigenvalue functions of $F(x)$ do not have multiple zeros in $]-\pi, \pi]$ anymore.
 - Solve the system with the multigrid method for block Toeplitz matrices described in 3.5.1.

The more equidistant the distribution of the zeros of f is, the better are the results obtained with this approach. In other words, if f is zero at all $x = l \cdot 2\pi/k$, then all

functions f_j have an isolated zero in $]-\pi, \pi]$, which is located at the origin. In this case, the method from [71] converges rapidly. However, if f is nonzero at many of the points $x = l \cdot 2\pi/k$, then f_j is nonzero at $x = l \cdot 2\pi$ due to Corollary 4. The f_j are not 2π -periodic and the multigrid method converges significantly slower. The following examples illustrate under which circumstances the block approach yields fast convergence.

- Excellent convergence is obtained for the functions from Example 2. Since $f(x)$ is zero for all $x = 2\pi l/k$ ($l \in \{0, \dots, k-1\}$), all functions $f_j(x)$ are zero at all multiples of 2π .
- Rather poor convergence is obtained for functions like

$$f(x) = (1 - \cos(3x)) \cdot (1 - \cos(5x)) . \quad (3.6.6)$$

To avoid multiple zeros in $]-\pi, \pi]$, we have to choose $k = 15$, and this implies that f is zero at only some of the points $x = k/15 \cdot 2\pi$. The multigrid method still converges, but this convergence is extremely slow.

In two-dimensions, i.e. for BTTB matrices corresponding to generating functions with multiple zeros, the same behavior of the multigrid methods is observed.

3.6.3 Multigrid methods with blocks and splitting

The block approach presented in the previous subsection ran into problems for functions f which are zero at $x = l \cdot 2\pi/k$ for only some $l \in \mathbb{Z}$. A typical example is f from (3.6.6). For examples like this, however, the block approach can still be used for the design of an efficient method. By combining the block interpretation with strategy **(2)**, we obtain another strategy for the development of multigrid methods.

- (4)** Divide the zeros of f into two or more subsets and apply the block approach from **(3)** to compute a coarse grid correction for each of them. This leads to significantly smaller blocks than in **(3)**. If multigrid is used as a preconditioner, it is of the same form as in (3.6.3), but the number of coarse grid corrections l is usually significantly smaller.

A typical example where this approach works well is f from (3.6.6). Two coarse grid corrections are necessary. One of them is computed with blocks of size $k_1 = 3$, the other one with blocks of size $k_2 = 5$. The zero at the origin occurs in both subsets, and one has to keep in mind that it has the order 4.

Strategy **(4)** also works if the block approach is only used for some of the subsets. For example, the function

$$f(x) = (1 - \cos(5x))(1 + \cos(x))$$

has zeros at $x = l \cdot 2\pi/5$ for all $l \in \mathbb{Z}$ and at $x = \pi$. Hence, the first subset contains the zeros $x = l \cdot 2\pi/5$, leading to a blocksize $k_1 = 5$ for the first coarse grid correction, whereas the second subset only contains $x = \pi$. This implies that $k_2 = 1$, i.e. no blocks are used for the computation of the second coarse grid correction.

Chapter 4

Multigrid methods for anisotropic systems

Many applications such as the discretization of partial differential equations lead to anisotropic linear systems of two-level Toeplitz or trigonometric algebra type. Although for some of these systems, the classical convergence theory from Chapter 3 still holds, standard multigrid methods converge so slowly that they become totally impractical. Therefore, we devise multigrid methods which are especially designed for application to anisotropic problems. We wish to emphasize that in this Chapter, multigrid is only used as a standalone solver. The use of multigrid as a preconditioner will be discussed in Chapter 5, where the differences between multigrid solver and multigrid preconditioner are more evident.

In Section 4.1, we start with defining anisotropy in the context of generating functions. Then, we describe what kind of difficulties the multigrid methods from Chapter 3 encounter when they are applied to anisotropic linear systems. In the rest of this chapter, multigrid algorithms for two different types of anisotropic problems are developed. In Sections 4.2 and 4.3, we consider systems where anisotropy occurs along coordinate axes. In Section 4.2, we develop multigrid methods based on a suitable combination of semicoarsening and full coarsening steps, whereas in Section 4.3, the focus is on the use of more sophisticated smoothers. Some of these results are known from the solution of partial differential equations, but here we present them in a slightly different context, making explicit use of the strong connection between structured matrices and generating functions with their level curves. This has the advantage that the methods can be extended to more general classes of matrices, and that convergence can be proved in a formal way. The problems considered in Sections 4.4 and 4.5 are more difficult to solve, because anisotropy occurs in other directions. We develop multigrid methods which are suitable for this case by carrying over the results from Sections 4.2 and 4.3. We focus on Toeplitz and circulant matrices and on directions where anisotropy occurs in an angle of $\frac{k}{k+l} \cdot 90^\circ$ towards one of the axes (with k and l being small integers).

Although the classical two-level matrix structure with blocks of equal size is lost for Toeplitz matrices, the methods still work in this case. For circulant matrices, the exact correspondence between generating functions and matrices with blocks of equal size can be carried over. We are mostly interested in sparse examples, which arise e.g. from the discretization of partial differential equations. However, our methods also work in the more general case of dense matrices corresponding to arbitrary generating functions. The development of fast multigrid methods for dense matrices is especially interesting in the Toeplitz case, because there is no fast direct solver available.

4.1 Anisotropy in terms of generating functions

The phenomenon of anisotropy is widely known from the numerical solution of partial differential equations. Here we introduce it in the context of structured linear systems and generating functions. After explaining what anisotropy is, we describe the convergence-related difficulties of the methods from Chapter 3.

4.1.1 Definition of Anisotropy

Anisotropy in general is a function's property of being directionally dependent, i.e. of having a different value when measured in different directions. In the context of PDEs, this behavior occurs rather frequently. Two of the reasons are the following (see e.g. [19]):

- The coefficients of the derivatives are quite different. A typical example is the anisotropic version of the Poisson equation:

$$-\epsilon \cdot u_{xx} - u_{yy} = g \quad (0 \leq \epsilon \ll 1) . \quad (4.1.1)$$

- The discretization uses a different mesh size in each direction.

Since in this work, we solve linear systems corresponding to generating functions, we wish to treat anisotropy also in terms of generating functions. Let us start with two examples. The first of them is obtained from finite-difference discretization of the anisotropic Poisson equation with a five point stencil on a uniform mesh. Depending on the type of boundary condition, one obtains a two-level Toeplitz matrix or a matrix from a trigonometric algebra.

Example 3 Let $A_n[f]$ be the matrix belonging to the Toeplitz class or to a trigonometric matrix algebra corresponding to the generating function

$$f(x, y) = \alpha \cdot (1 - \cos(x)) + (1 - \cos(y)) . \quad (4.1.2)$$

If $\alpha = 1$, we get one of the isotropic standard model problems, the discrete Poisson equation. For $\alpha \ll 1$, the problem becomes strongly anisotropic.

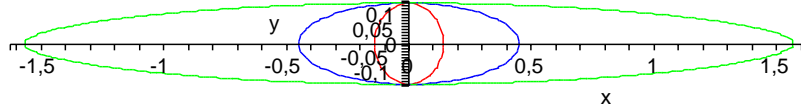


Figure 4.1: Curves $f(x, y) = 0.01$ for the function from Example 3 with $\alpha = 1, 0.1, 0.01$

In terms of generating functions anisotropy means that the level curves of $f(x, y)$ become extremely flat. This is illustrated in Figure 4.1, which depicts the curve $f(x, y) = 0.01$ for three different values of α , i.e. for three different degrees of anisotropy. The second example presents a matrix which has a similar type of anisotropy, but which is not sparse.

Example 4 Let $A_n[f]$ be the matrix belonging to the Toeplitz class or to a trigonometric matrix algebra with the underlying generating function

$$f(x, y) = \alpha x^2 + y^2 \quad (\alpha \ll 1) . \quad (4.1.3)$$

It also has a single zero of order two at the origin, and it can be written as the Fourier sum

$$f(x, y) = (1 + \alpha) \frac{\pi^3}{3} + 4\pi \sum_{j=1}^{\infty} \frac{(-1)^j}{j^2} (\alpha \cos(jx) + \cos(jy)) . \quad (4.1.4)$$

These two functions are examples where anisotropy occurs along one of the coordinate axes. In most of this article, we restrict ourselves to anisotropic linear systems which correspond to generating functions with a single zero in $[-\pi, \pi]^2$ of order two, because these can be described best in a formal way. However, we will also outline how to solve certain anisotropic systems whose function has several zeros or zeros of higher order. Let us now assume that f has a single zero at the origin of order 2. Then the Taylor expansion of f is of the form

$$f(x, y) = ax^2 + bxy + cy^2 + \dots = \begin{pmatrix} x & y \end{pmatrix} M \begin{pmatrix} x \\ y \end{pmatrix} + \dots \quad (4.1.5)$$

with $M = \begin{pmatrix} a & b/2 \\ b/2 & c \end{pmatrix}$. For the analysis of $f(x, y)$ in the neighborhood of the origin, we omit all higher order terms and describe f only with the symmetric matrix M . Since f is nonnegative, M is positive semidefinite, i.e. its eigenvalues are nonnegative. The eigenvalues λ_1, λ_2 of M give information about the degree of anisotropy, the corresponding orthogonal eigenvectors v_1, v_2 about the direction in which anisotropy occurs. If exactly one of the eigenvalues is close to zero, anisotropy is strong. In the limit case, λ_1 or λ_2 is zero. This means that f is zero

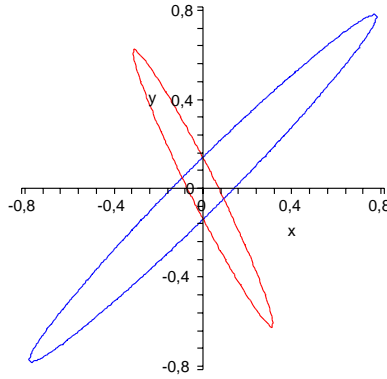


Figure 4.2: Level curves $f(x, y) = 0.01$ and $g(x, y) = 0.01$ for the functions from Example 5 with $\alpha = 0.01$

along a whole line which passes through the origin. In Example 3, the function f can be described by the diagonal matrix $M = \frac{1}{2} \cdot \begin{pmatrix} \alpha & 0 \\ 0 & 1 \end{pmatrix}$.

In this work, we are also interested in more general classes of functions, where anisotropy occurs in other directions. The following two functions correspond to model problems of this more general type.

Example 5 Let $A_n[f]$ and $A_n[g]$ be two-level Toeplitz or trigonometric algebra matrices corresponding to the functions

$$\begin{aligned} f(x, y) &= \alpha \cdot (1 - \cos(x + y)) + (1 - \cos(x - y)) , \\ g(x, y) &= (1 - \cos(2x + y)) + \alpha \cdot (1 - \cos(x - 2y)) . \end{aligned} \tag{4.1.6}$$

Figure 4.2 shows how the two functions behave in the neighborhood of their zero at the origin, i.e. with what kind of anisotropy we have to deal with. f is anisotropic along the line $y = x$, which means it is rotated by an angle of 45 degrees from the x -axis. The anisotropy of g occurs along $y = -2x$, which corresponds to an angle of 30 degrees from the y -axis.

4.1.2 Problems arising from anisotropic systems

The multigrid methods from Chapter 3 do not work with the same efficiency when they are applied to anisotropic problems. If the anisotropy is strong, they fail completely. In order to explain the reasons for this behavior, we begin with the following observation. As described in Chapter 3, the V-cycle multigrid method converges if f_j and b_j satisfy conditions such as (3.3.22) and (3.3.23). A necessary condition for this is that, on each level, for each zero of f_j , the function b_j is

zero at the mirror points, and f_j itself is nonzero at these points [99, 3, 103]. However, apart from these convergence results, there is also one important negative observation which will be crucial for the development of our multigrid methods.

Remark 8 Let $f(x, y)$ be a nonnegative generating function which has a zero at $(0, 0)$. If f has another zero at one of the mirror points, the multigrid method from Chapter 3 fails completely in all numerical experiments. A theoretical reason for this behavior can be found in [100, 3]: The convergence theory for the multigrid method of Serra [54, 100] requires that b is nonzero at the origin and zero at all mirror points. Hence, if f is zero at the origin and at one of the mirror points, b cannot meet both requirements. Even if f is close to zero at one of the three points $(0, \pi)$, $(\pi, 0)$, (π, π) , convergence of the multigrid method is extremely slow.

In the following, we describe what problems arise when the standard multigrid methods from Chapter 3 are applied to anisotropic linear systems. Without loss of generality, we assume that the generating function is zero at the origin and anisotropic in its neighborhood. The function f from Example 3 serves to illustrate how generating functions are used for an analysis of multigrid methods. We distinguish two different cases, linear systems with strong anisotropy and linear systems with only moderate anisotropy.

1. If anisotropy is strong, i.e. if $0 < \alpha \ll 1$ or even $\alpha \rightarrow 0$, then the methods fail completely for the following reasons:
 - If anisotropy occurs along one of the axes or in an angle of 45° to the axes, the function f becomes close to zero at one of the mirror points. In Example 3, this happens at $(\pi, 0)$. From Remark 8 we know that convergence is extremely slow in this case.
 - For anisotropy in arbitrary directions, f becomes close to zero along a whole line. In the case of f from Example 3, this line is the x -axis. This means that multigrid methods which are designed for functions with a single isolated zero are no more applicable.
 - Most interesting sparse examples with anisotropy in other directions, i.e. not along the coordinate axes, have several zeros in $[-\pi, \pi]^2$, which means several lines of zero in the limit case.
2. If the linear system is only moderately anisotropic, the methods from Chapter 3 can still be applied. However, they become more and more inefficient with an increasing degree of anisotropy. The reason for this is the weak connection in one direction due to small coefficients in front of some terms in f , which, in the case of f from (4.1.2) are the x -terms. This can be illustrated very well with the level curves of the generating function f from Example 3. Figure 4.1 depicts the curve $f(x, y) = 0.01$ for three different values of α , i.e. for three different degrees of anisotropy. The solid curve for

$\alpha = 1$ is totally isotropic, i.e. almost like a circle. The dashed curve with $\alpha = 0.1$ is only moderately anisotropic, and the dotted curve for $\alpha = 0.01$ is significantly more anisotropic, which means that the curve is very flat. If anisotropy is even stronger, the value of the function hardly depends on x , making coarsening in x -direction rather useless. In the limit case, i.e. for $\alpha = 0$, we obtain n_2 independent 1D Poisson systems.

4.1.3 How to design multigrid methods for anisotropic systems?

Having defined anisotropic systems and having described the problems arising from the use of multigrid methods, we wish to develop multigrid solvers which overcome these problems and which converge optimally. There are two fundamentally different concepts for the design of multigrid methods for anisotropic linear systems. Both are known from the solution of partial differential equations, but in this work, we present them in the context of generating functions and the corresponding matrices. This leads to a more formal treatment of the two concepts. The first of them is based on semicoarsening, the second on line smoothing.

- **Semicoarsening:** Since strong coupling occurs only in one direction, relaxation leads to a sufficiently smooth error only in this direction. Therefore, we perform coarsening only in this direction. This strategy known as semicoarsening reduces the degree of anisotropy in each step. Usually, semicoarsening is combined with a standard smoother such as the Jacobi or Gauss-Seidel method.
- **Line smoothing:** The standard coarsening strategy from Chapter 3 can still be used if we choose a more sophisticated smoother, which must be especially well-suited for anisotropic systems. Instead of performing pointwise relaxation, the smoother solves for entire lines of unknowns, perpendicular to the direction of the anisotropy. This is achieved with block relaxation, for example with the block-Jacobi method.

After having introduced the two main strategies for the development of multigrid methods, we now divide anisotropic linear systems into two classes.

1. The first class contains matrices where anisotropy occurs along coordinate axes such as the matrices from Example 3. Multigrid methods will be described in a formal way using the notation of generating functions. This gives us a different view on this type of anisotropic problems, which is known from the solution of partial differential equations.
2. This rigorous treatment enables us to carry over the results to the second class of anisotropic problems, where anisotropy occurs in other directions. These systems are more difficult to solve, because in general, the two-level Toeplitz or algebra structure is lost on coarser levels. However, in some cases,

the matrix structure can be retained, for example in the circulant case. Then, the methods for the first class of anisotropic systems can be directly carried over. For the other matrix classes, they still work as a heuristic.

In Sections 4.2 and 4.3, we develop multigrid methods for anisotropic systems belonging to the first class. In Section 4.2, the methods are based on semicoarsening, whereas in Section 4.3 line smoothers are used. Sections 4.4 and 4.5 contain methods for systems with anisotropy in other directions, the former methods based on semicoarsening, the latter methods based on line smoothing.

4.2 Anisotropy along coordinate axes: Semicoarsening

One possible way to get rid of the problems described in the previous section is to use semicoarsening in the direction perpendicular to the anisotropy. The smoother is chosen to be a pointwise one such as the damped Jacobi method. We start with a description of the two-grid method and with theoretical results on the reduction of anisotropy. Then, the two-grid method is extended to a multigrid method, where reduction of anisotropy is obtained on each level. Convergence proofs are given for the two-grid method and for the W -cycle based multigrid method. First, this is done for the trigonometric algebra case, extending results from [99, 103, 28]. Then, similar results are obtained for the Toeplitz case, extending results from [109]. At the end of this section, we present several numerical examples to illustrate the behavior of the semicoarsening-based multigrid methods. The main results of this section are summarized in the article [55].

4.2.1 A two-grid method with semicoarsening

Since we solve two-level Toeplitz or trigonometric matrix algebra systems, we wish to describe semicoarsening in terms of generating functions. Let us assume that we have anisotropy along the x -axis (e.g. a function such as f from Example 3 with $\alpha = 0.01$), and that coarsening is done in y -direction only. $R = B \cdot E$ is the product of the following two matrices. B is the two-level structured matrix corresponding to a function which is chosen to match, for each $x \in [-\pi, \pi]$, possible zeros of f . If f has a zero of order two at the origin, the simplest choice is

$$b(x, y) = 1 + \cos(y) \quad , \quad (4.2.1)$$

corresponding to a matrix of the form

$$B = \begin{pmatrix} 1 & 0.5 & & & \\ 0.5 & 1 & 0.5 & & \\ & \ddots & \ddots & \ddots & \\ & & 0.5 & 1 & 0.5 \\ & & & 0.5 & 1 \end{pmatrix} \otimes I_n \quad (4.2.2)$$

or its BCCB or two-level DCT-III equivalent. For zeros of higher order, b is chosen to be a power of the function from (4.2.1), because, for all points on the x -axis,

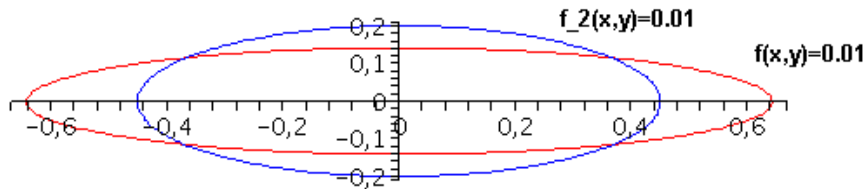


Figure 4.3: Level curves $f(x, y) = 0.01$ and $f_2(x, y) = 0.01$ obtained by one semicoarsening step with f from Example 3 and $\alpha = 0.01$

a condition such as (3.3.24) must be satisfied. For all choices of b , the product $\hat{A} = B^T \cdot A_{\mathbf{n}}[f] \cdot B$ (or $\hat{f}(x, y) = f(x, y) \cdot b(x, y)^2$ in terms of functions) is computed as in the isotropic case. The elementary restriction matrix E is chosen to be

$$E_{\mathbf{n}} = E_{n_1} \otimes I_{n_2} \quad (4.2.3)$$

with the one-dimensional elementary restriction matrix E_{n_1} defined in Chapter 3. Translated to generating functions, this becomes

$$f_2(x, y) = \frac{1}{2} \left(\hat{f}\left(x, \frac{y}{2}\right) + \hat{f}\left(x, \frac{y}{2} + \pi\right) \right) \quad (4.2.4)$$

or its tau or DCT-III equivalent. The result is a coarse grid matrix A_C with half as many blocks as $A_{\mathbf{n}}[f]$, but with the same block size. In the Toeplitz case, A_C is BTTB if n_1 is odd and b from (4.2.1) is used. For general n_1 , additional cutting (as suggested in [99, 3]) has to be applied. If anisotropy occurs along the y -axis, semicoarsening is done in x .

Why is this approach superior to the standard coarsening method? Since anisotropy occurs along coordinate axes, we have to focus on two of the difficulties described in Section 4.1.2. The problem that f is almost zero along a whole axis is overcome by coarsening in only one direction, and therefore by treating the other variable as if it were a constant. With the same argument, we get rid of the problem with the zeros at mirror points, because they are all located on the x -axis.

The use of semicoarsening is a good idea for all types of anisotropic problems, not just for the extrem case described above, because it yields a matrix A_C which is less anisotropic than $A_{\mathbf{n}}[f]$. Figure 4.3 suggests that generating functions can be used to illustrate this property. Whereas full coarsening does not change the degree of anisotropy, semicoarsening leads to level curves on the coarser grid which are less flat. This fact can be described more formally with the following definition.

Definition 2

Let $f(x, y) = c$ be a level curve of f with a sufficiently small, positive real number c , and let f_2 be the function computed in (4.2.4). We consider the points $(x_F(c), 0)$

and $(0, y_F(c))$, where the level curve of f intersects the coordinate axes for positive $x_F(c)$ and $y_F(c)$. $x_C(c)$ and $y_C(c)$ are the analogues on the coarser level, i.e. for the curve $f_2(x, y) = c$. For the sake of abbreviation, let us omit the parameter c , i.e. let us denote for example $x_F = x_F(c)$. The ratios $r_F = \frac{x_F}{y_F}$ and $r_C = \frac{x_C}{y_C}$ are used as a measure to describe the degree of anisotropy along the coordinate axes x and y for small c .

For the functions in Example 3, we observe that r_F is reduced by a factor 2 after one semicoarsening step, independent of α and for $c \ll 1$. In the following, we prove that this property holds in a more general context.

Theorem 10

Let f be a nonnegative generating function with a zero of order 2 at the origin which is of the form

$$f(x, y) = [\lambda_1(1 - \cos(x)) + \lambda_2(1 - \cos(y))] \cdot h(x, y) \quad (4.2.5)$$

with $h(x, y) > 0$ and $\lambda_1, \lambda_2 > 0$. Let f_2 be the function obtained by one semicoarsening step with b from (4.2.1). Let $r_F = \frac{x_F}{y_F}$ and $r_C = \frac{x_C}{y_C}$ be the ratios described above for f and f_2 , respectively. Then the degree of anisotropy is reduced by a factor 2, i.e. $\frac{r_F}{r_C} \rightarrow 2$ for $c \rightarrow 0$.

Proof: 1. Computation of r_F :

Since $h(x, y) > 0$, we can assume that $h_0 := h(0, 0)$ and $h_\pi := h(0, \pi)$ are bounded away from 0. f is approximated in the neighborhood of $(0, 0)$ by the following Taylor expansion with terms of order at most 2 :

$$f(x, y) \doteq (\lambda_1 \frac{x^2}{2} + \lambda_2 \frac{y^2}{2}) \cdot h_0 . \quad (4.2.6)$$

With this approximation, x_F and y_F can be computed as follows

$$\begin{aligned} f(x_F, 0) \doteq c &\Leftrightarrow \lambda_1 \frac{x_F^2}{2} h_0 \doteq c \\ f(0, y_F) \doteq c &\Leftrightarrow \lambda_2 \frac{y_F^2}{2} h_0 \doteq c . \end{aligned} \quad (4.2.7)$$

This leads to the ratio

$$r_F = \frac{x_F}{y_F} \doteq \sqrt{\frac{\lambda_2}{\lambda_1}} . \quad (4.2.8)$$

2. Computation of r_C :

First, we compute \hat{f} with b from (4.2.1) :

$$\begin{aligned} \hat{f}(x, y) = [\lambda_1(1 - \cos(x))(\frac{3}{2} + 2 \cos(y) + \frac{1}{2} \cos(2y)) \\ + \frac{1}{2} \lambda_2(1 + \cos(y) - \cos(2y) - \cos(y) \cos(2y))] \cdot h(x, y) . \end{aligned} \quad (4.2.9)$$

With the abbreviations $\tilde{h}_0 := h(x, y/2)$ and $\tilde{h}_\pi := h(x, y/2 + \pi)$ and with (4.2.4) and (4.2.9) we obtain the following function f_2 :

$$\begin{aligned} f_2(x, y) &= \frac{\lambda_1}{2}(1 - \cos(x)) \left(\frac{3}{2}(\tilde{h}_0 + \tilde{h}_\pi) + 2 \cos\left(\frac{y}{2}\right)(\tilde{h}_0 - \tilde{h}_\pi) + \frac{1}{2} \cos(y)(\tilde{h}_0 + \tilde{h}_\pi) \right) \\ &+ \frac{\lambda_2}{4} \left((\tilde{h}_0 + \tilde{h}_\pi) + \cos\left(\frac{y}{2}\right)(\tilde{h}_0 - \tilde{h}_\pi) - \cos(y)(\tilde{h}_0 + \tilde{h}_\pi) - \cos\left(\frac{y}{2}\right) \cos(y)(\tilde{h}_0 - \tilde{h}_\pi) \right). \end{aligned} \quad (4.2.10)$$

With Taylor expansion of the cosine terms at $(0, 0)$, and with approximation of \tilde{h}_0 and \tilde{h}_π by h_0 and h_π this becomes

$$\begin{aligned} f_2(x, y) &\doteq \frac{\lambda_1 x^2}{4} \left(\frac{3}{2}(h_0 + h_\pi) + 2(h_0 - h_\pi) + \frac{1}{2}(h_0 + h_\pi) \right) \\ &+ \frac{\lambda_2 y^2}{4} \left((h_0 + h_\pi) + \left(1 - \frac{y^2}{8}\right)(h_0 - h_\pi) - \left(1 - \frac{y^2}{2}\right)(h_0 + h_\pi) - \left(1 - \frac{y^2}{2} - \frac{y^2}{8}\right)(h_0 - h_\pi) \right) \\ &= \lambda_1 x^2 h_0 + \frac{\lambda_2}{4} y^2 h_0. \end{aligned} \quad (4.2.11)$$

With this approximation of f_2 we can compute r_C in the same way as r_F above:

$$\begin{aligned} f_2(x_C, 0) &\doteq c \Leftrightarrow \lambda_1 x_C^2 h_0 \doteq c \\ f_2(0, y_C) &\doteq c \Leftrightarrow \frac{\lambda_2}{4} y_C^2 h_0 \doteq c \\ r_C &= \frac{x_C}{y_C} \doteq \sqrt{\frac{\lambda_2}{4\lambda_1}} \doteq \frac{1}{2} r_F. \quad \blacksquare \end{aligned} \quad (4.2.12)$$

Since this theorem is stated purely in terms of generating functions, it holds for all structured matrix classes treated in this work.

4.2.2 Extension to a multigrid method

We wish to develop a multigrid method for the solution of anisotropic systems which combines semicoarsening and full coarsening steps. Therefore, we have to state a criterion for the choice between the two different coarsening strategies. Since we know that the system becomes less anisotropic with each semicoarsening step, we can use a straightforward heuristic: *Apply semicoarsening until the system is not anisotropic anymore, and then switch to full coarsening.* So far, we have proved that the first semicoarsening step reduces the degree of anisotropy by a factor 2. This shall now be generalized to more than two grids by using the result on generating functions and their level curves which was obtained in the previous subsection. If the matrix $A_{\mathbf{n}}[f]$ is anisotropic, level curves are flat. The following theorem shows that Theorem 10 can be applied recursively, and therefore that each semicoarsening step reduces the ratio $\frac{x_F}{y_F}$ exactly by a factor 2.

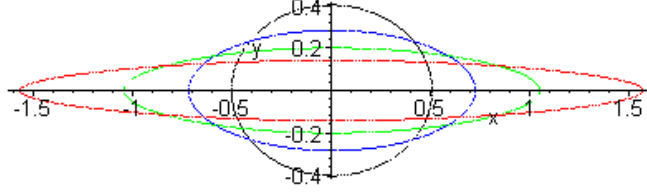


Figure 4.4: Three steps of semicoarsening applied to the system from Example 3 with $\alpha = 0.01$

Theorem 11

Let f be a nonnegative generating function with a zero of order 2 at the origin which is of the form (4.2.5). Let $r_F = \frac{x_F}{y_F}$ be the ratio of the intersection points on the finest level.

Then, the function f_2 obtained after one semicoarsening step is also of the form (4.2.5). Therefore, Theorem 10 can be applied recursively, and each semicoarsening step reduces r_F by a factor 2 for small c .

Proof: The coarse grid function $f_2(x, y)$, which has been computed in (4.2.10) in the proof of Theorem 10, can be slightly rewritten:

$$\begin{aligned}
 f_2(x, y) = & \lambda_1(1 - \cos(x)) \left(\frac{3}{4}(\tilde{h}_0 + \tilde{h}_\pi) + \cos\left(\frac{y}{2}\right)(\tilde{h}_0 - \tilde{h}_\pi) + \frac{1}{4} \cos(y)(\tilde{h}_0 + \tilde{h}_\pi) \right) \\
 & + \frac{\lambda_2}{4}(1 - \cos(y)) \left((\tilde{h}_0 + \tilde{h}_\pi) + \cos\left(\frac{y}{2}\right)(\tilde{h}_0 - \tilde{h}_\pi) \right) .
 \end{aligned}
 \tag{4.2.13}$$

In the neighborhood of the origin, f_2 is approximated by Taylor expansion and by replacing \tilde{h}_0 and \tilde{h}_π with h_0 and h_π :

$$f_2(x, y) \doteq \lambda_1(1 - \cos(x))2h_0 + \frac{\lambda_2}{4}(1 - \cos(y))2h_0 .
 \tag{4.2.14}$$

Thus, f_2 is of the form (4.2.5), and Theorem 10 can be applied recursively. ■

Remark 9 Theorem 11 has the consequence that level curves become less flat on each level. If we start with the ratio r_F on the finest level, we need $\log_2(r_F)$ semicoarsening steps until the level curves are almost like circles, i.e. until the ratio is close to 1. Figure 4.4 shows the result of three steps of semicoarsening applied to the function f , whose level curve is the flattest of the four. Since $r_F = 10$ on the finest level, a multigrid method following our heuristic should start with three semicoarsening steps followed by full coarsening.

4.2.3 Convergence results for trigonometric matrix algebras

After having presented a multigrid method for matrices from trigonometric algebras and for Toeplitz matrices whose generating functions are anisotropic with

isolated zeros, we wish to give a more theoretical analysis in the following. We start with a two-grid convergence result for trigonometric matrix algebras which is based on Corollary 1. The result is achieved by extending two-grid convergence proofs for two-level tau [99], DCT-III [28], DST-III, and circulant matrices [103] which were stated by Serra et al. for isotropic generating functions with isolated zeros. In the anisotropic case, the generating function is allowed to be zero or very small along a whole line, which is located in parallel to one of the coordinate axes. Since in the tau, DCT-III, and DST-III case, a zero at $x_0 \neq 0$ implies that f also has a zero at $-x_0$, f has two lines of small values, unless these lines coincide at one of the axes.

Remark 10 In order to prove convergence, we impose the following conditions on the functions b_j . For every point on these lines,

- conditions (3.3.22) and (3.3.23) must be satisfied in the tau and circulant case,
- conditions (3.3.34) and (3.3.35) must hold in the DCT-III and DST-III case.

For the two-grid proof, these conditions are required to hold only on the finest level, i.e. for f and b . Under these conditions, the following theorem proves convergence for anisotropy in y -direction. The case of anisotropy in x -direction follows immediately by interchanging x and y . Furthermore, let us assume that f has a zero at the origin, i.e. that the line of possible zeros is one of the coordinate axes.

Theorem 12

Let $A_{\mathbf{n}}[f]$ be a two-level tau, circulant, DCT-III, or DST-III matrix of size \mathbf{n} corresponding to a nonnegative trigonometric polynomial f . Let the smoother be the damped Richardson or Jacobi method and the restriction chosen to be $R = B \cdot E$ such that E is defined in (4.2.2) and B satisfies the criteria stated in Remark 10. Then, conditions (3.1.17) and (3.1.18) are satisfied, and the two-grid method converges with a convergence factor $\|TG\|_1 < 1$.

Proof: The smoothing condition for these matrix algebras has been proved in [99, 28, 103]. We prove the correcting condition first for the circulant case, extending the result from [103]. The matrix $A_{\mathbf{n}}[f]$ is of size $\mathbf{n} = n_1 \cdot n_2$ with n_1 and n_2 being even, and the coarse grid matrix A_C of size $n_1 \cdot k_2$ with $k_2 = n_2/2$. Following the strategy from [103], we prove the correcting condition by showing that (3.1.18) holds if we choose, for each $\mathbf{x} \in \mathbb{C}^{\mathbf{n}}$, the following \mathbf{y} :

$$\mathbf{y} = [RR^H]^{-1}R\mathbf{x} \quad . \tag{4.2.15}$$

In [103], it is shown that this is true if there exists $\gamma > 0$ such that

$$I - R^H[RR^H]^{-1}R \leq \frac{\gamma}{a}A_{\mathbf{n}}[f] \tag{4.2.16}$$

where ν_1 denotes the respective block and $\nu_2 = \lfloor n_2/2 \rfloor$. Since by Remark 10, f cannot vanish for $y = \pi/2$, (4.2.21) is satisfied for each block.

For DCT-III and DST-III matrices, the proof is similar, extending the result from [28] and from Chapter 3. If the line of small values of f is not located at $x = \pi$, the result is the same. In the case of $x = \pi$, the zero of f_2 is of higher order. However, the second condition in Remark 10 guarantees that the correcting condition also holds in this case. ■

4.2.4 Convergence results for Toeplitz matrices

In this section, the convergence proofs of [24] and [109] shall be carried over to anisotropic BTTB systems. For the convergence proofs, let us assume that anisotropy occurs along the y -axis, because notation is slightly simpler in this case. Anisotropy along the x -axis is treated similarly. If anisotropy occurs along the y -axis, i.e. if $f(x, y)$ is small for $x = 0$ and $y \in [-\pi, \pi]$, coarsening is done only in x . Instead of (4.2.1) and (4.2.2), we use $b(x, y) = 1 + \cos(x)$ and

$$R_{\mathbf{n}}^T = I_{n_1} \otimes R_{n_1}^T \quad \text{with} \quad R_{n_1}^T = \begin{pmatrix} 0.5 & 1 & 0.5 & & \\ & 0.5 & 1 & 0.5 & \\ & & \ddots & \ddots & \\ & & & & \ddots \end{pmatrix}. \quad (4.2.22)$$

In this case, f is allowed to be zero on the whole line $x = 0$. Thus, (3.4.1) is replaced by

$$\min_{(x,y) \in [-\pi, \pi]^2} \frac{f(x, y)}{1 - \cos x} = C > 0. \quad (4.2.23)$$

The following theorem proves convergence of the two-level method.

Theorem 13

Let $T_{\mathbf{n}}[f]$ be a positive definite BTTB matrix whose generating function is real-valued even and satisfies (4.2.23). Let $t_{0,0}$ denote the entries in its main diagonal. Moreover, let the prolongation matrix $R_{\mathbf{n}}^T$ be given by (4.2.22), and let the smoother be the damped Jacobi method.

Then, the convergence factor of the two-level method is uniformly bounded below 1 independent of n_1 and n_2 . The following estimate for the convergence factor holds:

$$\|TG\|_1 \leq \sqrt{1 - \frac{C}{2\rho(T_{\mathbf{n}}[f])}}. \quad (4.2.24)$$

Proof: The proof of the postsmoothing condition (3.1.17) is the same as in [24] and [109]. Therefore, we only have to prove the correcting condition (3.1.18). First assume that $n_2 = 2k + 1$ with k being the size of the blocks on the coarse level. Following [24], we define, for any

$$e = (e_1, e_2, \dots, e_{n_1})^T = (e_{1,1}, \dots, e_{1,n_2}, e_{2,1}, \dots, e_{2,n_2}, \dots, e_{n_1,1}, \dots, e_{n_1,n_2})^T \in \mathbb{R}^{\mathbf{n}},$$

the vector

$$e_C = (\tilde{e}_1, \tilde{e}_2, \dots, \tilde{e}_{n_1})^T = (\tilde{e}_{1,1}, \dots, \tilde{e}_{1,k}, \tilde{e}_{2,1}, \dots, \tilde{e}_{2,k}, \dots, \tilde{e}_{n_1,1}, \dots, \tilde{e}_{n_1,k})^T \in \mathbb{R}^{n_1 k},$$

where $\tilde{e}_{i,j} = e_{i,2j}$. If $j \leq 0$ or $j > n_2$, we set $e_{i,j} = 0$ in order to complete the notation. For this special choice of e_C , we try to find an upper bound for $\|e - R_{\mathbf{n}}^T e_C\|_0^2$ of the form $\beta\|e\|_1$ with β independent of e . Then, the correcting condition would follow immediately. With $R_{\mathbf{n}}^T$ from (4.2.22), the following upper bound is found:

$$\begin{aligned} \|e - R_{\mathbf{n}}^T e_C\|_0^2 &= \|e - (I_{n_1} \otimes R_{n_2}^T) e_C\|_0^2 = \sum_{i=1}^{n_1} \|e_i - R_{n_2}^T \tilde{e}_i\|_0^2 \\ &\leq \sum_{i=1}^{n_1} t_{0,0} \langle e_i, T_{n_2}[1 - \cos(x)] e_i \rangle = t_{0,0} \langle e, (I_{n_1} \otimes T_{n_2}[1 - \cos(x)]) e \rangle . \end{aligned} \quad (4.2.25)$$

The n_1 one-dimensional inequalities hold because of a result in [24]. It remains to show that there exists a β independent of e such that

$$t_{0,0} \langle e, (I_{n_1} \otimes T_{n_2}[1 - \cos x]) e \rangle \leq \beta \langle e, T_{\mathbf{n}}[f] e \rangle , \quad \forall e \in \mathbb{R}^{\mathbf{n}} . \quad (4.2.26)$$

Because of Theorem 1 and Remark 1, condition (4.2.23) has the consequence

$$C \cdot (I_{n_1} \otimes T_{n_2}[1 - \cos x]) \leq T_{\mathbf{n}}[f] . \quad (4.2.27)$$

This implies that (4.2.26) is satisfied with

$$\beta = \frac{t_{0,0}}{C} ,$$

and the correcting condition is proved for the case $n_2 = 2k + 1$. For $n_2 = 2k$, the vector e is embedded into the vector \hat{e} of size $n_1 \hat{n}_2 = n_1(2k + 1)$ by filling zeros into the additional positions. Then the correction condition also holds because of

$$\|e - R_{\mathbf{n}}^T e_C\|_0^2 \leq \|\hat{e} - \hat{R}_{n_1 \hat{n}_2}^T e_C\|_0^2 \quad (4.2.28)$$

and

$$\langle \hat{e}, I_{n_1} \otimes T_{\hat{n}_2}[1 - \cos x] \hat{e} \rangle = \langle e, I_{n_1} \otimes T_{n_2}[1 - \cos x] e \rangle .$$

■

To obtain a result for the multilevel method we prove that if (4.2.23) holds on some level, it also holds on the next coarser level after one semicoarsening step. Again, the proof is obtained by extending the one from [109]. Let T^h and T^H denote the BTTB matrices on the finer and on the coarser level, and n_h and n_H the respective block sizes. Since semicoarsening is used, the number of blocks is constant on all levels. Furthermore, $t_{0,0}^h$ and $t_{0,0}^H$ are the main diagonal entries of T^h and T^H .

Theorem 14

Let T^h be a positive definite BTTB matrix of size mn_h satisfying

$$T^h \geq \frac{t_{0,0}^h}{\beta^h} T_{mn_h}[1 - \cos x] \quad (4.2.29)$$

for some β^h from (3.1.18) independent of mn_h , and let the restriction be defined with (4.2.22). Then

$$T^H \geq \frac{t_{0,0}^H}{\beta^H} T_{mn_H} [1 - \cos x] \quad (4.2.30)$$

with

$$\beta^H = 2 \frac{t_{0,0}^H \beta^h}{t_{0,0}^h}. \quad (4.2.31)$$

This implies that Theorem 13 can be applied on each level, stating that the correcting condition is also satisfied on coarser levels. If q levels are used, the following estimate for the convergence factor holds:

$$\|TG^q\|_1 \leq \sqrt{1 - \frac{\alpha^q}{\beta^q}} = \sqrt{1 - \frac{\alpha^h}{4^{q-1} \beta^h}}. \quad (4.2.32)$$

Proof: Define the $(n_H + 1)$ -by- n_H matrix $K = \frac{1}{2} \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & \ddots & \ddots \end{bmatrix}$. Then there exists a permutation matrix Q such that

$$Q \cdot R_{mn_n}^T = I_m \otimes \begin{pmatrix} I_{n_H} \\ K \end{pmatrix} \quad (4.2.33)$$

and

$$Q \cdot T_{mn_n} [1 - \cos x] \cdot Q^T = I_m \otimes \begin{pmatrix} I_{n_H} & -K \\ -K^T & I_{n_H+1} \end{pmatrix}. \quad (4.2.34)$$

With these prerequisites, we can derive the lower bound (4.2.30) for T^H . By (4.2.23) and (4.2.29), we have

$$T^H = R_{mn_n} T^h R_{mn_n}^T \geq \frac{t_{0,0}^h}{\beta^h} R_{mn_n} T_{mn_n} [1 - \cos x] R_{mn_n}^T. \quad (4.2.35)$$

With (4.2.33) and (4.2.34), the right-hand side can be simplified in the following way:

$$\begin{aligned} & \frac{t_{0,0}^h}{\beta^h} \left[I_m \otimes \left((I_{n_H}, K^T) \begin{pmatrix} I_{n_H} & -K \\ -K^T & I_{n_H+1} \end{pmatrix} \begin{pmatrix} I_{n_H} \\ K \end{pmatrix} \right) \right] \\ &= \frac{t_{0,0}^h}{\beta^h} [I_m \otimes (I_{n_H} - K^T K)] \\ &= \frac{t_{0,0}^h}{2\beta^h} [I_m \otimes T_{n_H} [1 - \cos x]] \\ &= \frac{t_{0,0}^h}{2\beta^h} T_{mn_H} [1 - \cos x], \end{aligned} \quad (4.2.36)$$

where the third line follows from the second by the definition of K . (4.2.30) and (4.2.31) are immediate consequences of (4.2.36). \blacksquare

coarsening	$\mathbf{n}=(2^6-1)^2$	$\mathbf{n}=(2^7-1)^2$	$\mathbf{n}=(2^8-1)^2$
y,xy,xy,xy,xy	170	> 200	> 200
y,y,y,xy,xy	12	19	23
y,y,y,y,y	7	7	7

Table 4.1: Iteration numbers for $T_{\mathbf{n}}[f]$ with f from Example 3 with $\alpha = 0.001$

Remark 11 Theorem 14 implies that if (4.2.23) holds on some level, it also holds on the next coarser level. Therefore, Theorem 13 can be applied on each level. This property is known as level independency. From [112] we know that level independency implies optimal convergence of the multigrid method with W-cycles, but not necessarily optimal convergence of the V-cycle based method.

Remark 12 If the anisotropy is moderate, our heuristic suggests the use of a multigrid method which consists of some semicoarsening steps followed by full coarsening on the coarser levels. In this case, we can combine the convergence results on full coarsening from [109] with our results. This is done by computing β^H in (4.2.31) as in [109] if full coarsening is used at some level. As a consequence we obtain an estimate such as (4.2.32).

4.2.5 Numerical results

After having analyzed our multigrid methods theoretically concerning reduction of anisotropy and convergence, we perform numerical tests to illustrate the optimal convergence behavior. The methods consist of a certain number of semicoarsening steps followed by full coarsening steps. The damped Jacobi method or the Gauss-Seidel method is used as a smoother. Theorem 10 states that the ratio r_F is reduced by a factor 2 with each semicoarsening step. Following our heuristic, we apply semicoarsening until this ratio is close to one, i.e. until the system is not anisotropic anymore, and then proceed with standard coarsening. For the function $f(x, y)$ from Example 3 with $\alpha = 0.001$, the ratio r_F is 31.62, which means that on the five finest levels, semicoarsening should be performed. In our numerical experiments, we test different coarsening strategies on the matrix $T_{\mathbf{n}}[f]$, which belongs both to the two-level Toeplitz and the two-level tau class. The first strategy (denoted y,xy,xy,xy,xy) consists of one semicoarsening step, followed by four full coarsening steps, the second (denoted y,y,y,xy,xy) of three semicoarsening steps and two full coarsening steps, and the third (denoted y,y,y,y,y) of five semicoarsening steps. Table 4.1 shows the number of V-cycle iterations our method requires until the residual is smaller than 10^{-6} . These results are obtained with one pre-smoothing step and one post-smoothing step on each level with the symmetric Gauss-Seidel method. If the damped Jacobi method is used, the difference between the coarsening strategies is even more striking. Since we are also interested in mul-

coarsening	$\mathbf{n}=(2^5-1)^2$	$\mathbf{n}=(2^6-1)^2$	$\mathbf{n}=(2^7-1)^2$
y,xy,xy,xy,xy	76	182	> 200
y,y,y,xy,xy	9	11	15
y,y,y,y,y	9	9	9

Table 4.2: Iteration numbers for $T_{\mathbf{n}}[f]$ with f from (4.1.3) with $\alpha = 0.001$

α	coarsening	$\mathbf{n}=(2^6)^2$	$\mathbf{n}=(2^7)^2$	$\mathbf{n}=(2^8)^2$
0.02	y,xy,xy,xy,xy	24	23	23
0.02	y,y,y,xy,xy	5	5	5
0.02	y,y,y,y,y	8	8	8
0.001	y,xy,xy,xy,xy	> 200	> 200	> 200
0.001	y,y,y,xy,xy	27	27	26
0.001	y,y,y,y,y	5	5	5

Table 4.3: Iteration numbers for $\tilde{C}_{\mathbf{n}}[f]$ with f from Example 3 with $\alpha = 0.02$ and $\alpha = 0.001$

tilevel Toeplitz matrices which are not necessarily connected with PDEs, we take a look at the BTTB matrices with f from (4.1.3), which are not sparse. Again, we choose $\alpha = 0.001$ and use one iteration of the Gauss-Seidel method as pre- and postsmoother. Table 4.2 shows a similar behavior of the multigrid method as we have observed in Example 3. Again, the damped Jacobi smoother leads to similar results. It is cheaper concerning computational cost, but the V-cycle requires a few more iterations.

Multigrid methods for circulant matrices are usually only efficient if the matrix is sparse, because for dense matrices, the inverse can be directly computed in $O(\mathbf{n} \log(\mathbf{n}))$ with the FFT. The circulant matrix $C_{\mathbf{n}}[f]$ with f from Example 3, however, is singular. Therefore, we replace it by its Strang correction. This means we add $\frac{1}{(\mathbf{n})^2} \cdot I_{\mathbf{n}}$ to $C_{\mathbf{n}}[f]$, which corresponds to a shift of the grid points and results in the ill-conditioned matrix $\tilde{C}_{\mathbf{n}}[f]$. In our numerical experiments, we use a V-cycle with six levels and the damped Jacobi method as a smoother. For $\alpha = 0.02$, the value r_F suggest that we use three steps of semicoarsening followed by full coarsening. For the stronger anisotropy with $\alpha = 0.001$, five semicoarsening steps are supposed to yield optimal results. The results of the numerical calculations in Table 4.3 confirm that these suggestions lead to the fastest convergence in both cases.

Furthermore, we wish to give results for the DCT-III algebra. We use the function $f(x, y)^2$ with f from Example 3. In this case, the proper choice of the number of semicoarsening steps is even more important. Again, our heuristic

α	coarsening	$\mathbf{n}=(2^6)^2$	$\mathbf{n}=(2^7)^2$	$\mathbf{n}=(2^8)^2$
0.01	y,xy,xy,xy,xy	> 200	> 200	> 200
0.01	y,y,y,xy,xy	6	6	6
0.01	y,y,y,y,y	10	10	10
0.001	y,xy,xy,xy,xy	> 200	> 200	> 200
0.001	y,y,y,xy,xy	> 200	> 200	> 200
0.001	y,y,y,y,y	7	6	6

Table 4.4: Iteration numbers for $R_{\mathbf{n}}[f^2]$ with f from Example 3 with $\alpha = 0.01$ and $\alpha = 0.001$

suggests the same number of semicoarsening as above. Table 4.4 contains the number of iterations. The results for the DST-III matrices corresponding to f or f^2 are very similar.

4.3 Anisotropy along coordinate axes: Line Smoothing

Anisotropic systems can be solved with a different multigrid strategy where standard coarsening from Chapter 3 can still be used. However, this requires application of an adequate smoother on each grid. First, we describe the components which are necessary to construct a two-grid method and a multigrid method. Then, we obtain convergence results concerning the smoother, and finally, we present numerical results. The main results of this section can be found in the article [56].

4.3.1 A multilevel method with line smoothing

The multigrid method presented in this chapter uses the same full coarsening strategy as the methods from Chapter 3. On each level, we choose a function b_j which satisfies (3.3.22) and (3.3.23). The function f_{j+1} on the next coarser level is computed with (3.3.5). Thus, a more sophisticated smoothing technique must be used on each level. Line smoothers such as the damped block Jacobi method smooth along a whole line of grid points, which corresponds to a block of unknowns in the solution vector. These unknowns, and therefore also the rows and columns of the matrix, are permuted and then grouped into blocks. Instead of inverting the main diagonal as in the pointwise Jacobi method, the diagonal blocks are inverted. However, for most of our matrix classes, almost all diagonal blocks are the same. Figure 4.5 shows how the blocks of the matrix are built, i.e. how the rows and columns of A_j must be permuted for smoothing. Each point in the pictures corresponds to one unknown in the solution vector or to one line or column of the matrix. In the context of PDEs, the points in the picture can be interpreted as grid points. If anisotropy occurs along the y -axis, the blocks

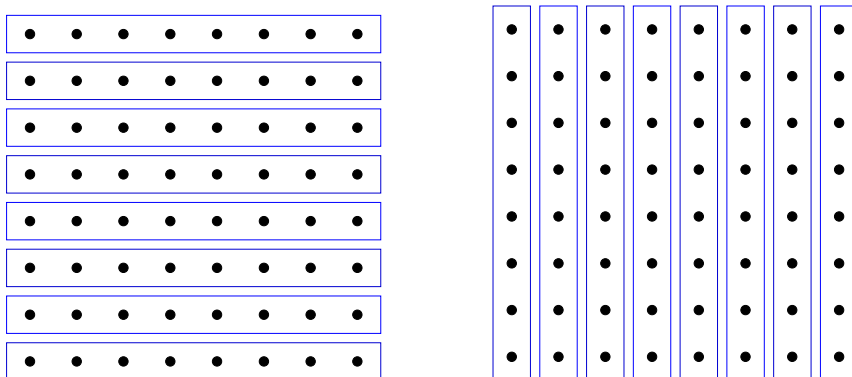


Figure 4.5: Partitioning of the original matrix into blocks

of A_j must be constructed as it is shown in the left picture. No permutation is necessary, and all blocks have size n_2 . The matrix D_j , which contains the diagonal blocks of A_j , corresponds to a generating function g_j . If f_j can be written as a trigonometric sum, g_j is obtained from f_j by eliminating all terms of the sum containing y . If anisotropy occurs along the x -axis, the right picture applies. The rows and the columns are permuted with the vector

$$(1, n_2 + 1, 2n_2 + 1, \dots, (n_1 - 1)n_2 + 1, 2, n_2 + 2, 2n_2 + 2, \dots, (n_1 - 1)n_2 + 2, \dots, n_2 - 1, n_2 + n_2 - 1, 2n_2 + n_2 - 1, \dots, (n_1 - 1)n_2 + n_2 - 1) .$$

This permutation can be interpreted as a change of the variables, i.e. as a new coordinate system, where x and y are interchanged. This idea of permutation and changing variables will become more important in Section 4.4. The blocks in this case are of size m . Since full coarsening is used, the degree of anisotropy remains the same on each level. Therefore, block Jacobi smoothing is applied on all levels. In theory, other block smoothers could be used instead of the block Jacobi method. For example the block version of the Gauss-Seidel method leads to excellent convergence results. However, such a method requires not only inversion of the diagonal block, but also of all blocks below the main diagonal. This is computationally too expensive for an efficient algorithm.

4.3.2 Convergence results

The following theorem shows that the block Jacobi method indeed satisfies the smoothing conditions (3.1.16) and (3.1.17) if anisotropy occurs along the y -axis. The condition for anisotropy in x -direction follows immediately. The following theorem proves the conditions on the finest level. Later we will show that they also hold on coarser levels. Moreover, we assume that f has a zero at the origin of order two.

Theorem 15

Let f be a nonnegative generating function with a zero of order 2 at the origin which is of the form

$$f(x, y) = [(1 - \cos(x)) + \alpha(1 - \cos(y))] \cdot h(x, y) \quad (4.3.1)$$

with $0 < \alpha \ll 1$ and the trigonometric polynomial h satisfying $0 < h_{\min} \leq h(x, y) \leq h_{\max} < \infty$. Let $A = A_{\mathbf{n}}[f]$ be the corresponding two-level tau, circulant, DCT-III, or DST-III matrix, and D the block diagonal matrix with the same diagonal blocks as A , corresponding to the generating function

$$g(x, y) = (1 + \alpha - \cos(x)) \cdot \tilde{h}(x) . \quad (4.3.2)$$

\tilde{h} is obtained by eliminating all terms in h containing y . Let $M_f, M_g, M_{\frac{f}{g}}$ denote the maximum values of $f, g, \frac{f}{g}$.

If $\tilde{h}(x) > 0$, then the block Jacobi method

$$x^{(k+1)} = x^{(k)} + \omega D^{-1}(b - Ax^{(k)}) ,$$

i.e. $S = I - \omega D^{-1}A$, satisfies the smoothing conditions (3.1.16) and (3.1.17). More precisely, for $0 \leq \omega \leq 2/M_{\frac{f}{g}}$, there exist nonnegative $\alpha_{pre}, \alpha_{post}$ with

$$\alpha_{pre} \leq \min \left\{ 2\omega , \frac{\omega(2 - \omega M_{\frac{f}{g}})}{(1 - \omega M_{\frac{f}{g}})^2} \right\} \quad (4.3.3)$$

$$\alpha_{post} \leq \omega(2 - \omega M_{\frac{f}{g}}) .$$

Proof: First of all, we show that the functions f , g , and f/g are bounded by M_f, M_g , and $M_{\frac{f}{g}}$. For the first two of them, this is straightforward:

$$M_f = \max_{(x,y) \in [-\pi, \pi]^2} f(x, y) = (2 + 2\alpha) \cdot h_{\max}$$

$$M_g = \max_{(x,y) \in [-\pi, \pi]^2} g(x, y) = (2 + \alpha) \cdot \tilde{h}_{\max}$$

For the third function, we estimate the quotient

$$\begin{aligned} \frac{f(x, y)}{g(x, y)} &\leq \frac{1 - \cos(x) + \alpha(1 - \cos(y))}{1 - \cos(x) + \alpha} \cdot \frac{h_{\max}}{\tilde{h}_{\min}} \\ &\leq \left(1 + \frac{-\alpha \cos(y)}{1 - \cos(x) + \alpha} \right) \cdot \frac{h_{\max}}{\tilde{h}_{\min}} \\ &\leq \left(1 + \frac{-\alpha \cdot (-1)}{\alpha} \right) \cdot \frac{h_{\max}}{\tilde{h}_{\min}} \leq 2 \cdot \frac{h_{\max}}{\tilde{h}_{\min}} . \end{aligned}$$

This implies that

$$M_{\frac{f}{g}} = \max_{(x,y) \in [-\pi, \pi]^2} \frac{f(x, y)}{g(x, y)} = 2 \cdot \frac{h_{\max}}{\tilde{h}_{\min}} .$$

Moreover, f/g has the minimum value $m_{\frac{f}{g}} = 0$.

In the following, we prove the smoothing conditions with $Y = D^{-1}$ (see Remark 3). It should be noted that the proof technique of translating the conditions of Theorem 6 into function inequalities was introduced in [99]. Here, we use this technique for tau, circulant, and DCT-III matrices. In [2], the authors use a similar proof technique with $Y = I$ for the damped Richardson method. With our choice of Y , the presmoothing condition (3.1.16) can be written

$$SAS \leq A - \alpha_{pre}SA^2D^{-1}S, \quad (4.3.3)$$

which is equivalent to

$$(I - \omega D^{-1}A)A(I - \omega D^{-1}A) \leq A - \alpha_{pre}(I - \omega D^{-1}A)D^{-1}A^2(I - \omega D^{-1}A). \quad (4.3.4)$$

This is implied by the function inequality

$$(1 - \omega \cdot \frac{f}{g})f(1 - \omega \cdot \frac{f}{g}) \leq f - \alpha_{pre}(1 - \omega \cdot \frac{f}{g})\frac{f^2}{g}(1 - \omega \cdot \frac{f}{g}), \quad (4.3.5)$$

and because of $\frac{f}{g} \geq 0$ by

$$1 + \alpha_{pre} \cdot \frac{f}{g} \leq \frac{1}{(1 - \omega \cdot \frac{f}{g})^2}. \quad (4.3.6)$$

There exists a nonnegative α_{pre} in (4.3.6) only for $0 \leq \omega \leq \frac{2}{M_{\frac{f}{g}}}$. Since f/g can take values between 0 and $M_{\frac{f}{g}}$, (4.3.6) holds if

$$1 + \alpha_{pre} \cdot t \leq \frac{1}{(1 - \omega \cdot t)^2} \quad (4.3.7)$$

is true for all $0 < t \leq M_{\frac{f}{g}}$. As suggested in [2], we can deduce that this holds if $\alpha_{pre} \leq 2\omega$ for $0 \leq \omega \leq \frac{2}{M_{\frac{f}{g}}}$, and, in addition,

$$1 + \alpha_{pre} \cdot M_{\frac{f}{g}} \leq \frac{1}{(1 - \omega \cdot M_{\frac{f}{g}})^2}$$

for $\frac{1}{M_{\frac{f}{g}}} < \omega \leq \frac{2}{M_{\frac{f}{g}}}$.

The postsmoothing condition is equivalent to

$$(I - \omega D^{-1}A)A(I - \omega D^{-1}A) \leq A - \alpha_{post}D^{-1}A^2. \quad (4.3.8)$$

This inequality is translated to generating functions, and because of $f/g \geq 0$, it can be simplified to

$$(1 - \omega \cdot \frac{f}{g})^2 \leq 1 - \alpha_{post} \cdot \frac{f}{g}, \quad (4.3.9)$$

which leads to nonnegative α_{post} only for $0 \leq \omega \leq \frac{2}{M_f/g}$. (4.3.9) is satisfied if

$$(1 - \omega \cdot t)^2 \leq 1 - \alpha_{post} \cdot t \quad (4.3.10)$$

holds for $0 < t \leq M_f/g$. Similar to [2], this is shown to be true if

$$1 - \alpha_{post} M_f/g \geq (1 - \omega \cdot M_f/g)^2 . \quad \blacksquare$$

In the following corollary, we deduce optimal values for $\alpha_{pre}, \alpha_{post}, \omega$ from (4.3.3).

Corollary 5

Under the same assumptions as in Theorem 15 we obtain the following optimal values for the parameters $\alpha_{pre}, \alpha_{post}, \omega$:

1. *If one presmoothing step and no postsmoothing is performed, then*

$$\omega_{best} = \frac{3}{2 \cdot M_f/g} , \quad \alpha_{pre,best} = \frac{3}{M_f/g} .$$

2. *If one postsmoothing step and no presmoothing is performed, then*

$$\omega_{best} = \frac{1}{M_f/g} , \quad \alpha_{post,best} = \frac{1}{M_f/g} .$$

3. *If only one step of smoothing shall be performed, then the optimal rate of convergence is obtained with*

- *one presmoothing step with $\omega = \frac{3}{2 \cdot M_f/g}$ if $\frac{\alpha_{post}}{\beta} \in [0, \frac{2}{3}]$,*
- *one postsmoothing step with $\omega = \frac{1}{M_f/g}$ if $\frac{\alpha_{post}}{\beta} \in [\frac{2}{3}, 1]$.*

Proof: The proof uses the same technique as the one in [103]. The first two parts are proved using the estimates for α_{pre} and α_{post} from Theorem 15. The first inequality in (4.3.3) implies that

$$\alpha_{pre,best} = \max_{\omega \in (0, 2/M_f/g)} \left\{ \begin{array}{ll} 2\omega & \text{if } \omega \leq 3/(2M_f/g) \\ (\omega(2 - \omega M_f/g))/(1 - \omega M_f/g)^2 & \text{if } \omega > 3/(2M_f/g) \end{array} \right\} ,$$

which is obtained for $\omega = 3/(2M_f/g)$. The second inequality in (4.3.3) leads to

$$\alpha_{pre,best} = \max_{\omega \in (0, 2/M_f/g)} \omega(2 - \omega M_f/g) .$$

The third part is proved by comparing the convergence factors from Theorem 6. One presmoothing step without postsmoothing leads to

$$\sqrt{1/(1 + \alpha_{pre,best}/\beta)} = \sqrt{1/(1 + 3\alpha_{post,best}/\beta)} ,$$

whereas one postsmoothing step without presmoothing leads to

$$\sqrt{1 - \alpha_{post,best}/\beta} . \quad \blacksquare$$

Remark 13 An extension of Theorem 15 and Corollary 5 to the BTTB case seems rather difficult. For example, the implication from (4.3.4) to (4.3.5) does not hold in the Toeplitz case. We have information on the localization of the spectrum of $A, A^2, D^{-1}A$ (see [91]), but not on their linear combinations. The nontrivial structure of D poses a serious problem. Nonetheless, block Jacobi smoothing leads to fast multigrid convergence for BTTB systems.

Remark 14 In Theorem 15, the smoothing conditions have been proved on the finest level. If full coarsening is used, as described in Chapter 3, the theorem can be applied on all levels. In [3, 2, 28], it is shown that for matrix algebras, the functions on coarser grids also have a zero of the same order at the origin. Moreover, the corresponding matrices are in the same class as on finer levels. Therefore, all functions f_j can be written in the form (4.3.1), and Theorem 15 can be applied on each level.

4.3.3 Numerical results

The following numerical results are obtained from a multigrid method using standard coarsening in combination with a line smoother such as the damped block Jacobi method. If the anisotropy is very strong, this method converges extremely fast, because the block diagonal matrix of the smoother is a very good approximation of $A_n[f]$. If the problem is only mildly anisotropic, we observe the typical multigrid convergence behavior which is fairly fast and independent of the matrix size. In this case, standard coarsening still works, because the anisotropy is not so strong, and the block Jacobi method has good smoothing properties. The most difficult case for our method are problems which are quite anisotropic, e.g. when α is between 0.01 and 0.001 in our examples. However, even in this case we obtain fast convergence if we apply two block Jacobi iterations as pre- and postsmoother, although the number of V-cycle iterations increases slightly with the matrix size. Table 4.5 summarizes the results for the BTTB matrices from Example 3, comparing the number of iterations for different degrees of anisotropy. In each case, we use a V-cycle with five levels and standard coarsening. Since the diagonal blocks have to be inverted, the block Jacobi method is too expensive for dense BTTB matrices. For sparse matrices, however, their results are comparable with those obtained by semicoarsening. Furthermore, it is possible to start with some

α	$\mathbf{n}=(2^6-1)^2$	$\mathbf{n}=(2^7-1)^2$	$\mathbf{n}=(2^8-1)^2$
0.1	8	8	8
0.001	3	4	6
0.00001	2	2	3

Table 4.5: Iteration numbers for $T_{\mathbf{n}}[f]$ with f from Example 3 with different α

α	$\mathbf{n}=(2^6-1)^2$	$\mathbf{n}=(2^7-1)^2$	$\mathbf{n}=(2^8-1)^2$
0.1	5	5	5
0.001	5	5	5
0.00001	5	5	5

Table 4.6: Iteration numbers for $\tilde{C}_{\mathbf{n}}[f]$ with f from Example 3 with different α

semicoarsening steps, and then, on coarser levels, use the block Jacobi method. Such a mixture of both types of algorithms is less expensive and further improves convergence, especially for those α where standard coarsening combined with a line smoother has difficulties.

The circulant matrix corresponding to the function $f(x, y)$ from Example 3 is also solved quite efficiently with standard coarsening and the block Jacobi smoother. Table 4.6 shows the number of V-cycle iterations. For BCCB matrices, the block Jacobi method can be applied even if the diagonal blocks are not sparse, because inversion of a circulant block has a worst case complexity of $O(\mathbf{n} \log(\mathbf{n}))$.

Again, we conclude this section with numerical results for the DCT-III and DST-III algebras. The generating function is the square of $f(x, y)$ from Example 3, which is, of course, more complicated than f itself. As for the other tests, we use a V-cycle with five grids and, on each grid, one step of block Jacobi smoothing as a presmoothen and one as a postsmoothen. Table 4.7 contains the number of iterations for different degrees of anisotropy. The numbers are given for the DCT-III class, but those for the DST-III class are almost identical.

α	$\mathbf{n}=(2^6-1)^2$	$\mathbf{n}=(2^7-1)^2$	$\mathbf{n}=(2^8-1)^2$
0.1	7	7	7
0.001	6	7	7
0.00001	4	4	4

Table 4.7: Iteration numbers for $R_{\mathbf{n}}[f^2]$ with $f(x, y)$ from Example 3 and different α

4.4 Anisotropy in other directions: Semicoarsening

So far, we have developed multigrid methods for anisotropic problems where anisotropy occurs along coordinate axes. Generating functions and their level curves were used for a theoretical analysis of the methods. Now we introduce linear systems with anisotropy in other directions. In this case, neither standard multigrid from Chapter 3 nor the methods from Sections 4.2 and 4.3 work properly, and for $\alpha \rightarrow 0$, they fail completely. Semicoarsening along an axis does not help to treat anisotropy in other directions well. Moreover, we have to take into account that $f(x, y)$ has another zero at (π, π) , independent of α , which is another obstacle to a convergent multigrid method, see Remark 8. Now, we develop similar techniques as in Sections 4.2 and 4.3 for the more general case where anisotropy occurs in other directions. This shall be done for both the two-level Toeplitz class and the two-level circulant algebra. In this section, we present methods based on the use of semicoarsening and pointwise smoothers. We start with the case of anisotropy occurring in an angle of 45° to the coordinate axes. After introducing our heuristic, we present two-grid and multigrid methods. Then, we describe how the method is generalized to anisotropy in other directions and present theoretical results concerning convergence and the reduction of anisotropy. Numerical test conclude this section, the main results of which can be found in the article [55].

4.4.1 The heuristic: Transformation of the coordinate system

For anisotropic systems with anisotropy occurring not along the coordinate axes, all multigrid methods based on semicoarsening are constructed with the following heuristic. It is stated both in terms of generating functions and in terms of matrices. The two descriptions correspond to each other.

- **Generating functions:** On the finest level, define a new coordinate system (s, t) such that anisotropy occurs along one of the new axes. Then define a multigrid method in terms of generating functions in the new coordinates similar to the ones from Section 4.2.
- **Matrices:** Initially, permute the rows and columns of $A_{\mathbf{n}}[f]$ and partition the matrix into new blocks. Then, construct a multigrid method as a combination of semicoarsening and full coarsening steps.

We start with the case where anisotropy occurs in an angle of 45° to the coordinate axes, i.e. along the line $y = x$ or $y = -x$. With the 45° case, we can illustrate all ideas which are important for the design of a multigrid method for angles of the form $\frac{k}{k+l} \cdot 90^\circ$ with integers k and l . The new coordinate system (s, t) must be chosen such that anisotropy occurs in s - or t -direction. For anisotropy along the line $y = x$, this is done by defining

$$s := x - y \quad \text{and} \quad t := x + y \quad . \quad (4.4.1)$$

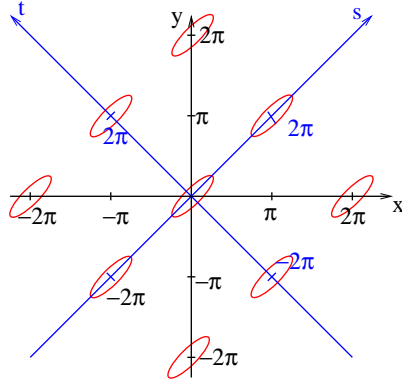


Figure 4.6: The change of coordinates and level curves of f

Then, the function f from Example 5 becomes

$$f(s, t) = (1 - \cos(s)) + \alpha \cdot (1 - \cos(t)) . \quad (4.4.2)$$

The right picture of Figure 4.6 illustrates the consequences of this transformation. For $(s, t) \in [-\pi, \pi]^2$, the function f has only one zero, and anisotropy occurs along the t -axis. Hence, we will construct a multigrid method similar to the ones from Section 4.2.

Since the method should be stated in terms of matrices, the change to new coordinates is translated to the corresponding classes of structured matrices. Defining new coordinates corresponds to permuting rows and columns of $A_{\mathbf{n}}[f]$ and partitioning the resulting matrix into blocks. However, since there are differences between the matrix classes, let us start with the Toeplitz case and describe the circulant case later. If anisotropy occurs in an angle of 45 degrees, permutation and partitioning are done as shown in the left picture of Figure 4.7. Each block of the matrix corresponds to one diagonal in the picture. Under the assumption that $n_1 = n_2$, this means permutation must be done with the permutation vector

$$(1, 2, n_1 + 1, 3, n_1 + 2, 2n_1 + 1, \dots, n_1, n_1 + n_1 - 1, 2n_1 + n_1 - 2, \dots, (n_1 - 1)n_1 + 1, \dots, (n_1 - 1)n_1, n_1^2 - 1, n_1^2) , \quad (4.4.3)$$

and the blocks are of size $1, 2, 3, \dots, n_1 - 1, n_1, n_1 - 1, \dots, 2, 1$. The resulting matrix is denoted \tilde{A} . The only disadvantage of this transformation is that the blocks of \tilde{A} are not all of the same size. Hence, it will not always be possible to retain the structure of the matrix \tilde{A} on coarser levels. Nonetheless, the correspondence of structured matrices and generating functions serves as a good heuristic for anisotropies in other directions.

The situation is significantly better for two-level circulant matrices. The block structure is obtained from the one in Figure 4.7 by uniting two blocks each to form

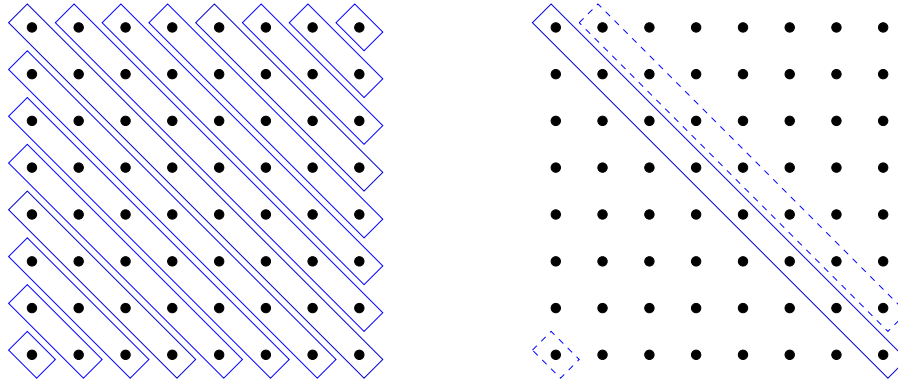


Figure 4.7: Partitioning of the original matrix into blocks in the 45° case: BTTB matrices (left) and BCCB matrices (right)

a single one: blocks 1 and $n_1 + 1$, blocks 2 and $n_1 + 2$, and so on. The n_1 -th block remains the same. This yields the permutation vector

$$(1, 2n_1, 3n_1 - 1, \dots, n_1^2 - (n_1 - 2), 2, n_1 + 1, 3n_1, 4n_1 - 1, \dots, n_1^2 - (n_1 - 3), \dots, n_1, 2n_1 - 1, \dots, n_1^2 - (n_1 - 1)) . \quad (4.4.4)$$

This permutation procedure is illustrated in the right picture of Figure 4.7. The first block is depicted with the dashed line, whereas the n_1 -th block is surrounded by the solid line. The main advantage of this permutation is that all n_1 blocks are of equal size. Therefore, the correspondence between BCCB matrices and generating functions holds as in the case of anisotropy along coordinate axes. This will allow us to develop more efficient multigrid methods than for the other matrix classes.

4.4.2 The 45° case: Two-grid and multigrid methods

With the basic heuristic from the previous subsection in mind, we start with the development of two-grid and multigrid methods for the 45° case. The smoother is chosen to be an elementary pointwise one such as the damped Jacobi method. The main ingredients for the coarsening process are semicoarsening and full coarsening steps, which will be described in the following. Having performed the coordinate transformation, we state the same multigrid method in terms of generating functions as was done in Section 4.2 for anisotropy along coordinate axes. Then, the goal will be to carry over as many results from Section 4.2 as possible for BTTB and BCCB matrices. For f from Example 5, semicoarsening must be performed in s -direction, for example with the function

$$b(s, t) = 1 + \cos(s) . \quad (4.4.5)$$

If full coarsening should be applied, the simplest choice for b is

$$b(s, t) = (1 + \cos(s)) \cdot (1 + \cos(t)) \quad . \quad (4.4.6)$$

If f has a zero of higher order, conditions such as (3.3.24) or (3.3.35) have to be satisfied with x and y being replaced by s and t . For all choices of b , \hat{b} is computed with

$$\hat{f}(s, t) = f(s, t) \cdot b(s, t)^2 \quad . \quad (4.4.7)$$

Elementary projection within a semicoarsening step is done with

$$f_2(s, t) = \frac{1}{2} \cdot [\hat{f}(\frac{s}{2}, t) + \hat{f}(\frac{s}{2} + \pi, t)] \quad , \quad (4.4.8)$$

whereas within a full coarsening step with

$$f_2(s, t) = \frac{1}{4} \cdot [\hat{f}(\frac{s}{2}, \frac{t}{2}) + \hat{f}(\frac{s}{2} + \pi, \frac{t}{2}) + \hat{f}(\frac{s}{2}, \frac{t}{2} + \pi) + \hat{f}(\frac{s}{2} + \pi, \frac{t}{2} + \pi)] \quad (4.4.9)$$

in the circulant case and with

$$f_2(s, t) = \frac{1}{4} \cdot [\hat{f}(\frac{s}{2}, \frac{t}{2}) + \hat{f}(\pi - \frac{s}{2}, \frac{t}{2}) + \hat{f}(\frac{s}{2}, \pi - \frac{t}{2}) + \hat{f}(\pi - \frac{s}{2}, \pi - \frac{t}{2})] \quad (4.4.10)$$

in the Toeplitz or tau case. A semicoarsening step in s -direction reduces the degree of anisotropy in the same way as one semicoarsening step in y -direction did in Section 4.2. This is because after defining the new coordinates we carry out the same calculations in s and t as we did in x and y . The result is summarized in the following theorem. The positive numbers s_F and t_F are the points where s - and t -axis are intersected by the level curves $f(s, t) = c$ for a small positive c .

Theorem 16

Let f be a nonnegative generating function with a zero of order 2 at the origin which is of the form

$$f(s, t) = [\lambda_1(1 - \cos(s)) + \lambda_2(1 - \cos(t))] \cdot h(s, t) \quad (4.4.11)$$

with the trigonometric polynomial $h(s, t) > 0$ and $\lambda_1, \lambda_2 > 0$. Let f_2 be the coarse grid function obtained by one semicoarsening step with b from (4.4.5). Let $r_F = \frac{s_F}{t_F}$ and $r_C = \frac{s_C}{t_C}$ be the ratios described above for f and f_2 , respectively. Then, the degree of anisotropy is reduced by a factor 2, i.e. $\frac{r_F}{r_C} \rightarrow 2$ for $c \rightarrow 0$. Since the coarse level function is of the form (4.4.11) if only terms of order at most 2 are considered in the Taylor expansion, the two-level result can be applied recursively. Then, the degree of anisotropy is reduced by a factor 2 on each level.

Now the multigrid method developed in terms of generating functions is translated to matrices. To simplify notation, we choose $n_1 = n_2$. We start with a two-grid method which consists of one semicoarsening step in s -direction. This step can be divided into three parts.

1. As described above, defining new coordinates corresponds to permuting rows and columns of $A_{\mathbf{n}}[f]$ and partitioning the matrix into new blocks. The resulting matrix is denoted \tilde{A} .
2. The matrix B_S is chosen corresponding to $b(s, t)$ from (4.4.5). In the Toeplitz case, this matrix has blocks of different size. With the ordering obtained after applying (4.4.3), B_S is the block diagonal matrix

$$B_S = \text{diag}(B_1, B_2, \dots, B_{n_1}, \dots, B_2, B_1) \quad , \quad (4.4.12)$$

where $B_1 = 1$ and all other blocks are $B_k = \text{tridiag}_k(0.5, 1, 0.5)$ of size k . In the circulant case, all blocks obtained from permutation with (4.2.34) are of equal size, i.e. we get a block diagonal matrix whose diagonals are 1D circulant matrices corresponding to $1 - \cos(x)$.

3. The coarse grid matrix A_C is again computed by applying elementary restriction to the matrix $\hat{A} = B_S \cdot \tilde{A} \cdot B_S$. This means we leave the number of blocks unchanged, and within each block, we pick every second row and every second column.

Since we wish to develop a multigrid method as a combination of semicoarsening steps in s and full coarsening steps, we must first describe how a single full coarsening step in s and t is carried out. A two-level method with full coarsening is defined by translating (4.4.6), (4.4.9), and (4.4.10) into matrices. Permutation of $A_{\mathbf{n}}[f]$ and partitioning of \tilde{A} are done in the same way as for a semicoarsening step. The matrix

$$B_F = B_S + B_T \quad (4.4.13)$$

is chosen to be the sum of B_S from (4.4.12) and B_T , which is defined as

$$B_T = \begin{pmatrix} 0 & B_{1,2} & & & & & \\ B_{2,1} & 0 & B_{2,3} & & & & \\ & B_{3,2} & 0 & B_{3,4} & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & B_{2,3} & 0 & B_{2,1} & \\ & & & & B_{1,2} & 0 & \end{pmatrix} . \quad (4.4.14)$$

In the Toeplitz case, $B_{k,k+1}$ is the matrix

$$B_{k,k+1} = \text{tridiag}(0.25, 0.5, 0.25)$$

of size k -by- $(k+1)$, and $B_{k+1,k}$ the same matrix of size $(k+1)$ -by- k . In the circulant case, all blocks are of size n_2 . After $\hat{A} = B_F \cdot \tilde{A} \cdot B_F$ is computed, A_C is obtained by elementary restriction. Within each block, we pick every second row and every second column. On the block level, we pick two rows, eliminate the next two, pick another two rows and so on. This procedure is shown in Figure 4.8.

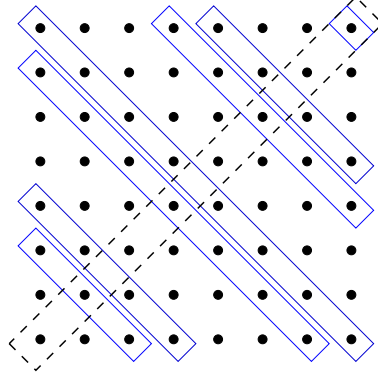


Figure 4.8: A full coarsening step in the 45° case

The solid lines mark the blocks which are retained on the coarser grid, whereas the other blocks are eliminated. The dashed line explains why precisely these blocks have to be chosen. Elimination on the block level must be done such that within the dashed line, every second element is retained and the other elements are eliminated. In the circulant case, the same technique is used, just with blocks of equal size.

We now define a multigrid method similar to the one from Section 4.2, i.e. as a suitable combination of semicoarsening steps followed by some full coarsening steps. The prolongation/restriction matrices and the elementary projection matrices are defined as described above, the change of coordinates has of course only to be done before the first step. Again, we use the same heuristic as in Section 4.2. Theorem 16 states that the ratio r_F is reduced by a factor 2 in each semicoarsening step. Therefore, semicoarsening steps are applied until level curves are close to circles, i.e. until r_F is almost 1. Then, we continue with full coarsening.

4.4.3 Generalization to other directions

The case where anisotropy occurs in an angle of 45° with respect to the coordinate axes is best suited to explain our method. Although systems with this angle arise in many applications, this is not the only important case. Therefore, we want to describe how to solve systems where anisotropy occurs in other directions. The function $g(x, y)$ from Example 5 is anisotropic in an angle of 30° with respect to the y -axis. Furthermore, it has the following zeros in the interval $[-\pi, \pi]^2$:

$$(0, 0), \left(\frac{2}{5}\pi, -\frac{4}{5}\pi\right), \left(\frac{4}{5}\pi, \frac{2}{5}\pi\right), \left(-\frac{4}{5}\pi, -\frac{2}{5}\pi\right), \left(-\frac{2}{5}\pi, \frac{4}{5}\pi\right) . \quad (4.4.15)$$

Again, we wish to define a multilevel method as a combination of semicoarsening and full coarsening steps. As in the 45° case, we define new coordinates s and t such that anisotropy occurs along coordinate axes, and then apply coarsening along

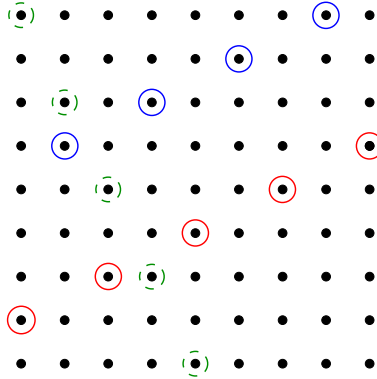


Figure 4.9: Partitioning of the original matrix corresponding to g into blocks

s and t . For a problem with an angle of 30° towards the y -axis, such as $g(x, y)$ from Example 5, the new coordinates are $s := 2x + y$ and $t := -x + 2y$. Having defined the new coordinates, we choose $b(s, t)$ either as in (4.4.5) or in (4.4.6), and proceed as in the 45° case. The coordinate transformation is translated into matrices by permuting rows and columns, and then partitioning the matrix into blocks. The grid points in Figure 4.9 which are highlighted by a solid circle show how two example blocks of the matrix are built. Since we have anisotropy in an angle of 30° , the points corresponding to one block are obtained by moving two steps in x -direction and one step in y -direction. The matrices B_S and B_T are defined as in (4.4.12) and (4.4.14), just the size of the blocks is different. Elementary restriction within a semicoarsening step is done exactly as in the 45° case, by eliminating every second row and column within each block, leaving the number of blocks the same. If we apply full coarsening, we eliminate every second row and column within each block, and on the block level, we eliminate five consecutive block rows, then pick the next five block rows, eliminate five block rows and so on. The reason for choosing this pattern is explained in Figure 4.9, where we have to eliminate every second grid point with a dashed circle. This is the equivalent to the dashed line in Figure 4.8. There are three other directions where anisotropy occurs in an angle of 30° to one of the coordinate axes. Each of them is treated as the one we have described here by an appropriate choice of s and t .

Finally, let us take a look at anisotropy which occurs yet in other directions. Functions of the form

$$f(x, y) = \alpha \cdot (1 - \cos(k \cdot x + l \cdot y)) + (1 - \cos(l \cdot x - k \cdot y)) \quad (4.4.16)$$

are examples, representing classes of problems where anisotropy occurs in an angle of $\frac{k}{k+l} \cdot 90^\circ$ to one of the coordinate axes. In this general case, transformation to new coordinates is done with

$$s := k \cdot x + l \cdot y \quad \text{and} \quad t := l \cdot x - k \cdot y \quad . \quad (4.4.17)$$

k	l	block size	angle
1	2	5	30°
1	3	10	22.5°
1	4	17	18°
1	5	26	15°
2	3	13	36°

Table 4.8: Angles and block size for different choices of k and l in (4.4.16)

If $|k|$ and $|l|$ are small, our method works very well for these systems. However, this approach is limited to small $|k|$ and $|l|$, because the block sizes which have to be used in the multigrid method become large if $|k|$ and $|l|$ increase. For $|k| = |l| = 1$, the size was 2, for $|k| = 1, |l| = 2$, it was 5, and in general it is $\det \begin{pmatrix} l & -k \\ k & l \end{pmatrix}$, i.e. $k^2 + l^2$. Table 4.8 illustrates what blocksize we have to use and what angle we get for different choices of k and l . For each line in Table 4.8, four problems with different angles can be obtained by interchanging k and l , and by moving α to the term $(1 - \cos(l \cdot x - k \cdot y))$ in (4.4.16). Most other angles lead to a block size which becomes too large for practical computations. In theory however, if the size of $A_{\mathbf{n}}[f]$ is large enough, any rational angle can be described by k and l , allowing our method to be applied. Thus, we suggest to approximate the angle of anisotropy and to treat a given problem as if anisotropy occurred along a direction from (4.4.16).

4.4.4 Convergence results

In the following, we wish to give a convergence result for the Toeplitz case. It is a generalization of Theorem 13 for anisotropies in other directions. Let us assume that anisotropy of a matrix $T_{\mathbf{n}}[f]$ occurs in a direction where the new coordinates for our method can be defined with (4.4.17). We prove the following theorem for the case where f is anisotropic along the t -axis. It is required that, in the new coordinates, f satisfies the condition

$$\min_{(s,t) \in [-\pi, \pi]^2} \frac{f(s,t)}{1 - \cos s} = C > 0 . \quad (4.4.18)$$

The coordinate transformation with (4.4.17) corresponds to a permutation of the rows and columns of $T_{\mathbf{n}}[f]$ with a vector $perm$ such as the one in (4.4.3), i.e. $\tilde{T} = T_{\mathbf{n}}[f](perm, perm)$. The inverse permutation is defined by the vector $iperms$. The restriction matrix for a semicoarsening step in s -direction is defined as the product of B_S from (4.4.12) and an elementary restriction matrix. We can now prove this more general version of the two-level result from Theorem 13.

Theorem 17

Let $T_{\mathbf{n}}[f]$ be a positive definite BTTB matrix whose generating function $f(x, y)$ is real-valued even. Assume that introduction of new coordinates s and t by (4.4.17) leads to a generating function $f(s, t)$ which satisfies (4.4.18). Let \tilde{T} be the permuted matrix defined above. Furthermore, let the restriction matrix \tilde{R} be constructed with B from (4.4.12), and let the smoother be the damped Jacobi method. Then, the correcting condition (3.1.18) is satisfied for \tilde{T} and \tilde{R} , and the convergence factor of the two-level method is uniformly bounded from below 1 independently of n .

Proof: The proof is similar to the one of Theorem 13, but this time we have to consider blocks of variable size. First, assume that all diagonal blocks are of odd size m_j , i.e. $m_j = 2k_j + 1$ for some integer k_j . For any

$$e = (e_{1,1}, \dots, e_{1,m_1}, e_{2,1}, \dots, e_{2,m_2}, \dots, e_{b,1}, \dots, e_{b,m_b})^T$$

we define

$$e_C = (\tilde{e}_{1,1}, \dots, \tilde{e}_{1,k_1}, \tilde{e}_{2,1}, \dots, \tilde{e}_{2,k_2}, \dots, \tilde{e}_{b,1}, \dots, \tilde{e}_{b,k_b})^T,$$

where $\tilde{e}_{i,j} = e_{i,2j}$. If $j \leq 0$ or $j > m_k$, then we set $e_{i,j} = 0$ in order to complete the notation. For this special choice of e_C we try to find an upper bound for $\|e - \tilde{R}^T e_C\|_0^2$ of the form $\beta \|e\|_1$ with β independent of e . Similar to [109] we obtain

$$\begin{aligned} \|e - \tilde{R}^T e_C\|_0^2 &= t_{0,0} \sum_{i=1}^b \sum_{j=0}^{k_i} \left\{ e_{i,2j+1} - \frac{1}{2} e_{i,2j+2} - \frac{1}{2} e_{i,2j} \right\}^2 \\ &\leq t_{0,0} \sum_{i=1}^b \sum_{j=0}^{n_i} (e_{i,j}^2 - e_{i,j} e_{i,j+1}) = t_{0,0} \langle e, \text{diag}(T_{m_1}, \dots, T_{m_b}) \cdot e \rangle \end{aligned} \quad (4.4.19)$$

with $T_{m_j} = T_{m_j}[1 - \cos(s)]$. Again, we have to find a parameter β independent of e such that

$$t_{0,0} \langle e, \text{diag}(T_{m_1}, \dots, T_{m_b}) \cdot e \rangle \leq \beta \langle e, \tilde{T}e \rangle, \quad \forall e \in \mathbb{R}^{\mathbf{n}}. \quad (4.4.20)$$

First, we permute the left-hand side back to x - and y -coordinates with the vector $iperms$, i.e. $e_{iperms} = e(iperms)$ and

$$T_{\mathbf{n}}[1 - \cos(kx + ly)] = \text{diag}(T_{m_1}, \dots, T_{m_b})(iperms, iperms).$$

As in the proof of Theorem 13, the inequality in the following expression is a consequence of (4.4.18):

$$\begin{aligned} t_{0,0} \langle e, \text{diag}(T_{m_1}, \dots, T_{m_b}) \cdot e \rangle &= t_{0,0} \langle e_{iperms}, T_{\mathbf{n}}[1 - \cos(kx + ly)] \cdot e_{iperms} \rangle \\ &\leq t_{0,0} \langle e_{iperms}, T_{\mathbf{n}}[f(x, y)] \cdot e_{iperms} \rangle = t_{0,0} \langle e, \tilde{T}e \rangle. \end{aligned} \quad (4.4.21)$$

α	coarsening	$\mathbf{n}=(2^6-1)^2$	$\mathbf{n}=(2^7-1)^2$	$\mathbf{n}=(2^8-1)^2$
0.002	t,t,st,st	10	12	13
0.002	t,t,t,t	6	6	6
0.0001	t,t,st,st	24	51	77
0.0001	t,t,t,t	6	6	6

Table 4.9: Iteration numbers for $T_{\mathbf{n}}[f]$ with f from Example 5 with $\alpha = 0.002$ and $\alpha = 0.0001$

The last equality is obtained by permutation with the vector $perm$, i.e. by transformation to s - and t -coordinates. From (4.4.21) we obtain the parameter $\beta = \frac{t_0,0}{C}$ in (4.4.20).

Finally, we must get rid of the assumption that the block sizes be odd. Let us therefore assume that the j -th block is of even size m_j . The vector $(e_{j,1}, \dots, e_{j,m_j})$ is embedded into a vector of size $m_j + 1$ by filling 0 into the additional position. If this is done for all parts of e corresponding to a block of even size, we obtain a vector \hat{e} which is slightly larger than e . Then, with (4.2.28) and

$$\langle \hat{e}, \text{diag}(T_{\tilde{m}_1}, \dots, T_{\tilde{m}_b})\hat{e} \rangle = \langle e, \text{diag}(T_{m_1}, \dots, T_{m_b})e \rangle ,$$

the correcting condition still holds. ■

4.4.5 Numerical results

We wish to give some numerical results for a multigrid method which is constructed as a suitable combination of semicoarsening steps followed by some full coarsening steps. The prolongation/restriction matrices and the elementary projection matrices are defined as described above. The change of coordinates, of course, has to be done only before the first step. Again, we use the same heuristic as in Section 4.2. Semicoarsening steps are applied until level curves are close to circles, i.e. until r_F is almost 1. Then we continue with full coarsening. Our theoretical results state that the ratio r_F is reduced by a factor 2 in each semicoarsening step. We wish to test our multilevel method with the function $f(x, y)$ from Example 5, where α takes the values 0.002. The corresponding matrices $T_{\mathbf{n}}[f]$ belong both to the two-level Toeplitz class and to the two-level tau algebra. We use a five-grid method, where one step of symmetric Gauss-Seidel is used as pre- and postsmoother. Our theory suggests to use four semicoarsening steps, because $r_F = 22.36$. Table 4.9 shows that the number of V-cycle iterations is significantly lower if four semicoarsening steps are used instead of only two. For a strongly anisotropic problem such as the same function f with $\alpha = 0.0001$ and $r_F = 100$, two semicoarsening steps followed by full coarsening do not lead to satisfactory convergence at all. If only semicoarsening is used, we observe rapid convergence.

α	coarsening	$\mathbf{n}=(2^6-1)^2$	$\mathbf{n}=(2^7-1)^2$	$\mathbf{n}=(2^8-1)^2$
0.02	t,st,st,st,st	23	24	24
0.02	t,t,t,st,st	5	5	5
0.02	t,t,t,t,t	8	7	7
0.001	t,st,st,st,st	> 200	> 200	> 200
0.001	t,t,t,st,st	27	26	26
0.001	t,t,t,t,t	5	5	5

Table 4.10: Iteration numbers for $\tilde{C}_{\mathbf{n}}[f]$ with f from Example 5 with $\alpha = 0.002$ and $\alpha = 0.0001$

Let us finally consider a two-level circulant example. The matrix $\tilde{C}_{\mathbf{n}}[f]$ with f from Example 5 is obtained from $C_{\mathbf{n}}[f]$ by adding $\frac{1}{(n_1 \cdot n_2)^2}$. As we have seen in Section 4.4.1, these BCCB examples can be treated almost like the BCCB systems where anisotropy occurs along coordinate axes. The semicoarsening steps are exactly the same as in Chapter 4.2. For full coarsening, we have to use the block interpretation of [71], which is equally valid for BCCB matrices. Table 4.10 shows the number of V-cycle iterations for different numbers of semicoarsening steps. As we expect from our theory, best results for $\alpha = 0.02$ are obtained with three semicoarsening steps and for $\alpha = 0.001$, with five semicoarsening steps.

4.5 Anisotropy in other directions: Line smoothing

For problems with anisotropy along coordinate axes we have used standard coarsening in combination with a line smoother to define a different kind of multigrid algorithm. The same can be done with the systems introduced in Example 5. We must take into account that the functions $f(x, y)$ and $g(x, y)$ have multiple zeros in $[0, 2\pi]^2$. In the following, we describe how line smoothing is applied to this type of anisotropy, state a result concerning the smoothing conditions of the Ruge-Stüben-theory, and finally present numerical results.

4.5.1 Line smoothing and block coarsening

A multigrid method based on full coarsening and line smoothing requires the transformation of coordinates, and therefore the permutation of rows and columns of the corresponding matrices, only for the smoothing operations. For prolongation and restriction as well as for the computation of the system matrices on coarser grids, we use the original coordinates x and y . Since we wish to apply a line smoother, the rows and columns of $A_{\mathbf{n}}[f]$ are permuted with the vector from (4.4.3) if $A_{\mathbf{n}}[f]$ is Toeplitz. Circulant matrices are permuted with the vector (4.4.4). Then, we partition the permuted matrix into blocks in the same way as it was done for the

semicoarsening method. Each line in Figure 4.8 corresponds to one block of the matrix.

Computation of the coarse grid matrix A_C cannot be done in such a straightforward way as in Section 4.3, because f has an additional zero at (π, π) . Therefore, we make use of the results from Sections 3.5 and 3.6, where one- and two-dimensional Toeplitz systems were solved with a block multigrid method whose generating functions have multiple zeros. This approach can also be applied to the tau and circulant algebras. The BTTB matrix $T_{\mathbf{n}}[f]$ of size n^2 -by- n^2 is considered to be a block BTTB matrix with blocks of size 4. Thus, the generating function becomes a 4-by-4 matrix $F(x, y)$, whose entries are functions in x and y . The eigenvalues of $F(x, y)$ in Example 5 only become zero at $(0, 0)$. Then, $B(x, y)$ is for example chosen to be the 4-by-4 diagonal matrix with $b(x, y) = (1 + \cos(x))(1 + \cos(y))$ in each position of the diagonal, taking care of the zero in $F(x, y)$. The coarse grid matrix is computed by picking every second 2-by-2 block on both levels.

4.5.2 Theoretical results

The following theorem is a generalization of Theorem 15 which covers anisotropies occurring in arbitrary rational angles.

Theorem 18

Let $f(x, y)$ be a nonnegative generating function with a zero of order 2 at the origin which, in the rotated coordinates, is of the form

$$\tilde{f}(s, t) = [(1 - \cos(s)) + \alpha(1 - \cos(t))] \cdot h(s, t) \quad (4.5.1)$$

with the trigonometric polynomial $h(s, t) > 0$ and $0 < \alpha \ll 1$. Let $A = A_{\mathbf{n}}[f]$ be the two-level circulant matrix corresponding to $f(x, y)$, and \tilde{A} its permuted version corresponding to $\tilde{f}(s, t)$. Let D be the matrix corresponding to $g(x, y)$, which, in the rotated coordinates is of the form

$$\tilde{g}(s, t) = (1 + \alpha - \cos(s)) \cdot \tilde{h}(s) \quad , \quad (4.5.2)$$

where \tilde{h} is obtained by eliminating all terms in h containing t . \tilde{D} , the permuted version of D , is a block diagonal matrix with the same diagonal blocks as \tilde{A} . If $\tilde{h}(s) > 0$, then the block Jacobi method satisfies the smoothing conditions (3.1.16) and (3.1.17).

Proof: After replacing the functions $f(x, y)$ and $g(x, y)$ by $\tilde{f}(s, t)$ and $\tilde{g}(s, t)$ the calculations are the same as in the proof of Theorem 15. As in the proof of Theorem 15, the matrix Y is chosen to be D^{-1} . ■

Remark 15 Again, the proof does not include BTTB matrices for the technical problems mentioned in Remark 13. However, the numerical results of the multigrid solution of anisotropic BTTB systems with the block Jacobi smoother are quite promising (see Section 4.5.3).

α	$\mathbf{n}=(2 \cdot (2^5-1))^2$	$\mathbf{n}=(2 \cdot (2^6-1))^2$	$\mathbf{n}=(2 \cdot (2^7-1))^2$
0.1	9	8	8
0.001	3	6	11
0.00001	2	2	2

Table 4.11: Iteration numbers for the block Jacobi smoother and $T_{\mathbf{n}}[f]$ with f from Example 5

4.5.3 Numerical results

Finally, we wish to present numerical results for multigrid methods using line smoothers for relaxation and the block strategy mentioned above for full coarsening. Again, this is too expensive if the diagonal blocks are full. However, if the matrix is sparse, this type of method is a good alternative to the multigrid algorithm with semicoarsening. As we have observed for the systems in Section 4.3, this method obtains its best results if the anisotropy is either moderate or very strong. If α is somewhere between 0.05 and 0.005, the method based on semicoarsening is preferable. The following iteration numbers we obtained with a three-level method, where two steps of block Jacobi were used as pre- and postsmoother.

For circulant matrices $\tilde{C}_{\mathbf{n}}[f]$ we can also define a multigrid method using standard prolongation and a line smoother. The convergence behavior is the same as for the circulant example with anisotropy along coordinate axes, where the results were shown in Table 4.6. For circulant matrices, however, inversion of the diagonal blocks is too expensive in most cases, since the whole system can be solved in $O(\mathbf{n} \log \mathbf{n})$ with the FFT. Therefore, we suggest to use the multilevel method which is based on semicoarsening for BCCB matrices.

Chapter 5

Generating functions with whole zero curves

All structured linear systems for which multigrid methods have been developed so far correspond to nonnegative generating functions with isolated zeros. Typical applications for these types of matrices in the field of PDEs are the solution of the discrete Laplace or Poisson equation. Also anisotropic versions of these equations fall into this category. In the following, however, we are interested in matrices corresponding to nonnegative functions having a whole zero curve instead of isolated zeros. Such linear systems arise for example when the discrete Helmholtz equation is solved with normal equations. Before working on applications we wish to develop multigrid methods for these systems in a more theoretical setting, again using generating functions for the definition of restriction and coarse grid matrices. Let us start with an example, which will be used to illustrate the main features of our multigrid methods and which will be the basis for application of our method to the discrete Helmholtz equation.

Example 6 Let $A_n[f]$ be the two-level matrices belonging to a trigonometric algebra or to the Toeplitz class which correspond to the generating function

$$f(x, y) = (\rho - \cos(x) - \cos(y))^2 \quad (0 < \rho \leq 2). \quad (5.0.1)$$

If $\rho = 2$, $f(x, y)$ has a single isolated zero at the origin of order 4, and the corresponding linear systems can be solved with the method from Chapter 2. For $\rho < 2$, f is zero along a whole curve. These zero curves become larger as ρ decreases.

For this type of matrices, the classical convergence theory for two-level structured matrices [24, 99, 3], which was presented in Chapter 3, does not hold anymore, and standard multigrid methods fail. Therefore, we seek to devise multilevel methods which are especially designed for application to structured linear systems whose generating functions have zero curves.

Starting from a Galerkin method, we present a multigrid algorithm which is based on rediscrretization on coarser levels and on approximation of the zero curve. Since such a method preserves the bandedness of a given matrix, it is more suitable for practical use than a pure Galerkin method. Then, we introduce a splitting technique which divides the original problem into several subproblems on coarser grids which are easier to solve. This technique is then combined with the Galerkin approach. Furthermore, we wish to solve anisotropic linear systems with whole zero curves. We propose a combination of the methods from this chapter and the ones from Chapter 4. Since multigrid methods can also be used as preconditioners for Krylov subspace methods, we develop a slightly different splitting technique for the design of multigrid preconditioners.

5.1 Galerkin-based multigrid

In this section, we present a Galerkin-based multigrid method for linear systems corresponding to functions with whole zero curves. The first strategy for the design of a multigrid solver is to extend the Galerkin methods from Chapter 3 and make them suitable for functions with zero curves instead of isolated zeros. This has been suggested at the end of the article [72]. After giving a convergence proof for the two-grid method, we explain the practical problems of extending the two-grid method to a multigrid method.

5.1.1 Extending standard multigrid

In Chapter 3, we have described the standard Galerkin-based multigrid method for systems with an isolated zero (x_0, y_0) , which was introduced by Fiorentino and Serra [54]. Let us recall that the prolongation and restriction matrices are defined in terms of generating functions such that the functions of the coarse grid matrices have a single zero at $(2x_0, 2y_0)$, $(4x_0, 4y_0)$, $(8x_0, 8y_0)$, etc. Matrices of the tau algebras have two or four zeros on each level, unless the zero is located at the origin, because they correspond to even functions. The prolongation function $b(x, y)$ is zero at the mirror points defined in (3.3.20) or (3.3.21) and strictly positive at (x_0, y_0) . As described in Chapter 2, the elementary projection matrix is chosen such that the coarse grid matrix A_2 belongs to the same class or algebra as $A_n[f]$.

For the solution of linear systems whose generating functions have zero curves, we wish to design a method which is based on the same idea. We use the same standard smoother, i.e. the damped Jacobi or Gauss-Seidel method, and the same elementary restriction matrices E as in Chapter 3. Prolongation and restriction must be defined such that every zero (x, y) on the curve is mapped to $(2^k x, 2^k y)$ on coarser grids. The zero curve $f(x, y) = 0$ on the finest level must become $f(x/2, y/2) = 0$, $f(x/4, y/4) = 0$, $f(x/8, y/8) = 0$, etc. on the next coarser grids. In other words, we impose conditions similar to (3.3.22) and (3.3.23) on the

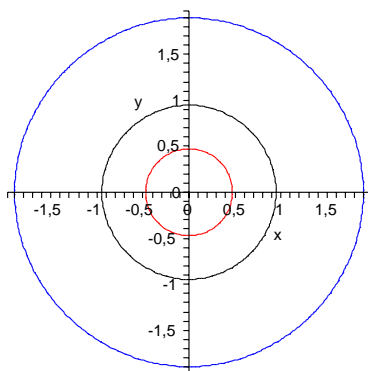


Figure 5.1: Zero curves of f, f_2, f_3 for the function $f(x, y) = (1.9 - \cos(x) - \cos(y))^2$ on the three finest levels

function b corresponding to the matrix B . This time there is not a single zero, but a whole curve of zeros for which (3.3.22) and (3.3.23) must be satisfied. This means that instead of three mirror points, b must be zero at three *mirror curves*, which are obtained by considering the three mirror points for every point on the curve $f(x, y) = 0$. The function $b(x, y)$, which is chosen

$$b(x, y) = f(x + \pi, y) \cdot f(x, y + \pi) \cdot f(x + \pi, y + \pi) \quad (5.1.1)$$

in the circulant case and

$$b(x, y) = f(\pi - x, y) \cdot f(x, \pi - y) \cdot f(\pi - x, \pi - y) \quad (5.1.2)$$

in the tau, DCT-III, DST-III, and Toeplitz case, meets these requirements. After having made this choice of b , the coarse grid function f_2 is computed with (3.3.2) and (3.3.19). The following properties of f_2 are easily verified by direct calculation.

Lemma 1 *Let $A_{\mathbf{n}}[f]$ be a tau or circulant matrix whose generating function has a curve of zeros in $]-\pi/2, \pi/2[^2$. Let b be chosen as in (5.1.1) or (5.1.2), and let f_2 be computed with (3.3.2) and (3.3.19). Then $f_2(x, y)$ has the same zero curve as $f(x/2, y/2)$, and the zeros are of the same order as the ones of $f(x, y)$. Moreover, conditions (3.3.22) and (3.3.23) hold for each point (x_0, y_0) on the zero curve of f .*

For the construction of a Galerkin-based multigrid method we chose

$$b_2(x, y) = f_2(x + \pi, y) \cdot f_2(x, y + \pi) \cdot f_2(x + \pi, y + \pi) \quad (5.1.3)$$

ρ	$\mathbf{n}=(2^5-1)^2$	$\mathbf{n}=(2^6-1)^2$	$\mathbf{n}=(2^7-1)^2$	$\mathbf{n}=(2^8-1)^2$
1.9	18	18	18	18
1.8	18	18	18	18
1.6	18	18	18	18

Table 5.1: Iteration numbers for the Galerkin-based two-grid method applied to $\tau_{\mathbf{n}}[f]$ with f from Example 6

or its tau equivalent to compute the restriction matrix on the next level. This leads to a function $f_3(x, y)$, which has the same zero curve as $f(x/4, y/4)$. Figure 5.1 depicts the zero curves of f , f_2 , and f_3 for the function f from Example 6 with $\rho = 1.9$. To illustrate that our method converges fast we apply it to the matrices from Example 6. For $f(x, y)$ from (5.0.1), the function b becomes

$$b(x, y) = (\rho - \cos(x) + \cos(y))^2 (\rho + \cos(x) - \cos(y))^2 (\rho + \cos(x) + \cos(y))^2 \quad (5.1.4)$$

for all trigonometric matrix algebras and for Toeplitz matrices. The following table contains iteration numbers of the two-grid method for the two-level tau matrices corresponding to $f(x, y)$. As in the isolated zero case, our method can also be applied to matrices belonging to the two-level Toeplitz class. However, we either have to apply additional cutting to enforce BTTB structure of A_2 , or accept that A_2 is BTTB with low-rank perturbations. Nevertheless, we obtain similar numerical results if we apply our two-grid method to the BTTB matrices corresponding to $f(x, y)$ from Example 6.

5.1.2 Two-grid convergence results

For the two-level method, we wish to give a theoretical convergence result which is based on Theorem 6 and Corollary 1. We extend two-grid convergence proofs for two-level tau [99], DCT-III [28], DST-III, and circulant matrices [103], which were stated by Serra et al. for generating functions with isolated zeros. Since the proofs for the different algebras are similar, the following theorem is formally proved for two-level τ -matrices. At the end, we explain the differences in the proof for the other two algebras.

Theorem 19

Let $A := A_{\mathbf{n}}[f]$ be a two-level matrix from the circulant, tau, DCT-III, or DST-III algebra. Assume that $f(x, y)$ is a cosine nonnegative polynomial (not identically zero) with a zero curve in $]-\frac{\pi}{2}, \frac{\pi}{2}[^2$. Suppose that the smoother is the damped Richardson or Jacobi method. Furthermore, let the restriction be $R = B \cdot E$ with B from (5.1.1) or (5.1.2) and with the elementary restriction matrix E of the respective algebra. Then, there exists $\beta > 0$ such that condition (3.1.18) is satisfied, and the TGM converges.

Proof: From Chapter 3 we know that the elementary smoothers such as the Richardson and Jacobi method satisfy both the presmoothing condition (3.1.16) and the postsmoothing condition (3.1.17). Thus it remains to show that the correcting condition (3.1.18) is satisfied. Since the technique of the proof is similar to the one used in [99, 28, 103] and in Section 4.2.3, we abbreviate it here. (3.1.18) is proved by showing that it holds if we choose, for each $\mathbf{x} \in \mathbb{C}^n$, the following \mathbf{y} :

$$\mathbf{y} = [RR^H]^{-1}R\mathbf{x} \quad .$$

In [99], it is shown that this is true if there exists $\gamma > 0$ such that

$$I - R^H[RR^H]^{-1}R \leq \frac{\gamma}{\hat{a}}A_{\mathbf{n}}[f]$$

with $\hat{a} = A_{j,j} > 0$. By performing a block diagonalization procedure with a permuted version of $Q_{\mathbf{n}}^{(tau)}$ from (2.2.6) this is equivalent to proving $n_1n_2 - 4n_1^C n_2^C$ scalar inequalities and $n_1^C n_1^C$ 4-by-4 matrix inequalities, where $n_1^C = \lfloor n_1/2 \rfloor$ and $n_2^C = \lfloor n_2/2 \rfloor$. The scalar inequalities are of the form $\hat{a} \leq \gamma f(x_\mu, y_\nu)$ with either $\mu = n_1^C + 1$ or $\nu = n_2^C + 1$ or both. Due to our assumption that the zero curve is located within $]-\frac{\pi}{2}, \frac{\pi}{2}[^2$, f cannot vanish if x or y takes the value $\pi/2$, and the scalar inequalities hold. Because of the continuity of f and b the matrix inequalities can be reduced to a unique inequality involving 4-by-4 matrix-valued functions [99]. This inequality is of the form $L(x, y) \leq \frac{\gamma}{\hat{a}}I_4$ with

$$L(x, y) = \text{diag}(f[x, y])^{-1/2} \left(I_4 - \frac{1}{\|b[x, y]\|_2^2} b[x, y](b[x, y])^T \right) \text{diag}(f[x, y])^{-1/2} ,$$

where $f[x, y] = (f(\bar{\mathbf{x}}_1), f(\bar{\mathbf{x}}_2), f(\bar{\mathbf{x}}_3), f(\bar{\mathbf{x}}_4))$ with $\bar{\mathbf{x}}_1 = (x, y)$ and its mirror points $\bar{\mathbf{x}}_2, \bar{\mathbf{x}}_3, \bar{\mathbf{x}}_4$. $b[x, y]$ is defined analogously. This inequality holds if $L(x, y)$ is uniformly bounded in spectral norm, which is in turn true if each element $L_{i,j}(x, y)$ of the 4-by-4 matrix function $L(x, y)$ is bounded in L^∞ . For $i \neq j$,

$$L_{i,j}(x, y) = -\frac{b(\bar{\mathbf{x}}_i)b(\bar{\mathbf{x}}_j)}{\sqrt{f(\bar{\mathbf{x}}_i)f(\bar{\mathbf{x}}_j)}} \frac{1}{\|b[x, y]\|_2^2}$$

is bounded, because, due to (3.3.23), one of the four terms in $b[x, y]$ is nonzero and, due to (3.3.22), $\frac{b(\bar{x}_i)}{f(\bar{x}_i)}$ is bounded. For all other $\mathbf{x} \in]-\pi, \pi[^2$ which are not located on the zero curve, f is strictly positive. For $i \in \{1, 2, 3, 4\}$, the functions

$$L_{i,i}(x, y) = -\frac{\sum_{y \in M(\bar{x}_i)} b^2(y)}{f(\bar{x}_i)} \cdot \frac{1}{\|b[x, y]\|_2^2}$$

are also bounded, because the first factor is bounded due to (3.3.22).

For circulant matrices, the proof is almost the same. Since matrices are assumed to be of even size, there are no scalar inequalities. $L(x, y)$ and $b[x, y]$ are

the same as for tau matrices. For DCT-III and DST-III matrices, $b[x, y]$ must be chosen differently. If

$$b[x, y] := \left(\cos\left(\frac{x}{2}\right) \cos\left(\frac{y}{2}\right) b(\bar{\mathbf{x}}_1), -\sin\left(\frac{x}{2}\right) \cos\left(\frac{y}{2}\right) b(\bar{\mathbf{x}}_2), \right. \\ \left. -\cos\left(\frac{x}{2}\right) \sin\left(\frac{y}{2}\right) b(\bar{\mathbf{x}}_3), \sin\left(\frac{x}{2}\right) \sin\left(\frac{y}{2}\right) b(\bar{\mathbf{x}}_4) \right) .$$

then $L(x, y)$ is proved to be bounded in infinity norm. This is done by showing that all $L_{i,j}$ are bounded due to (3.3.34) and (3.3.35). ■

5.1.3 Limitations of the Galerkin approach

The Galerkin-based two-grid method converges after a low number of iterations, independent of the matrix size, and we have proved convergence in Theorem 19. However, the design of a multigrid method for practical applications runs into two major problems.

- Most of the matrices we are interested in are sparse, i.e. their corresponding generating functions are trigonometric polynomials of low degree. Application of the Galerkin method with $b(x, y)$ from (5.1.1) or (5.1.2) results in matrices which are significantly denser on coarser levels. To illustrate this fact, let us examine the matrix $A_1 = A_{\mathbf{n}}[f]$ and the corresponding coarse grid matrix A_2 from Example 6. The stencil of A_1 has 13 nonzero entries, compared to 113 of A_2 . This enormous increase in density occurs in each coarsening step, and therefore makes a multigrid method with more than three levels inefficient. In some cases, it is possible to define a function $b(x, y)$ which is less dense, but which has the same zero curves as b from (5.1.1) or (5.1.2). For our example function f , we can use $\sqrt{b(x, y)}$, resulting in a stencil of A_2 which has only 41 nonzero entries. However, such a prolongation is only possible if this square root is itself a trigonometric polynomial of low degree.
- Zero curves become larger on each level. Figure 5.1 shows that two steps of coarsening transform a zero curve of moderate size into a considerably larger one. However, our multigrid method only works well if the zero curve is located within the region $]-\pi/2, \pi/2[^2$. Even if the curve only approaches the boundaries of this area, i.e. if for a zero (x_0, y_0) on the curve, either x_0 or y_0 becomes greater than 1.3 or 1.4, convergence is extremely slow. For the example function $f(x, y)$ from Example 6, we see in Figure 5.1 that at most three levels, i.e. two coarsening steps can be used. To illustrate this restriction, the following table shows, for the desired number m of grids in the multigrid method, how small ρ is allowed to be in (5.0.1) such that an m -grid method can still be applied.

# levels	2	3	4	5	6	7
ρ_{min}	1.25	1.80	1.95	1.987	1.997	1.9995

5.2 Rediscretization and approximation of the zero curve

The Galerkin method described in the previous section leads to optimal convergence for matrices whose generating functions have a whole curve of zeros. Although the two problems described at the end of the section prevent the multigrid version of the method from being applicable in practical computations, it is the basis for the development of fast multigrid methods. In the following, we focus on the problem of finding coarse grid matrices which are less dense, but which do not increase the number of V-cycle iterations. This shall be achieved by using the following heuristic: *Carry out smoothing, prolongation, and restriction as before, but do not compute the coarse grid matrix A_2 with a Galerkin approach.* Instead, we choose A_2 corresponding to a generating function which has the same zero curve as $f(x/2, y/2)$, or which is at least a good approximation to this curve. In other words, we use a form of rediscrization on coarser grids. Finding a sparse matrix A_2 whose generating function has exactly the same zero curve as $f(x/2, y/2)$ is difficult, because the function $f(x/2, y/2)$ in general corresponds to a dense matrix, and even in special cases, matrices become denser on coarse grids. Therefore, f_2 is obtained by multiplying $f(x/2, y/2)$ with other cosine terms such that the result is a polynomial in $\cos(x)$ and $\cos(y)$. With such an approach, coarse grid matrices are difficult to compute if more than two-levels are used. Furthermore, matrices still become significantly denser on each level, because there are no trigonometric polynomials of low degree with exactly the same zero curve. Because of these difficulties, we do not use coarse grid functions with exactly the same zero curve as $f(x/2, y/2)$. Therefore, we approximate $f(x/2, y/2)$ by a trigonometric polynomial of low degree.

We start with a description of our approximation technique for the development of a two-grid method. Then, we explain how this two-grid idea can be applied recursively for the development of a multigrid method. Numerical examples conclude this section.

5.2.1 A two-level method based on rediscrization

In our new two-grid method, we use the damped Richardson, Jacobi, or Gauss-Seidel smoother and the restriction matrix $R = B \cdot E$ with E from (3.3.3) or (3.3.4) and B corresponding to b from (5.1.1) or (5.1.2). We define a function $f_2(x, y)$ in such a way that its zero curve approximates the one of $f(x/2, y/2)$. Furthermore, f_2 must be a trigonometric polynomial with only few nonzero coefficients such that the corresponding matrices are sparse. Such an approximation is carried out in two steps.

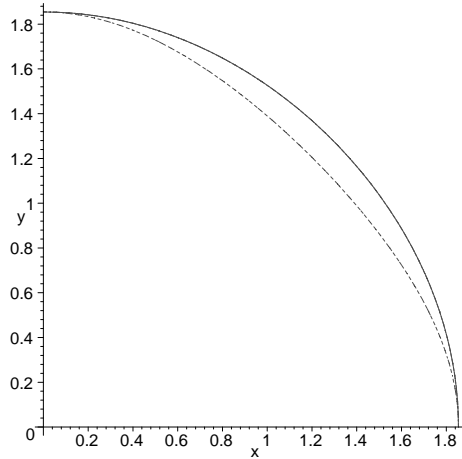


Figure 5.2: Zero Curves of $f(x/2, y/2)$ and $f_2^{(1)}(x, y)$ for $\rho = 1.6$

1. First, we choose a function $f_2(x, y)$ which is similar to the original function f and which has some free parameters.
2. In the second step, these parameters are computed such that the zero curve of f_2 shares some points with the curve $f(x/2, y/2) = 0$. A function f_2 with these two properties is expected to have a zero curve very similar to the one of $f(x/2, y/2)$. The number of free parameters should correspond to the number of points we want to fix on the curve, depending on the desired accuracy of the approximation. If f and f_2 have symmetry properties, their zero curves have additional points in common.

We illustrate the construction of the coarse grid function with our example function f , which is symmetric in x and in y . The first idea for the choice of f_2 is

$$f_2^{(1)}(x, y) = [\sigma - \cos(x) - \cos(y)]^2, \quad (5.2.1)$$

which only contains one parameter, and which results in very sparse matrices A_2 . We choose σ such that the zero curves of $f(x/2, y/2)$ and $f_2^{(1)}(x, y)$ share the point $(x_1, 0)$ with $x_1 = 2 \arccos(\rho - 1)$ on the positive x-axis. Then, from $f_2^{(1)}(x_1, 0) = 0$, we obtain that

$$\sigma = 1 + \cos(2 \arccos(\rho - 1)). \quad (5.2.2)$$

Because of the symmetries in f and $f_2^{(1)}$, also the points $(0, x_0)$, $(-x_0, 0)$, and $(0, -x_0)$ lie on both zero curves. For $\rho = 1.6$, the zero curves are shown in Figure 5.2, where the exact curve is drawn as a solid line and the approximation as a dotted line. Since this approximation is not accurate enough, we choose a slightly

more complicated f_2 by adding one additional term:

$$f_2^{(2)} = [\sigma - \alpha(\cos(x) + \cos(y)) - \beta \cos(x) \cos(y)]^2 . \quad (5.2.3)$$

We fix σ (e.g. by taking the value computed above) and use α and β as the free parameters. Now we determine α and β such that the zero curves of $f(x/2, y/2)$ and $f_2^{(2)}(x, y)$ have the following two points in common:

$$\begin{aligned} (x_1, y_1) &= (2 \arccos(\rho - 1), 0) \\ (x_2, y_2) &= (2 \arccos(\rho/2), 2 \arccos(\rho/2)) . \end{aligned}$$

Due to the symmetry mentioned above, the two zero curves have eight points in common. With the abbreviations

$$c_1 = \cos(2 \arccos(\rho - 1)) \quad \text{and} \quad c_2 = \cos(2 \arccos(\rho/2)) ,$$

the parameters α and β are computed from $f_2^{(2)}(x_1, 0) = 0$ and $f_2^{(2)}(x_2, x_2) = 0$:

$$\beta = \frac{c_2^2 \sigma - c_1 \sigma}{c_2^2 (c_1 + 1) - 2c_1 c_2} , \quad \alpha = \frac{\sigma}{c_1} - \frac{\beta(c_1 + 1)}{c_1} .$$

The zero curve of $f_2^{(2)}(x, y)$ is an excellent approximation to the one of $f(x/2, y/2)$. Even for a large zero curve, e.g. for $\rho = 1.6$, the two curves are almost equal. The zero curve of $f_2^{(2)}$ can hardly be distinguished from the curve $f(x/2, y/2) = 0$ in Figure 5.2. When we use A_2 corresponding to $f_2^{(2)}(x, y)$ in the two-grid method instead of the coarse grid matrix computed with the Galerkin approach, similar numerical results are obtained. For more complicated functions f with less symmetry or for extremely large matrix sizes, the approximation with $f_2^{(2)}$ may not be sufficiently close. In that case, we can use a third free parameter and a function of the form

$$f_2^{(3)} = [\sigma - \beta(\cos(x) + \cos(y)) - \alpha \cos(x) \cos(y) - \gamma(\cos(2x) + \cos(2y))]^2 , \quad (5.2.4)$$

with fixed σ and free α, β, γ . The third common point of the two zero curves is $(2x_3, x_3)$, leading to seven more common points due to the symmetry of f . In total, the two zero curves have 16 points in common. After computing

$$x_3 = 2 \arccos [(-1 + \sqrt{9 + 8\rho})/4] ,$$

the parameters are obtained from

$$f_2^{(3)}(x_1, 0) = 0 , \quad f_2^{(3)}(x_2, x_2) = 0 , \quad f_2^{(3)}(2x_3, x_3) = 0 .$$

This leads to the linear system

$$\begin{pmatrix} \cos(x_1) + 1 & \cos(x_1) & \cos(2x_1) + 1 \\ 2 \cos(x_2) & (\cos(x_2))^2 & 2 \cos(2x_2) \\ \cos(x_3) + \cos(2x_3) & \cos(x_3) \cos(2x_3) & \cos(2x_3) \cos(4x_3) \end{pmatrix} \cdot \begin{pmatrix} \beta \\ \alpha \\ \gamma \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \end{pmatrix} . \quad (5.2.5)$$

If $f_2^{(3)}$ is still not accurate enough, another term of the form

$$-\delta \cos(2x) \cos(2y)$$

is added in (5.2.4) within the square brackets. This adds another free parameter δ and forces the resulting $f_2^{(4)}$ to be zero at another point such as $(3x_4, x_4)$. If $f_2^{(4)}$ or even a trigonometric polynomial of slightly higher degree is used, the matrix A_2 is not denser as the one obtained from the Galerkin approach. The true benefit of our approximation technique will become apparent when we use several levels of restriction to define a multigrid method. Since we use this approximation-based rediscrretization approach on each level, the system matrix on coarser grids will not increase in bandwidth.

5.2.2 Extending the idea to the multigrid case

Multilevel methods based on the Galerkin approach suffer from an increasing density of the coarse grid matrices. This fact restricts the number of grids which can be used in a multigrid method to two or three. The rediscrretization-based method just described, on the other hand, do not suffer from these limitations. The two-grid method has the advantage that the coarse grid matrix A_2 has roughly the same density as the original matrix $A_n[f]$. The true benefit of our rediscrretization approach only becomes apparent when we construct a multigrid method with several coarsening levels. Such a method is defined by applying the two-grid idea from the previous subsection recursively. The only restriction concerning the number of levels in the multigrid method lies in the increased size of the zero curves. However, if ρ is close to 2, several levels can be used as described in Section 5.1.3. The multigrid method uses a standard smoother such as damped Jacobi on each level. In the following, we explain how restriction and coarse level matrices are computed.

- On the finest level, $b(x, y)$ is defined as in the Galerkin method with (5.1.1) or (5.1.2). Then, f_2 is computed as described in the previous subsection, using (5.2.3), (5.2.4), or an even better approximation.
- With this choice of f_2 , $b_2(x, y)$ is computed with (5.1.3) or the tau equivalent on the second finest level. The function $f_3(x, y)$, which corresponds to the system matrix A_3 on the next coarser level, is computed in a similar way as f_2 . f_3 is again a function of the form (5.2.3) or (5.2.4), but it must have common points with the curve $f(x/4, y/4) = 0$. With

$$\begin{aligned} x_1 &= 4 \arccos(\rho - 1), \quad x_2 = 4 \arccos(\rho/2), \\ x_3 &= 4 \arccos[(-1 + \sqrt{9 + 8\rho})/4], \end{aligned}$$

the coefficients are computed as above.

ρ	$\mathbf{n}=(2^6)^2$	$\mathbf{n}=(2^7)^2$	$\mathbf{n}=(2^8)^2$	$\mathbf{n}=(2^9)^2$
1.99	15	15	15	15
1.98	16	16	16	16
1.97	18	18	18	18

Table 5.2: Iteration numbers for the redscretization-based four-grid method applied to $C_{\mathbf{n}}[f]$ with f from Example 6

ρ	$\mathbf{n}=(2^6)^2$	$\mathbf{n}=(2^7)^2$	$\mathbf{n}=(2^8)^2$	$\mathbf{n}=(2^9)^2$
1.9995	15	15	15	15
1.9990	15	15	15	15
1.9985	20	20	20	20

Table 5.3: Iteration numbers for the redscretization-based six-grid method applied to $\tau_{\mathbf{n}}[f]$ with f from Example 6

- On the next level, we compute

$$b_3(x, y) = f_3(x + \pi, y) \cdot f_3(x, y + \pi) \cdot f_3(x + \pi, y + \pi)$$

and $f_4(x, y)$ as above. f_4 must have points in common with $f(x/8, y/8) = 0$.

- On coarser levels, this procedure continues until the zero curve

$$f(x/2^d, y/2^d) = 0$$

becomes too large, i.e. reaches the boundaries of $]-\pi/2, \pi/2[^2$.

The main advantage of this approach is that the matrices corresponding to f_2 have the same sparsity pattern as the matrices corresponding to f_3 , f_4 , and so on.

5.2.3 Numerical examples

In the following, we carry out numerical tests using matrices from the circulant and tau algebras corresponding to the function f from Example 6. For the coarse grid functions, we use an approximation in 8 points, i.e. $f_2^{(2)}$ from (5.2.3) and its analogs on coarser grids. First, we use a four-grid method for matrices of circulant type. Table 5.2 contains the number of V-cycle iterations for different values of ρ . If ρ becomes significantly smaller than 1.97, convergence deteriorates significantly. For problems where ρ is close to 2, even six levels can be used for the construction of a multigrid method. Table 5.3 summarizes the results for matrices of the tau algebra. For the DCT-III and DST-III algebras, similar iteration numbers are obtained.

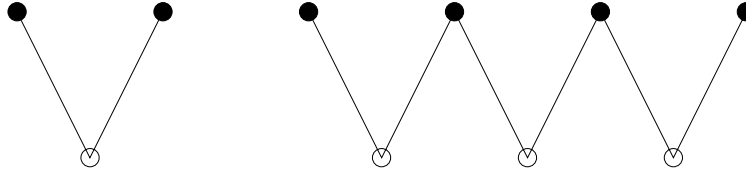


Figure 5.3: Two-grid iteration with one coarse grid correction (left) and with three coarse grid corrections (right)

5.3 A new splitting technique

All multigrid methods we have described in this chapter are based on computing exactly one system matrix on each grid: $A_1 = A_n[f]$ on the finest grid and A_2, A_3, A_4, \dots on coarser grids. If the original matrix A_n corresponds to a function $f(x, y)$ with a whole curve of zeros, this curve becomes larger on each grid, limiting the maximum number of levels for a multigrid method. So far,

- **we have** developed a method where all coarse grid matrices have the same sparsity pattern and do not become denser on coarser levels
- **we have not** found a remedy for the problem of increasing zero curves.

In the following, we propose a method which overcomes this problem. Instead of a single coarse grid correction, several coarse grid corrections are computed in every iterative step, each of them representing one part of the zero curve of f . We start with the description of a two-grid method, explaining the principal ideas of our approach. Then, we extend the two-grid method to a multigrid method and finally combine it with the multigrid methods from Sections 5.1 and 5.2. At the end of this section, we present numerical results to compare the different types of multigrid methods.

5.3.1 A two-grid method with splitting

In the Galerkin-based two-grid method from Chapter 5.1, the coarse grid correction X in each iteration is computed from one coarse grid matrix A_2 , which corresponds to the function $f_2(x, y)$. f_2 has the same zero curve as $f(x/2, y/2)$, and therefore represents the whole zero curve of $f(x, y)$ on the coarse grid. The left picture in Figure 5.3 shows one iteration of this two-grid method. The filled circles denote possible smoothing iterations, the empty circles represent the exact solution of the coarse grid problems. In our splitting-based method, we compute k coarse grid corrections $X_{1,j}$ with restriction matrices $R_{1,j}$ and coarse grid matrices $A_{2,j}$ in every iteration of the two-grid method. Each of the corresponding generating functions $f_{2,j}$ is a local coarse grid representation of the zero curve

$f(x/2, y/2) = 0$ in the neighborhood of at least one of its points (x_j, y_j) . The iteration matrix TG of the two-grid method is of the form

$$TG = S^{\nu_{k+1}} \cdot \prod_{j=1}^k (X_{1,j} \cdot S^{\nu_j}) \quad (5.3.1)$$

with smoother S and coarse grid corrections

$$X_{1,j} = I - R_{1,j}^H A_{2,j}^{-1} R_{1,j} A_{\mathbf{n}}[f] .$$

This means one iteration of the two-grid method consists of k coarse grid corrections $X_{1,j}$ and smoothers between the $X_{1,j}$. One iteration of this two-grid method is depicted in the right picture of Figure 5.3. It uses three coarse grid corrections with smoothing between them. There are two important goals when designing a two-grid method with several coarse grid corrections: *Each part of the zero curve must be approximated well and the number of coarse grid corrections per iteration must be small.*

Each coarse grid matrix $A_{2,j}$ is computed as follows. The restriction matrices are of the form $R_{1,j} = B_{1,j} \cdot E_1$, where B_j corresponds to a function $b_{1,j}(x, y)$ and E_1 is the elementary restriction matrix defined in (3.3.3) or (3.3.4). Since f from (5.0.1) is symmetric in x and y , b is chosen such that $f_{2,j}$ represents the zero curve of $f(x/2, y/2)$ in two or four point. A zero of f at (x_j, y_j) implies that f is also zero at $(-x_j, y_j), (x_j, -y_j), (-x_j, -y_j)$. If $b_{1,j}$ is chosen to be

$$b_{1,j}(x, y) = (\cos(x_j) + \cos(x))^2 (\cos(y_j) + \cos(y))^2 , \quad (5.3.2)$$

and if $A_{2,j}$ is computed with the Galerkin approach ($A_{2,j} = R_{1,j} A_{\mathbf{n}}[f] R_{1,j}^H$), then the zero curve of $f(x/2, y/2)$ is approximated very well in the neighborhood of the points $(2x_j, 2y_j), (-2x_j, 2y_j), (2x_j, -2y_j), (-2x_j, -2y_j)$. For a point $(x_j, 0)$ on the x -axis, $f_{2,j}$ is zero at $(-2x_j, 0), (2x_j, 0)$ and very small in the neighborhood of the two points. The left picture of Figure 5.4 shows the zero curves of $f(x, y)$ and $f(x/2, y/2)$, and the level curve $f_{2,j}(x, y) = 0.005$. From this picture we see that large parts of the zero curve of $f(x/2, y/2)$ are approximated well by $f_{2,j}$. The following theorem summarizes properties of $A_{2,j}$ and $f_{2,j}$.

Theorem 20

Let $f(x, y)$ be the function defined in (5.0.1) with $\rho > 1$, and $A_{\mathbf{n}}[f]$ the corresponding matrix of the two-level tau, circulant or DCT-III algebra. Furthermore, let (x_j, y_j) be a point on the zero curve of $f(x, y)$, and assume that $R_{1,j} = B_{1,j} \cdot E_1$, with $B_{1,j}$ corresponding to $b_{1,j}$ from (5.3.2). Moreover, assume that $A_{2,j}$ is computed with the Galerkin approach. Then the following holds:

1. $A_{2,j}$ is a symmetric positive definite matrix. Its generating function $f_{2,j}$ has zeros at

$$(2x_j, 2y_j), (-2x_j, 2y_j), (2x_j, -2y_j), (-2x_j, -2y_j)$$

and it is strictly positive elsewhere in $]-\pi, \pi]^2$.

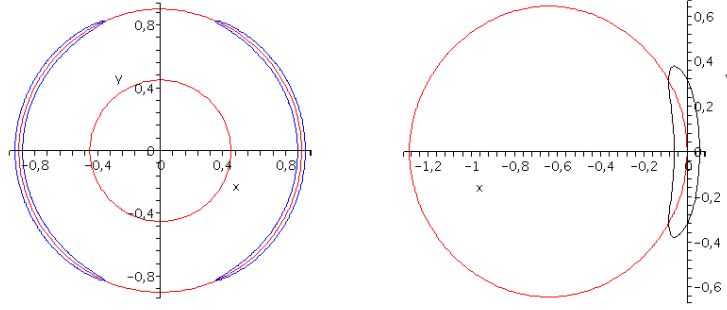


Figure 5.4: Zero curves of $f(x, y)$ and $f(x/2, y/2)$ with the level curve $f_{2,j} = 0.005$ approximating $f(x/2, y/2) = 0$ in the neighborhood of the points $(\pm x_j, 0)$ (left) and approximation after shifting a zero to the origin (right)

2. In the direction t of the tangent on $f(x/2, y/2)$ in (x_j, y_j) , the first directional derivatives of $f(x/2, y/2)$ and $f_{2,j}(x, y)$ are both zero.

Proof: From [3] we know that $A_{2,j}$ is the matrix of the respective algebra corresponding to $f_{2,j}$. Since $f_{2,j}$ is real-valued even and nonnegative, $A_{2,j}$ is symmetric positive definite. Due to $\rho > 1$ the curve $f(x, y) = 0$ is located within $]-\pi/2, \pi/2[$. Then, by the results from [3], $f_{2,j}$ is zero at the four points. From (3.3.5) we can deduce that $f_{2,j}$ is strictly positive at all other points.

For zeros on the x - and y -axis, the result stated in (2) follows from direct calculation, for other zeros, it is obtained by a simple coordinate transformation.

■

In the Toeplitz and circulant case, another way to perform restriction is the following.

- The zero curve of f is shifted such that (x_0, y_0) moves to the origin. This corresponds to a diagonal transformation \tilde{A} of $A_n[f]$.
- Then, the restriction matrix $\tilde{R}_{1,j} = B_{1,j} \cdot E_1$ is applied to \tilde{A} , where E_1 is the elementary restriction matrix of the class and $B_{1,j}$ corresponds to a function of the form

$$b_{1,j}(x, y) = (1 + \cos(x))^2(1 + \cos(y))^2. \quad (5.3.3)$$

- The coarse grid matrix $A_{2,j}$ is computed with the Galerkin approach, i.e. $A_{2,j} = \tilde{R}_{1,j} \tilde{A} \tilde{R}_{1,j}^H$.

The right picture in Figure 5.4 shows the result on the corresponding generating function $f_{2,j}$, depicting the level curve $f_{2,j}(x, y) = 0.01$. $f_{2,j}$ is zero only at $(0, 0)$ and very small along the zero curve of $f(x/2, y/2)$ in the neighborhood of the

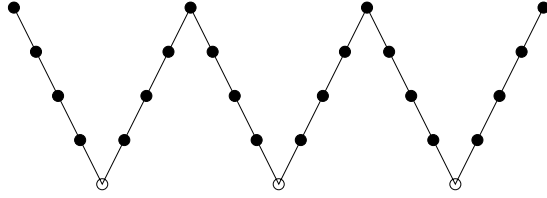


Figure 5.5: One iteration of a five-grid method with a splitting into three subproblems on the finest grid

origin. Thus, each coarse grid matrix corresponds to a nonnegative generating function with an isolated zero at the origin. This property will be very useful for the development of a multigrid method in the next section. In the tau and DCT-III algebras, we cannot perform diagonal transformation because the resulting matrix \tilde{A} would not be in the same matrix class. Therefore, we apply restriction directly to the matrix $A_n[f]$. Shifting the zeros to the origin in the Toeplitz and circulant case has one main advantage and one main disadvantage:

- The **advantage** is that the zero remains at the origin on coarser levels. Thus, no further complications arise from the zero, and a large number of coarsening steps can be used in the multigrid method.
- The **disadvantage** is that the number of coarse grid corrections is significantly higher if the generating function is symmetric. In the case of f from (5.0.1), this means 8 corrections have to be used instead of 3 or 16 instead of 5.

5.3.2 From two-grid to multigrid

The two-grid method becomes a multigrid method if each coarse grid system in (5.3.1) with the matrix $A_{2,j}$ is solved recursively with the multigrid scheme from Chapter 3. This implies that one iteration of the multigrid method consists of k V-cycles instead of one in the method from Chapter 5.2. Figure 5.5 shows one iteration of the multigrid method which uses three coarse grid corrections and five grids for each of them. Since $A_{2,j}$ corresponds to a generating function $f_{2,j}$ with isolated zeros, the functions $b_{2,j}, b_{3,j}$, etc., corresponding to the restriction matrices on coarser levels, are of the form (5.3.2). Each zero $(2x_j, 2y_j)$ of $f_{2,j}$ moves to $(4x_j, 4y_j), (8x_j, 8y_j), \dots$ on the next levels. Thus, extra care has to be taken if either the x - or the y -value of the zero approaches $\pi/2 \pmod{\pi}$.

As described above, the problem of moving zeros can be avoided in the Toeplitz and circulant case by treating each zero (x_j, y_j) separately. This is done by shifting (x_j, y_j) to the origin and then using a function such as

$$b_{i,j}(x, y) = (1 + \cos(x))^2(1 + \cos(y))^2 \quad (5.3.4)$$

for restriction. From the form of the level curves of $f_{2,j}$, we see that the coarse grid matrices are of slightly anisotropic type. Especially when ρ becomes smaller, some of the coarse grid matrices have significant anisotropies. In these cases, we can use the results from [55] and apply one or two semicoarsening steps. This means that (5.3.2) is replaced by functions such as

$$b_{i,j}(x, y) = (\cos(x_0) - \cos(x))^2 .$$

Elementary restriction is done only in one direction, i.e. the matrix size is reduced only by a factor of 2.

For the design of an efficient multigrid method, the number of coarse grid corrections is critical. For zeros of $f(x, y)$ on the x - or y -axis, one coarse grid correction approximates the zero curve of $f(x/2, y/2)$ in the neighborhood of two points, for all other zeros, in the neighborhood of four points. Since we usually start with the zeros on the axes, this means that $\frac{k}{4} + 1$ coarse grid corrections X_j are needed for an approximation in k equidistant points. Thus, two corrections are needed for 4 points, three corrections for 8 points, and five corrections for 16 points. The number of necessary points depends on two factors, the size of the zero curve and the size of the matrices.

5.3.3 Combining splitting with the Galerkin method

So far, we have defined two different kinds of multigrid methods for matrices corresponding to generating functions with whole zero curves. Both strategies have advantages and disadvantages:

- (I) The strategy from Section 5.2 uses only one coarse grid matrix on each grid. The main advantage of this method is that the zero curve of f is represented very well by generating functions on coarser grids. This leads to fast convergence. The main disadvantage is the increase of the size of the zero curve on each grid, limiting the possible number of levels in our multigrid method.
- (II) The splitting strategy from Sections 5.3.1 and 5.3.2 has the main advantage that zero curves do not grow, and therefore a much larger number of levels can be used. The disadvantage of these methods is that for very large matrices, an approximation in 4 or 8 points is not accurate enough, and therefore k may become too big for a fast algorithm.

Therefore, we suggest to apply a multigrid method which combines the advantages of both types. We use the following heuristic: *Start with the first strategy until the zero curve is too large for another coarsening step. Then split the resulting coarse grid problem into k subproblems and apply further levels of restriction to each of them.* To illustrate this new strategy, Figure 5.6 depicts one multigrid iteration. It starts with two coarsening steps using the first strategy and then computes three

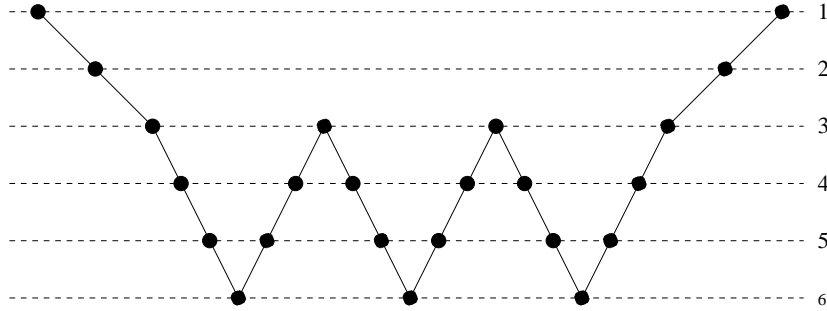


Figure 5.6: One iteration of a six-grid method with a combined Galerkin-splitting technique

coarse grid corrections on three further levels, leading to an approximation in the neighborhood of 8 points. The main advantage of this method is the following. After a few coarsening steps with strategy **(I)**, we obtain a matrix which is considerably smaller than $A_n[f]$, because in each step, the number of rows and columns is reduced by a factor 4 each. Thus, a slightly larger number of necessary coarse grid corrections (due to the increased size of the zero curve after a few coarsening steps) is still acceptable.

In many applications, discretization of the Helmholtz equation results in a function f from (5.0.1) with $1.99 < \rho < 2$. In this case, we can apply several coarsening steps with strategy **(I)** before the zero curve reaches the boundaries of $] - \pi/2, \pi/2[$. Then, we switch to strategy **(II)**.

5.3.4 Numerical results

With the following numerical tests we wish to illustrate that quite a small number of coarse grid problems are sufficient for fast multigrid convergence. Moreover, the pure splitting technique shall be compared with the combined strategy. To illustrate that indeed a fairly small number of coarse grid corrections is needed, we have tested our method for f from (5.0.1) and the corresponding two-level Toeplitz matrices. Table 5.4 contains the number of necessary coarse grid corrections ($\#cgc$) and the iteration numbers for a three-grid method. Since each iteration contains two or three V-cycles, it roughly corresponds to two or three iterations of a method from Section 5.2.

The numerical results contained in Table 5.5 are obtained with the combined method for two-level DST-III matrices corresponding to f from (5.0.1). For $\rho = 1.95$, we test two different methods. Method 1 uses two steps of the first strategy followed by splitting into three subproblems and one further level of prolongation. Method 2 uses one step of the first strategy followed by splitting into three subproblems and two more coarsening steps.

ρ	# <i>cgc</i>	$n=(2^5)^2-1$	$n=(2^6)^2-1$	$n=(2^7)^2-1$	$n=(2^8)^2-1$
1.95	2	4	4	4	4
1.9	2	6	6	6	6
1.8	3	4	4	4	4
1.6	3	5	5	5	6

Table 5.4: Iteration numbers for the splitting-based three-grid method applied to $T_n[f]$ with f from Example 6

ρ	method	$n=(2^5)^2$	$n=(2^6)^2$	$n=(2^7)^2$	$n=(2^8)^2$
1.95	1	13	13	14	14
1.95	2	9	9	10	10

Table 5.5: Iteration numbers for the combined four-grid method applied to $S_n[f]$ with f from Example 6

5.4 Anisotropic problems with zero curves

The whole Chapter 4 was devoted to the development of multigrid methods for anisotropic systems whose generating functions have isolated zeros. Now we wish to describe multigrid methods for anisotropic problems with whole zero curves.

Example 7 Let $A_n[f]$ be the two-level matrices belonging to a trigonometric algebra or to the Toeplitz class which correspond to the generating function

$$f(x, y) = (\rho - \alpha \cos(x) - (2 - \alpha) \cos(y))^2 \quad (0 < \rho \leq 2, \alpha \ll 1) . \quad (5.4.1)$$

For $\rho < 2$, f is zero along a whole curve. This zero curve becomes extremely flat if α tends to zero.

As their isotropic counterparts, these systems can be solved with two fundamentally different types of multigrid methods. Again, the first type is based on using a classical V-cycle with one coarse grid matrix on each level, whereas the second type splits the original problem into several coarse grid problems.

5.4.1 Galerkin-based methods

The starting point for the development of multigrid methods for this class of matrices is again the classical Galerkin approach. As for the simpler case of Chapter 4, multigrid methods shall be defined as a combination of semicoarsening and full coarsening steps. If anisotropy occurs along the x -axis, semicoarsening steps are performed in y -direction. The prolongation matrix on the finest level corresponds to

$$b(x, y) = f(x, \pi - y) , \quad (5.4.2)$$

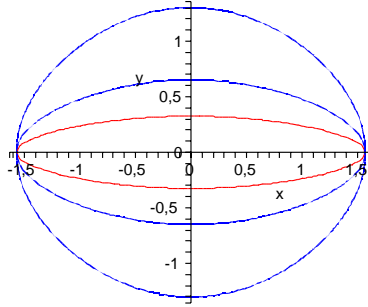


Figure 5.7: Zero curve of $f(x, y)$, $f(x, y/2)$, and $f(x, y/4)$ for f from Example 7

which is the anisotropic analog of (5.1.2). Elementary restriction for this semicoarsening step is done with (4.2.3). In the case of anisotropy in y -direction, semicoarsening is done with

$$b(x, y) = f(\pi - x, y) \quad (5.4.3)$$

and with elementary restriction in x -direction. We continue with semicoarsening until there is no more anisotropy, i.e. in Example 7 until zero curves are close to circles, and then switch to full coarsening as described in Section 4.2. Figure 5.7 illustrates what happens to the zero curve of f when two semicoarsening steps are applied. The solid curve is the zero curve of $f(x, y)$ with $\rho = 1.9$ and $\alpha = 0.1$, whereas the two dotted curves are the zero curves of f_2 and f_3 , which are equivalent to the zero curves of $f(x, y/2)$ and $f(x, y/4)$, respectively.

As in Chapter 4, we state two theoretical results. One of them describes the reduction of anisotropy and the other one proves convergence of the two-grid method. To quantify the reduction of anisotropy, we analyze zero curves instead of the more general level curves from Chapter 4. This implies that Theorem 10 can be greatly simplified. We define the ratios r_F and r_C in a slightly different way as in Chapter 4. The ratio $r_F = \frac{x_F}{y_F}$ measures the degree of anisotropy of the given matrix, where $(x_F, 0)$ and $(0, y_F)$ denote the points where the zero curve $f(x, y) = 0$ intersects the coordinate axes for positive x_F and y_F . $r_C = \frac{x_C}{y_C}$ is the same ratio for f_2 on the coarse grid.

Theorem 21

Let f be a nonnegative generating function with a zero curve of order 2. Let f_2 be the function obtained by one semicoarsening step with b from (5.4.2). Moreover, let $r_F = \frac{x_F}{y_F}$ and $r_C = \frac{x_C}{y_C}$ be the ratios described above for f and f_2 , respectively. Then the degree of anisotropy is reduced by a factor 2, i.e. $\frac{r_F}{r_C} = 2$.

Proof: Since semicoarsening is done in y -direction, the zero curve of f_2 is given by $f(x, y/2) = 0$. Hence, we obtain $x_C = x_F$ and $y_C = 2 \cdot y_F$, and therefore $\frac{r_E}{r_C} = 2$. ■

The second theoretical result concerns convergence of the two-grid method. It is the anisotropic analogue of Theorem 19.

Theorem 22

Let $A := A_n[f]$ be a two-level matrix from the circulant, tau, DCT-III, or DST-III algebra. Assume that $f(x, y)$ is a cosine nonnegative polynomial (not identically zero) with a zero curve in $]-\frac{\pi}{2}, \frac{\pi}{2}[^2$. Suppose that the smoother satisfies condition (3.1.17). Furthermore, let the restriction be defined by $R = B \cdot E$ with B corresponding to $b(x, y)$ from (5.4.3), and the elementary restriction matrix E of the respective algebra. Then, there exists $\gamma > 0$ such that condition (3.1.18) is satisfied and the two-grid method converges.

Proof: This theorem is proved by combining the proofs of Theorems 12 and 19. The smoothing condition here is the same as in the two theorems. The correcting condition is proved for \mathbf{y} defined in (4.2.15), which leads to the inequality (4.2.16). In the circulant case, block diagonalization of the matrices involved in (4.2.16) is done with $\tilde{Q}^{(circ)}$ from (4.2.18). Due to the continuity of f and b , this reduces to n_1 2-by-2 matrix inequalities of the form (4.2.20). Since f and b satisfy conditions (3.3.22) and (3.3.23) for each point on the zero curve, these inequalities hold. For the classes of tau, DCT-III and DST-III matrices, the proof is carried out analogously. ■

The Galerkin method which uses exact zero curves converges optimally, but suffers from the same two problems as in the isotropic case, increasing zero curves and denser matrices on coarser levels. First, we focus on the second problem and define a method with coarse grid matrices which are less dense. The first idea, which is only applicable to some problems, is to choose

$$b(x, y) = \sqrt{f(x, y + \pi)} \tag{5.4.4}$$

instead of (5.4.2).

Remark 16 If $b(x, y)$ from (5.4.4) is a trigonometric polynomial of low degree, i.e. if the corresponding matrices are sparse, then the two-grid method does not only converge optimally. Moreover, the coarse grid matrix is still as sparse as $A_n[f]$, i.e. we have no increase in bandwidth. Hence, we can use this choice of b for the Galerkin-based method.

This is, for example, possible for the matrices in Example 7. However, this approach will be limited to two-grid methods, because on coarser levels, it is even more difficult to construct $\sqrt{f_2(x, y + \pi)}$ explicitly. In addition, b from (5.4.4) satisfies (3.3.22), but not (3.3.24). Again, this is enough to prove two-grid convergence and level independency, but not convergence of the V-cycle. Nevertheless,

this choice of b can be useful for the construction of a multigrid method. The first coarsening step is carried out with b from (5.4.4), and the following steps on coarser grids with the methods described in the following.

5.4.2 Rediscretization and splitting techniques

As in the isotropic case, we do not require that the coarse grid functions have exactly the same zero curve as $f(x, y/2)$, $f(x, y/4)$, etc. In this subsection, we present two alternative methods, similar to the ones from Section 5.2 and 5.3 for isotropic problems. The first of these methods uses one coarse-grid correction in each V-cycle iteration, and the coarse grid matrices are obtained from rediscrctization on coarser grids. The second method is based on splitting the original problem and computing several coarse grid corrections in each iteration.

For the construction of the first method, we use a trigonometric polynomial f_2 as the coarse grid function, which is similar to the function f and whose zero curve is a good approximation to the one of f . In the anisotropic case, f has less symmetry properties, which means that more common points are necessary to obtain the same degree of approximation. $f(x, y)$ from Example 7 is not symmetric with respect to the line $y = x$. We compute approximations for f_2 in a similar way as with (5.2.1), (5.2.3), and (5.2.4) in Section 5.2.1. The zero curves of $f(x, y/2)$ and $f_2(x, y)$ have 4 common points if we use a function with two free parameters, which are computed at points $(x_1, 0)$ and $(0, y_2)$. Since this approximation is not good enough, we choose

$$f_2(x, y) = [2 - \alpha \cos(x) - \beta \cos(y) - \gamma \cos(x) \cos(y)]^2, \quad (5.4.5)$$

which has three free parameters. These are computed by requiring that f_2 has zeros at $(x_1, 0)$, (x_2, x_2) and $(0, y_3)$. By symmetry on both axes, $f(x, y/2)$ and $f_2(x, y)$ have 8 common points. For 16 common points we need a function with five parameters such as

$$f_2(x, y) = [2 - \alpha \cos(x) - \beta \cos(y) - \gamma \cos(x) \cos(y) - \delta \cos(2x) - \epsilon \cos(2y)]^2, \quad (5.4.6)$$

which, in addition to the three points from above, has to be zero at $(2x_4, x_4)$ and $(x_5, 2x_5)$. As in the isotropic case, this rediscrctization technique is applied recursively. On each level, the restriction matrix is constructed with a function b_j such as

$$b_j(x, y) = f_j(\pi - x, y) \quad \text{or} \quad b_j(x, y) = f_j(x, \pi - y). \quad (5.4.7)$$

The multigrid method based on splitting is of the form (5.3.1). As in Section 5.3, there are two options for the construction of the subproblems:

- For k of points x_j on the zero curve of f , we shift x_j to the origin, i.e. apply diagonal transformation to the matrix $A_n[f]$. Then, we use the function $b_{1,j}$ from (5.3.3) for restriction. This shifting technique, which results in

ρ	$\mathbf{n}=(2^5-1)^2$	$\mathbf{n}=(2^6-1)^2$	$\mathbf{n}=(2^7-1)^2$
1.9	14	14	14
1.8	14	14	14
1.6	15	15	15

Table 5.6: Two-grid iteration numbers for $T_{\mathbf{n}}[f]$ with f from Example 7 and $\alpha = 0.1$

subproblems with zeros at the origin on all grids, can only be used for Toeplitz and circulant matrices.

- We define k restrictions $R_{1,j}$ with $b_{1,j}$ from (5.3.2). Because of the symmetry of f , each of the resulting coarse grid matrices $A_{2,j}$ corresponds to a function $f_{2,j}$, which approximates the zero curve of $f(x, y/2)$ in the neighborhood of at least two points. However, on coarser grids, the isolated zeros of the subproblems are not located at the origin. On the other hand, a smaller number of coarse grid problems is sufficient.

The structure of one V-cycle, which is of the same form as in Section 5.3, is depicted in Figure 5.5. As in Section 5.3, the splitting technique can be combined with both the Galerkin and the discretization approach. The structure of one multigrid iteration is the same as in Figure 5.6. The difference is that we start with semicoarsening steps until the linear system is not anisotropic anymore, i.e. until the zero curve of the generating function is almost a circle.

5.4.3 Numerical results

In Theorem 22, we have proved optimal convergence of the two-grid method constructed with the Galerkin approach. This convergence behavior shall be illustrated with the function f from Example 7 and the parameter $\alpha = 0.1$. Table 5.6 contains the iteration numbers of the two-grid method for different values of ρ .

Following Theorem 21, we apply semicoarsening until the zero curve is almost a circle. This means coarsening is done only in one direction. For the example function f , the direction of coarsening is the y -axis. Moreover, semicoarsening can only be applied until the zero curve reaches the line $y = \pi/2$, i.e. as long as

$$\max_{(x,y) \in f(x,y)=0} y < \pi/2 \ .$$

Otherwise, the function f is also zero at the mirror points of some of its zeros, and the multigrid method diverges for the reasons described in Remark 8. For f from Example 7 with $\alpha = 0.1$ and $\rho = 1.9$ this implies that two semicoarsening steps can be applied (see Figure 5.7). Then, the zero curve of $f_3(x, y)$ is too large for a further semicoarsening or full coarsening step, and splitting must be used.

α	$\#(\text{semi})$	$\mathbf{n}=(2^6-1)^2$	$\mathbf{n}=(2^7-1)^2$	$\mathbf{n}=(2^8-1)^2$
0.05	2	21	23	24
0.05	3	18	18	18
0.1	2	24	24	24
0.1	3	21	21	21

Table 5.7: Five-grid iteration numbers for $\tau_{\mathbf{n}}[f]$ with f from Example 7, $\alpha = 0.1$, and $\rho = 1.99$

If $\alpha = 0.01$ and $\rho = 1.95$, three semicoarsening steps can be used. For this case, we apply the multigrid method based on rediscrretization, which uses the functions b_j from (5.4.7) on the three finest levels. Then, we must switch to splitting. For $\alpha = 0.01$ and $\rho = 1.99$, we can apply three semicoarsening steps followed by one full coarsening step can be used, before we switch to splitting. We use this last case for further numerical tests. A four-grid method is applied to the tau matrices corresponding to f from Example 7 with $\alpha = 0.05$ and $\rho = 1.99$. Table 5.7 shows the iteration numbers of the V-cycle. As our heuristic suggests, three semicoarsening steps is the best choice. In the case of $\alpha = 0.1$, best results are obtained with two semicoarsening steps followed by one full coarsening step.

5.5 Multigrid as a preconditioner

As described in Chapter 3, multigrid methods are not only used as solvers, but also as preconditioners for Krylov subspace methods. Since the matrices we consider in this chapter are Hermitian positive definite, the pcg method is used as the solver. The aim of a multigrid preconditioner P is to obtain a matrix $P^{-1}A_{\mathbf{n}}$, which is significantly less ill-conditioned than $A_{\mathbf{n}}$. On the other hand, application of the preconditioner must be less costly than the solution of the original problem.

Here, we develop different kinds of multigrid preconditioners for linear systems whose generating function has a zero curve. Some of these methods are based on the ideas presented in Sections 5.1 and 5.2, others on splitting the original problem into a fixed number of subproblems. After introducing preconditioners of both types, we describe how they can be combined to a preconditioner with the advantages of both types. This is similar to the combination of different types of multigrid solvers in Section 5.3.3.

5.5.1 A Galerkin- or rediscrretization-based preconditioner

The first type of preconditioner computes only one coarse grid correction in each iteration of the cg method. This coarse grid correction represents the whole zero curve of f . We apply the multilevel diagonal scaling preconditioner, which was introduced in Section 3.1.3. Although this additive preconditioner leads to slightly

ρ	$\mathbf{n}=(2^5-1)^2$	$\mathbf{n}=(2^6-1)^2$	$\mathbf{n}=(2^7-1)^2$
1.9	28	28	28
1.8	32	32	32
1.6	41	41	41

Table 5.8: Iteration numbers of the pcg method with the Galerkin-based two-grid preconditioner, applied to $\tau_{\mathbf{n}}[f]$ with f from Example 6

slower convergence of the pcg method than a multiplicative one, it is significantly more efficient as a parallel preconditioner. Moreover, it is ideally suited to study the convergence optimality of the multigrid preconditioner based on semicoarsening. The two-grid version of the diagonal scaling method is of the form (3.1.25), and the multigrid version is obtained by approximating A_C^{-1} recursively with the two-grid method. With the notation used in this chapter, the three-grid method is of the form

$$P = \text{diag}(A)^{-1} + R_1^H (\text{diag}(A_2)^{-1} + R_2^H A_3^{-1} R_2) R_1. \quad (5.5.1)$$

The first way of computing the coarse-grid matrices A_j follows the strategy applied in Section 5.1, i.e. the Galerkin approach. On each grid, the restriction matrix $R_j = B_j E_j$ consists of the elementary restriction matrix E_j and of the matrix B_j corresponding to

$$b_j(x, y) = f_j(\pi - x, y) \cdot f_j(x, \pi - y) \cdot f_j(\pi - x, \pi - y). \quad (5.5.2)$$

Since this choice of the matrices R_j results in coarse grid functions with exactly the same zero curve as $f(x/2, y/2)$, $f(x/4, y/4)$, etc., we expect optimal convergence of the pcg method, independent of the matrix size.

Table 5.8 illustrates that the two-grid preconditioner indeed leads to optimal convergence of the pcg method. However, the multigrid version of the Galerkin-based preconditioner is more a theoretical construction than a practical algorithm, because the matrices A_j become much denser as j increases. Nevertheless, the method is the starting point for the development of all other multigrid preconditioners presented in this section.

The rediscrretization-based preconditioner uses the same kind of restriction matrices as the Galerkin method. The coarse grid functions, on the other hand, are approximations for $f(x/2, y/2)$, $f(x/4, y/4)$, etc. We use the same heuristic as in Section 5.2 to keep matrices sparse on coarser levels. The restriction matrices correspond to (5.5.2), but the coarse grid matrices are computed with rediscrretization on each grid. This is done by choosing the functions f_j as in (5.2.3) or (5.2.4), depending on the number of points on which the zero curves should match. The great advantage of this method is that the sparsity pattern of the matrices is the same on all grids. Therefore, the number of grids is only limited by the size of the

zero curve, which becomes larger on coarser grids. This problem is addressed in the following subsection.

5.5.2 Approximation with auxiliary problems

The second type of preconditioner is constructed by splitting the original problem, where f has a whole zero curve, into a fixed number k of auxiliary problems, each of them corresponding to a generating function $f_{1,j}$ with isolated zeros. Then, the preconditioner P , which shall be described in the following, is the sum of k auxiliary problems, which are then replaced by coarse grid corrections. In more detail, P is constructed as follows:

- On the finest level, choose k matrices $A_{1,j}$ corresponding to nonnegative generating functions $f_{1,j}$ with isolated zeros, which approximate the zero curve of f in the neighborhood of at least one point. The linear systems involving $A_{1,j}$ are significantly easier to solve than the system involving A . The one-grid preconditioner, which is of course not used in practice, is of the form

$$P = A_{1,1}^{-1} + \cdots + A_{1,k}^{-1} . \quad (5.5.3)$$

- Instead of explicitly computing $A_{1,j}^{-1}$, we approximate it with the two-grid correction $R_{1,j}^H A_{1,j}^{-1} R_{1,j}$. Doing that, we obtain the two-grid preconditioner

$$P = c \cdot I + R_{1,1}^H A_{1,1}^{-1} R_{1,1} + \cdots + R_{1,k}^H A_{1,k}^{-1} R_{1,k} , \quad (5.5.4)$$

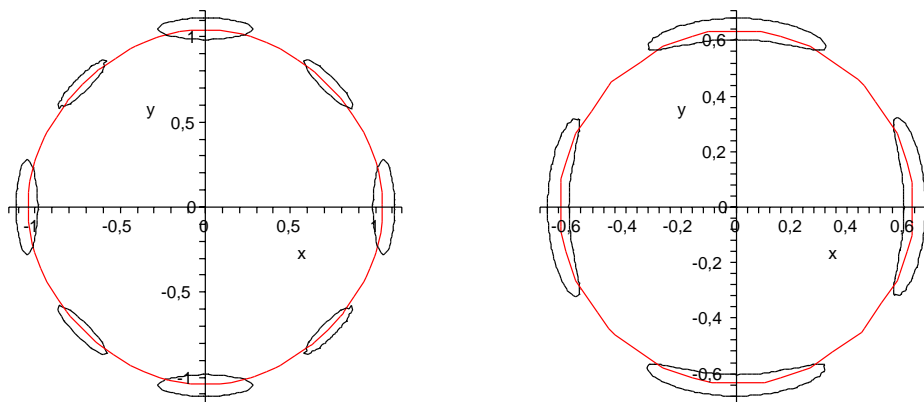
where each auxiliary problem $A_{1,i}$ is projected to the coarser grid with $R_{1,i}$. The constant c is chosen to be zero unless the matrix P is singular. In that case, c must be a small positive constant, such that P is regular, but still a good approximation for A^{-1} .

- P can easily be extended to a multilevel preconditioner by restricting the matrices $A_{1,i}$ to $A_{2,i}$ on the next coarser level and by replacing each $R_{1,i}^H A_{1,i}^{-1} R_{1,i}$ in (5.5.4) by

$$R_{1,i}^H (\text{diag}(A_{1,i})^{-1} + R_{2,i}^H A_{2,i}^{-1} R_{2,i}) R_{1,i} .$$

This procedure can be carried out on several levels, until the matrix on the coarsest level can be inverted exactly.

The number of auxiliary problems mainly depends on the size of the zero curve of f and on the matrix size. For f with a small zero curve and medium or large size matrices, four auxiliary problems are enough to get fast convergence. However, if matrix sizes become extremely large, and if zero curves also become larger, 8 or 16 auxiliary problems are necessary.



(a) 8 auxiliary problems corresponding to f_j (b) 4 auxiliary problems corresponding to h_j

Figure 5.8: Zero curve of $f(x, y)$ from Example 6 and level curves of 8 auxiliary problems

5.5.3 The choice of the generating functions

The crucial task when designing an additive multilevel preconditioner with (5.5.4) is the choice of the $A_{1,j}$. There will always be a tradeoff between approximating the zero curve of f well and picking auxiliary problems which are easy to compute. In this section, we choose matrices $A_{1,j}$ corresponding to generating functions with a single isolated zero in $] - \pi, \pi[$, which can be solved with the methods from Chapters 3 and 4. We describe three different choices for the matrices $A_{1,j}$ in the following:

1. A first idea for the choice of the $A_{1,j}$ is the shifted discrete Laplacian, which is described by

$$f_{1,j}(x, y) = 2 - \cos(x - x_j) - \cos(y - y_j) \quad (i \in \{1, \dots, k\}) \quad (5.5.5)$$

with the points (x_j, y_j) on the zero curve of f . Alternatively, one can use an anisotropic version of (5.5.5), which is a better approximation to the zero curve. Figure 5.8(a) shows how 8 anisotropic auxiliary problems approximate the zero curve of f , each of them being zero in one point and having small values in its neighborhood.

The three generating functions with zeros in the top right quarter are

$$\begin{aligned} f_{1,1}(x, y) &= \alpha(1 - \cos(x - x_1)) + 1 - \cos(y - y_1) \\ f_{1,2}(x, y) &= \alpha(1 - \cos(x + y - x_2 - y_2)) + 1 - \cos(x - y - x_2 + y_2) \\ f_{1,3}(x, y) &= 1 - \cos(x - x_3) + \alpha(1 - \cos(y - y_3)) . \end{aligned} \quad (5.5.6)$$

The others are defined analogously, with anisotropy occurring in the direction of the tangent to the zero curve at the respective point (x_j, y_j) . Each of these functions is a good approximation in the neighborhood of its zero, but this neighborhood is still too small for the construction of an efficient preconditioner (5.5.4).

2. A slightly more sophisticated choice for the $A_{1,j}$ is obtained by defining generating functions $f_{1,j}$ which are zero at one point (x_j, y_j) of the zero curve of f and whose partial derivatives in the point (x_j, y_j) are the same as the ones of f . Since the resulting matrices should be sparse, the functions $f_{1,j}$ must be polynomials in sine and cosine. For the example function f from (5.0.1), we choose

$$f_j(x, y) = a + b \cdot \cos(x) + c \cdot \sin(x) + d \cdot \cos(y) \quad (5.5.7)$$

for a point which is located on the x -axis. For points on the y -axis we interchange x and y in (5.5.7), and for points with $x_j = y_j$ we replace x by $x - y$ and y by $x + y$. For the point $(x_j, 0)$, the coefficients in (5.5.7) are computed with the simple conditions

$$a + b \cos(x_j) + c \sin(x_j) + d = 0 ,$$

obtained from $f_{1,j}(x_j, 0) = 0$, and

$$d = \frac{b \sin(x_j) - c \cos(x_j)}{\sin(x_j)} ,$$

obtained by comparing $\frac{d^2 x}{dy^2}$ of both functions. Since the first and the third derivative do not yield further equations, we can, for example, compute the third condition by comparing $\frac{d^4 x}{dy^4}$ and the fourth condition by minimizing the sum of the absolute values of the four coefficients. The approximation obtained with these functions $f_{1,j}$ is superior to (5.5.5), but still not good enough for practical application.

3. A significantly better approximation is obtained by adding the original function f to each of the auxiliary functions $f_{1,j}$. This is useful for both types of auxiliary functions defined above. The matrices become slightly denser, but the generating functions still have a single isolated zero, which makes them easy to solve. The generating functions can be written

$$h_{1,j}(x, y) = \gamma_{j_1} f_{1,j}(x, y) + \gamma_{j_2} f(x, y) \quad (j \in \{1, \dots, k\}) \quad (5.5.8)$$

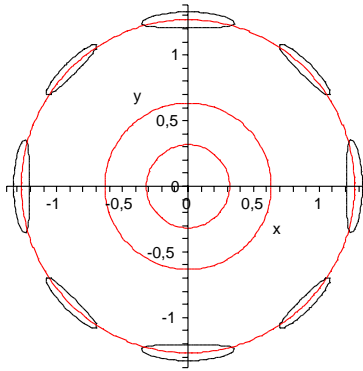


Figure 5.9: Zero curves of f , f_2 , and f_3 for f from Example 6 and level curves of auxiliary problems $h_{3,j}(x, y)$, which approximate the curve $f_3(x, y) = 0$

with $f_{1,j}(x, y)$ defined in (5.5.6) or (5.5.7). Typically, γ_{j_1} and γ_{j_2} are both chosen to be 0.5. A larger value of γ_{j_2} results in a better approximation to the zero curve of f , but in an auxiliary problem which is less efficiently solved with the methods from Chapters 3 and 4. The functions $h_{1,j}$ are less anisotropic than the $f_{1,j}$, but it is still advantageous to apply semicoarsening on the first one or two grids of the preconditioner. Figure 5.8(b) shows that the functions $h_{1,j}(x, y)$ are far better approximations to the zero curve of f , although they only have a single isolated zero.

5.5.4 Combining the Galerkin approach with auxiliary problems

As for multigrid solvers, there is a possibility of combining the two preconditioning strategies described in the previous subsections. The preconditioners from Section 5.5.1 lead to fast convergence due to a very good coarse grid representation of the zero curve of f , but the number of grids is limited. The preconditioners based on auxiliary problems, on the other hand, use an arbitrary number of grids, but require a larger number of auxiliary problems if matrices are very large. Therefore, we use a similar heuristic as in Section 5.3.3 for the multigrid solver: *Start with the first strategy until zero curves become too large, then approximate the zero curve on the coarsest level with k auxiliary problems and continue with the second strategy.* Figure 5.9 illustrates this method with the function f from Example 6. It depicts the zero curves of $f(x, y)$ and its counterparts on the next two coarser levels $f_2(x, y) = 0$ and $f_3(x, y) = 0$. Since $f_3(x, y) = 0$ is too large for a further coarsening step, we split it into 8 auxiliary problems $h_{3,j}(x, y)$. The curves $h_{3,j}(x, y) = 0.001$ show how these auxiliary problems behave in the neighborhood

ρ	$\mathbf{n}=(2^5-1)^2$	$\mathbf{n}=(2^6-1)^2$	$\mathbf{n}=(2^7-1)^2$
1.9	30	30	30
1.8	32	33	33
1.6	40	42	42

Table 5.9: Iteration numbers for the pcg method with the rediscretization-based two-grid preconditioner applied to $\tau_{\mathbf{n}}[f]$ with f from Example 6

of their zero.

This heuristic is translated to matrices as follows. The first two coarsening steps, i.e. the ones following the first strategy, are performed with the restriction matrices corresponding to (5.1.1), (5.1.2), and (5.1.3) and with the coarse grid matrices A_2 and A_3 from (5.2.3) or (5.2.4). To simplify notation, let us only use one coarsening step within each auxiliary problem. A_3 is then approximated by k matrices $A_{3,j}$ corresponding to the k auxiliary functions $h_{3,j}$ from (5.5.8). Their equivalents on the next coarser grid are computed by

$$A_{4,j} = R_{3,j} A_{3,j} R_{3,j}^H$$

with $R_{3,j}$ defined as in Chapter 3. Then, the whole preconditioner P can be written

$$P = \text{diag}(A)^{-1} + R_1^H (\text{diag}(A_2)^{-1} + R_2^H (c \cdot I + \sum_{j=1}^k R_{3,j}^H A_{4,j}^{-1} R_{3,j}) R_2) R_1 \quad . \quad (5.5.9)$$

In a practical algorithm, the matrices $A_{4,j}$ are again recursively approximated by further levels of restriction and computation of coarse grid matrices.

5.5.5 Numerical results

In the following, we wish to give numerical evidence of the optimal convergence obtained with the conjugate gradient method and a diagonal scaling preconditioner. We start with the two-grid method which is based on rediscretization. The coarse-grid function is chosen to be f_2 from (5.2.3), approximating the exact zero curve in 8 points. For two-level tau matrices corresponding to f from Example 6, we obtain very similar iteration numbers as in Table 5.8, although the zero curve is only approximated. The results are displayed in Table 5.9. The real advantage of this rediscretization approach is that matrices do not become denser on coarser grids. Hence, we apply a four-grid method which is not only optimal concerning convergence, but also efficient concerning computational cost. The only restriction is posed by the increasing size of the zero curves on coarser grids. This time, we use two-level Toeplitz instead of tau matrices. Table 5.10 summarizes the iteration numbers of the pcg method. The preconditioner based on splitting leads to

ρ	$\mathbf{n}=(2^6-1)^2$	$\mathbf{n}=(2^7-1)^2$	$\mathbf{n}=(2^8-1)^2$
1.99	33	33	33
1.98	36	36	36
1.97	41	41	41

Table 5.10: Iteration numbers for the pcg method with the rediscrctization-based four-grid preconditioner applied to $T_{\mathbf{n}}[f]$ with f from Example 6

ρ	\sharp (points)	$\mathbf{n}=(2^6-1)^2$	$\mathbf{n}=(2^7-1)^2$	$\mathbf{n}=(2^8-1)^2$
1.9	4	23	23	23
1.8	8	24	24	24
1.6	8	30	31	31

Table 5.11: Iteration numbers for the pcg method with the splitting-based four-grid preconditioner applied to $T_{\mathbf{n}}[f]$ with f from Example 6

fast convergence, too. Depending on the parameter ρ , we choose 2 or 3 auxiliary problems, approximating the zero curve in 4 or 8 points. Again, the number of iterations is considerable larger than in Section 5.3, because here we perform only very basic smoothing. The results of the four-grid method are displayed in Table 5.11.

Chapter 6

Applications

We have chosen two applications, out of many, for the solution of structured linear systems with multigrid methods. The first application comes from the field of image restoration. More precisely, we solve the deblurring problem with multigrid methods, under the assumption that the shift-invariant point spread function is anisotropic. The second application is the solution of partial differential equations. We focus on anisotropic PDEs of Poisson type and on indefinite PDEs of Helmholtz type. The structured matrix classes presented in this thesis arise both from discretization of constant coefficient PDEs and from the use of preconditioners for PDEs with variable coefficients.

6.1 Anisotropic image restoration problems

Deblurring problems in image restoration are a typical application for the structured linear systems presented in this work. The aim is to reconstruct an image which has been blurred and, in most realistic examples, affected by noise. In this work, we assume that the blur is spatially invariant, i.e. the blur does not depend on the position within the image. Moreover, we assume that the blurring operator is known and given by the *point spread function* (PSF). This function describes how a point source of light is blurred into a larger object. Since the blur is spatially invariant, application of the PSF to each point of the image leads to a structured linear system of Toeplitz or trigonometric algebra type. For the solution of this linear system during the deblurring process, multigrid methods are highly efficient. In this thesis, we are especially interested in deblurring problems where the PSF is of anisotropic nature.

We start with a description of the linear deblurring problem and a discussion about which types of boundary conditions should be imposed. Our presentation is mainly influenced by [83, 37, 39, 38]. Then, we introduce the anisotropic case, under the slightly idealized assumption that there is no noise. In this case, we can directly apply the multigrid methods from Chapter 4 for the deconvolution

process. We introduce test problems where the blur originates from different point spread functions. In the presence of noise, however, the straightforward deblurring approach does not work anymore, because the linear systems become ill-posed. In other words, the problem is so strongly ill-conditioned that already a small amount of noise is greatly amplified in the restored image. To overcome this problem, we review regularization strategies suggested in [83, 37, 39, 42]. In this work, we wish to discuss two of them in more detail, which are effectively combined with multigrid solvers: the regularization methods of Tikhonov and Riley.

6.1.1 The deblurring problem with boundary conditions

For the introduction of the general deblurring model, we assume that the blur is spatially invariant and that there is no noise. To simplify notation, we start with the 1D case, i.e. with signals, and describe the 2D case, i.e. images, as an extension of the 1D case by tensor arguments. The blurred signal g is obtained as the convolution $g = h \circ \tilde{s}$ of the blurring function

$$h = (\dots, 0, 0, h_{-m}, h_{-m+1}, \dots, h_0, \dots, h_{m-1}, h_m, 0, 0, \dots)^T \quad (6.1.1)$$

and the original signal

$$\tilde{s} = (\dots, s_{-m+1}, \dots, s_0, s_1, \dots, s_n, s_{n+1}, \dots, s_{n+m}, \dots)^T . \quad (6.1.2)$$

In other words, the j -th entry g_j of the blurred signal is computed with

$$g_j = \sum_{k=-\infty}^{\infty} h_{j-k} s_k .$$

The principal aim is to recover, from the given blurred signal

$$g = (g_1, \dots, g_n)^T ,$$

which is measured at n points, the original signal

$$s = (s_1, \dots, s_n)^T$$

at the same n points. Since the blurred signal g is not only determined by s_1, \dots, s_n , but also by s_{-m+1}, \dots, s_{-1} and by s_{n+1}, \dots, s_{n+m} , the deblurring problem is equivalent to the solution of the linear system

$$\left(\begin{array}{cccccccc} h_m & \dots & h_0 & \dots & h_{-m} & & & \\ & h_m & & h_0 & & h_{-m} & & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & & \ddots & \ddots & \ddots & \ddots & \ddots \\ & & & & h_m & h_0 & h_{-m} & \\ & & & & h_m & \dots & h_0 & \dots & h_{-m} \end{array} \right) \begin{pmatrix} s_{-m+1} \\ s_{-m+2} \\ \vdots \\ s_0 \\ \vdots \\ s_n \\ s_{n+1} \\ \vdots \\ s_{n+m} \end{pmatrix} = \begin{pmatrix} g_1 \\ \vdots \\ g_n \end{pmatrix} . \quad (6.1.3)$$

Since this linear system is underdetermined, we make certain assumptions on the values s_{-m+1}, \dots, s_{-1} and s_{n+1}, \dots, s_{n+m} , i.e. we impose boundary conditions. These are chosen for the problem at hand, mainly serving two purposes. On the one hand, adequate boundary conditions are supposed to yield a high precision of the reconstruction process near the boundaries. On the other hand, we obtain square matrices for which a fast solution algorithm is available. Depending on the type of boundary conditions, one obtains matrices of a certain class. In the following, we present the ones which are most important.

- **Zero Dirichlet** boundary conditions yield **Toeplitz** matrices. All values of \tilde{s} outside $1, \dots, n$ are chosen to be zero, i.e.

$$s_{-m+1} = \dots = s_{-1} = 0 \quad \text{and} \quad s_{n+1} = \dots = s_{n+m} = 0 \quad .$$

These boundary conditions have, in some cases, two disadvantages. In general, Toeplitz matrices are more difficult to solve than matrices from a trigonometric algebra. Moreover, Dirichlet boundary conditions can introduce discontinuities near the border, depending on the type of signal or image. These artifacts (often also called ringing effects) spread throughout the image due to the strong ill-conditioning of the blurring matrix.

- **Periodic** boundary conditions result in **circulant** matrices. They are imposed by requiring $f_j = f_{n+j}$ for all j . Periodic boundary conditions can still produce discontinuities at the border, but the linear system can be solved in $O(n \log n)$ arithmetic operations.
- **Neumann** or **reflective** boundary conditions are obtained by reflecting the data at the boundary, i.e. by choosing

$$s_{1-j} = s_j \quad \text{and} \quad s_{n+j} = s_{n+1-j} \quad \text{for all } j \quad .$$

The resulting matrices belong to the **DCT-III** class. Since boundary conditions preserve continuity of the image, they are better in avoiding ringing effects. Moreover, they also lead to linear systems which are solved in $O(n \log n)$ operations.

- **Anti-reflective** boundary conditions are obtained by performing an anti-reflection at the boundary, i.e. by choosing

$$s_{1-j} = 2s_1 - s_{j+1} \quad \text{and} \quad s_{n+j} = 2s_n - s_{n-j} \quad \text{for all } j \quad .$$

The resulting matrix is not exactly a **tau** matrix, but the solution of the linear system can be reduced to the solution of a linear system with tau structure. The use of antireflective boundary conditions for image deblurring was introduced in [101] and further investigated in [41, 40]. In addition to preserving continuity of the image, they also preserve continuity of the normal derivatives.

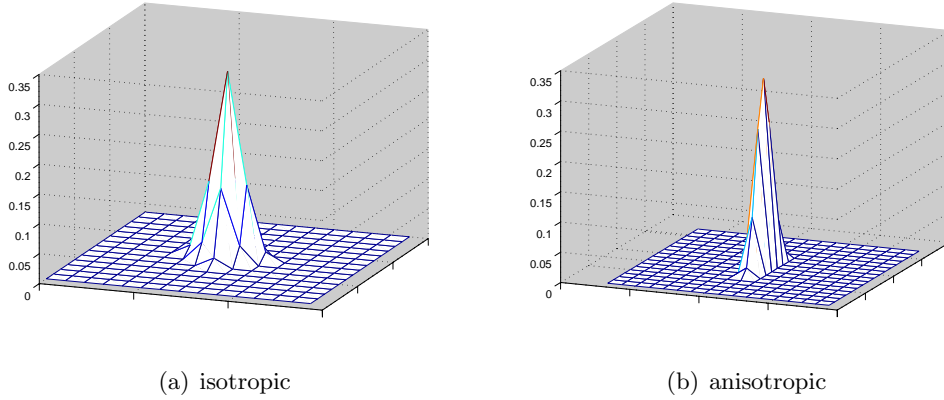


Figure 6.1: PSFs for isotropic and anisotropic blurring

Blurring matrices in the 2D case are obtained from the 1D matrices by tensor arguments. The same types of boundary conditions are applied as in the 1D case in order to treat the values of s outside the image boundaries. The result is a two-level, Toeplitz, circulant, or DCT-III matrix, or, in the case of anti-reflective boundary conditions, a linear system which can be reduced to a two-level tau system.

6.1.2 Multigrid for noise-free anisotropic deblurring

In this section, we assume that the blurred image is not affected by any noise, i.e. the quantity η in (1.1.1) is the zero vector. Hence, deblurring of an image, whose spatially invariant blurring function is known, reduces to solving a linear system of the form

$$A_{\mathbf{n}}x = b \quad , \quad (6.1.4)$$

where $A_{\mathbf{n}}$ is a two-level Toeplitz or trigonometric algebra type matrix. In many examples, the blur is modeled with a Gaussian-like PSF, which usually has a rather small support. It takes a large value at the center and decreases rapidly to zero when moving away from the middle. In [37, 39], the author suggests to approximate the Gaussian filter with compact support by a trigonometric polynomial with a similar behavior. The PSF is obtained by taking the coefficients of a polynomial of the form

$$f(x, y) = (2 + \cos(x) + \cos(y))^3 \cdot \phi(x, y)/c \quad . \quad (6.1.5)$$

In this function, $\phi(x, y)$ is a strictly positive function and c a normalization constant, which guarantees that the sum of the Fourier coefficients of $f(x, y)$ is 1. The function f has a zero of order 6 at (π, π) and is strictly positive elsewhere in $[0, 2\pi]^2$.

In Figure 6.1(a), the PSF corresponding to $f(x, y)$ from (6.1.5) is depicted with $\phi(x, y) = 1$.

In [37, 38], systems of the form (6.1.4) are solved with multigrid methods. As described in Chapter 3, a zero of f at (x_0, y_0) leads to a zero of f_2 at

$$(2x_0 \bmod 2\pi, 2y_0 \bmod 2\pi) .$$

This implies that on all grids except for the finest, the generating function has a zero at the origin. Hence, the same restriction matrix is used on each level, except for the finest. Since the multigrid methods converge optimally for functions such as f from (6.1.5), the total cost of the solution is $O(n \log n)$ operations in the case of a dense blurring function and $O(n)$ in the banded case. Therefore, multigrid methods are faster than the best direct methods if the PSF has small support.

In this work, we are interested in *anisotropic* blurring operators. This means blurring occurs mainly in one direction, which can be one of the axes or any other direction. We assume that the direction of anisotropy is the same across the whole picture. The following two point spread functions are used to illustrate this kind of blur.

Example 8 We wish to examine the blur obtained from the PSF which corresponds to the generating functions

$$f(x, y) = [(1 + \cos(x)) + \alpha \cdot (1 + \cos(y))]^2 \cdot \phi(x, y) / c . \quad (6.1.6)$$

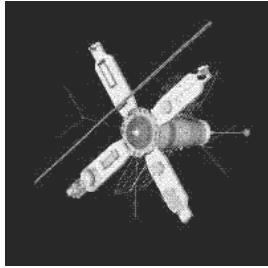
$$g(x, y) = [(1 + \cos(x + y))^2 + \alpha \cdot (1 + \cos(x - y))^2] \cdot \phi(x, y) / c . \quad (6.1.7)$$

In both cases, $\phi(x, y)$ denotes a strictly positive trigonometric polynomial, whereas c is a normalization constant and $0 \leq \alpha \ll 1$. When f is used, anisotropy occurs along the y -axis, i.e. the blurring happens mainly in x -direction. In the case of g , anisotropy and blurring occur in an angle of 45° with respect to the coordinate axes. In the limit case $\alpha = 0$, blurring occurs only in one direction. Figure 6.1(b) shows the anisotropic PSF corresponding to $f(x, y)$ from (6.1.6) with $\phi(x, y) = 1$ and $\alpha = 0.005$.

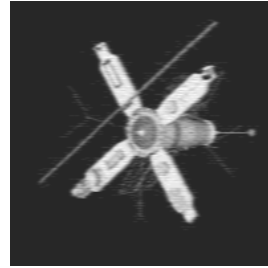
In Chapter 4, we have developed and analyzed multigrid methods for anisotropic linear systems of both kinds. Systems with anisotropies along coordinate axes are more straightforward to solve, but the methods from Sections 4.4 and 4.5 allow an efficient treatment of anisotropies in other directions, especially in the two-level circulant case, i.e. in the case of periodic boundary conditions.

6.1.3 Test problems and numerical results

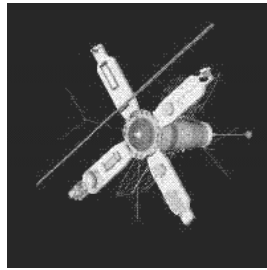
In this section, we carry out numerical tests to demonstrate the efficiency of multigrid methods for the deblurring process with anisotropic point spread functions. In order to focus on the efficiency of multigrid methods as solvers, we still assume that there is no noise. The first example picture for our tests is the well-know satellite image, which has been used in many articles such as [37, 39].



(a) Original image



(b) Blurred image



(c) Restored image

Figure 6.2: Application of the anisotropic PSF corresponding to f from Example 8 to the satellite image (size 200×200)

Figure 6.2(a) shows the original picture which is of size 200×200 . Since it is mostly black near the boundaries, all types of boundary conditions are more or less equivalent. Therefore, we can safely assume that periodic boundary conditions lead to a very low amount of ringing effects. In Figure 6.2(b), the picture has been blurred with the anisotropic PSF $f(x, y)$ from Example 8. The parameter α , which indicates the degree of anisotropy is set to 0.001, i.e. the PSF is strongly anisotropic. The positive function ϕ is chosen

$$\phi(x, y) = [(2 + \cos(x)) + \alpha \cdot (1 + \cos(x))]^2 ,$$

which also contributes to the strong anisotropy. This PSF implies that we construct a multigrid method which uses, in each V-cycle, 5 semicoarsening steps, followed by full coarsening. Since the zero f has order 4 and is located at (π, π) , semicoarsening on the top level is done with

$$b(x, y) = (1 - \cos(x))^2 , \quad (6.1.8)$$

whereas on all other grids where semicoarsening is applied, we choose

$$b_j(x, y) = (1 + \cos(x))^2 . \quad (6.1.9)$$

For the first full coarsening step on the 6-th finest grid, one must keep in mind that the y -coordinate of the zero of f_6 is still π . This implies that a function such as

$$b_6(x, y) = (1 + \cos(x))^2 \cdot (1 - \cos(y))^2 . \quad (6.1.10)$$

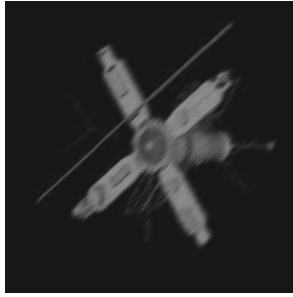
must be applied. If one step of the symmetric Gauss-Seidel is used as a smoother and as a postsmoother on each level, the multigrid method converges after 10 V-cycle iterations with a residual less than $1 \cdot 10^{-6}$. The result is shown in Figure 6.2(c). If the damped Jacobi method is used for smoothing, the number of iterations is slightly higher, but the multigrid method nevertheless converges optimally.

If reflective boundary conditions are used instead of periodic boundary conditions, the results are very similar. The PSF used in the example above produces the same picture as in Figure 6.2(b), and the same multigrid method as above is defined for the DCT-III algebra. Again, rapid convergence is obtained in this noise-free case.

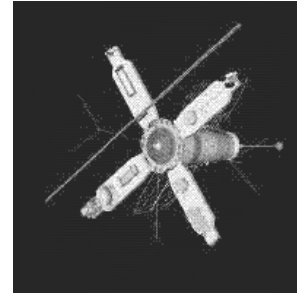
The second example PSF, $g(x, y)$ from Example 8, is anisotropic along the line $y = x$ and has zeros at $(0, \pi)$ and $(\pi, 0)$. Hence, blurring is done in an angle of 45° with respect to the coordinate axes. The positive function ϕ , which makes the blurring stronger, is chosen

$$\phi(x, y) = [(2 + \cos(x + y)) + \alpha \cdot (1 + \cos(x - y))]^2 ,$$

and α is 0.01.



(a) Blurred image



(b) Restored image

Figure 6.3: Application of the anisotropic PSF corresponding to g from Example 8 to the satellite image (size 200×200)

Figure 6.3(a) shows the result of this kind of blurring applied to the same satellite image as above. The multigrid method we apply for the restoration of the original image is of the same form as the methods in Section 4.4. First, we define new coordinates as suggested in (4.4.17) and then, we apply, in each V-cycle iteration, three semicoarsening steps followed by full coarsening steps. The only zero of g in the new coordinates is located at (π, π) . The generating functions $b(s, t)$ and $b_j(s, t)$ are of the same form as the functions in (6.1.8) and (6.1.9), with x and y being replaced by s and t . The result of the multigrid method, i.e. the restored picture, is shown in Figure 6.3(b). Again, the multigrid method converges after a low number of V-cycle iterations. If the symmetric Gauss-Seidel smoother is used, the number is 13. If the damped Jacobi method is applied, a few more iterations are necessary.

6.1.4 Regularization strategies in the presence of noise

So far, all computations were done under the assumption that the blurred images are not contaminated by noise. However, in most realistic examples, blurred images contain at least a small amount of noise. In [37, 39], the author describes that already 2% of noise cause the multigrid method to break down completely. The small eigenvalues of A_n amplify the noise and corrupt the image already after one V-cycle. In other words, the multigrid methods from Chapter 4 do not have regularization properties. In [83, 60], it is illustrated that even direct solvers such as the FFT do not yield satisfactory results when applied to a blurred image with noise. Since these methods fail for the same reason as our multigrid methods, other regularization strategies have to be employed. The main goal is to make the

unstable problem as stable as possible. On the other hand, we should not sacrifice too much of the accuracy of the computed solution.

In the following, we give a brief overview of some of the most popular regularization methods.

- The *Tikhonov regularization* (see [110]) is one of the most classical regularization techniques. The task of solving the ill-conditioned linear system (6.1.4) is replaced by the problem of minimizing

$$\|A_{\mathbf{n}}x - b\|_2^2 + \mu\|x\|_2^2 \quad (6.1.11)$$

over $x \in \mathbb{R}^{\mathbf{n}}$ with a fixed parameter $\mu \geq 0$. The minimum is obtained via normal equations, i.e. by solving the linear system

$$(A_{\mathbf{n}}^T A_{\mathbf{n}} + \mu I)x = A_{\mathbf{n}}b \quad , \quad (6.1.12)$$

where b on the right-hand side represents the blurred and noisy image. The parameter μ must be chosen such that the problem is regularized, but not too much of the original information is lost. In 6.1.12, the matrix I can be replaced by a low order differential operator $T^T T$, as suggested in [42]. The Tikhonov regularization has the disadvantage that the condition number of the system matrix is doubled when we change from $A_{\mathbf{n}}$ to $A_{\mathbf{n}}^T A_{\mathbf{n}}$. Moreover, in the Toeplitz case, $A_{\mathbf{n}}^T A_{\mathbf{n}}$ is not exactly a BTTB matrix.

- If $A_{\mathbf{n}}$ is symmetric and positive definite, *Riley* proposed a different regularization technique in [89]. Instead of resorting to normal equations, he chooses a parameter $\nu \geq 0$ and solves the linear system

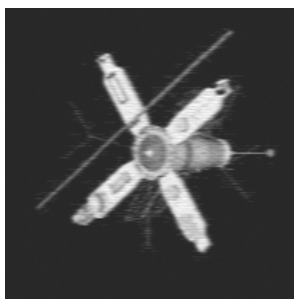
$$(A_{\mathbf{n}} + \nu I)x = b \quad . \quad (6.1.13)$$

This is equivalent to minimizing

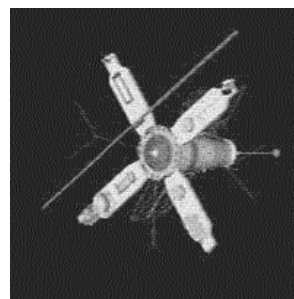
$$\|A_{\mathbf{n}}^{1/2}x - A_{\mathbf{n}}^{-1/2}b\|_2^2 + \nu\|x\|_2^2 \quad (6.1.14)$$

over all $x \in \mathbb{R}^{\mathbf{n}}$. This method circumvents the two problems of the Tikhonov regularization, but the matrix $A_{\mathbf{n}}^{-1/2}$ slightly amplifies the noise.

- *Iterative methods* such as the classical Landweber method [75] or its projected version are often used for regularization purposes. They are simple gradient descent algorithms which are often applied to normal equations. For positive definite linear systems, the cg method can be used instead and is usually more efficient. In [42], the authors propose a combination of geometric and algebraic multigrid as a regularizer.



(a) Blurred image with 2% noise



(b) Restored image

Figure 6.4: Deblurring with the anisotropic PSF corresponding to $f(x, y)$ from Example 8 in the presence of noise

6.1.5 Numerical tests with Tikhonov and Riley regularization

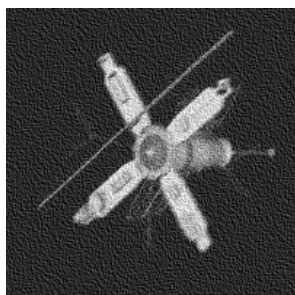
For the solution of the anisotropic deblurring problem, we concentrate on the classical regularization strategies of Riley and Tikhonov. The resulting linear systems are solved with the multigrid methods for anisotropic linear systems from Chapter 4. Let us start with the case of periodic boundary conditions, i.e. two-level circulant matrices. For our first example, we choose the PSF corresponding to $f(x, y)$ from (6.1.6). As was done in [37, 39], we add 2% of noise to the blurred image. The resulting picture is displayed in Figure 6.4(a).

As described in Section 6.1.4, deblurring without regularization amplifies the noise and returns a picture which is absolutely useless. Therefore, we apply Riley's regularization method and solve the linear system (6.1.13). The optimal parameter ν in this linear system is 0.08. The multigrid method from Section 6.1.3 applied to the regularized system (6.1.13) converges after 6 iterations. The resulting picture is shown in Figure 6.4(b). The Tikhonov regularization leads to a restored picture which cannot be distinguished from the one in Figure 6.4(b). As reported in [37, 39], the number of multigrid iterations, which is 25, is significantly higher compared to the Riley method, and the optimal regularization parameter is $\mu = 0.01$.

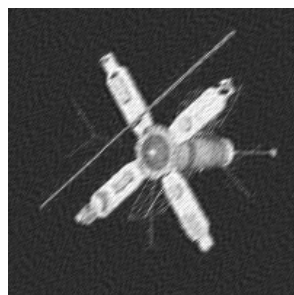
In the second example, the blurred image is affected by a larger amount of noise. We choose the PSF corresponding to $g(x, y)$ from (6.1.7). In this case, the blurred image is affected by 10% of noise. The resulting picture is displayed in Figure 6.5(a).



(a) Blurred image with 10 % noise



(b) Restored image with Riley regularization



(c) Restored image with Tikhonov regularization

Figure 6.5: Deblurring with the anisotropic PSF corresponding to $g(x, y)$ from Example 8 in the presence of noise

We perform regularization with both the Riley and the Tikhonov method, using the same parameters as above. Then, we solve the resulting linear systems with the same multigrid method as in Section 6.1.3. For the system obtained with the Riley method, the multigrid method converges after 8 iterations, whereas for the system obtained with the Tikhonov method, 38 iterations are needed. The resulting pictures are shown in Figures 6.5(b) and 6.5(b). The computational effort of the Tikhonov regularization is significantly higher, but the extra work pays off in this example with a larger amount of noise. The restoration with Tikhonov's method leads to a restored image which is less contaminated with noise.

6.2 Partial differential equations

The second type of applications presented in this thesis is the numerical solution of partial differential equations. Discretization of a PDE with finite differences or finite elements leads to a linear system which is sparse and, in most cases, banded. This section is divided into two parts. In the first subsection, the focus is on anisotropic PDEs of Poisson type. Hence, the multigrid methods from Chapter 4 are applied. The main topic of the second part is the application of the methods from Chapter 5 to the Helmholtz equation.

6.2.1 Multigrid methods for anisotropic PDEs

The idea of using multigrid methods for the solution of anisotropic PDEs is not new. In this work, however, we look at these problems from a slightly different perspective. The importance of structured linear systems corresponding to generating functions for the solution of isotropic elliptic PDEs has been described, for example, in the book of Ng [83] and in the articles [98, 103]. Here, we focus entirely on anisotropic PDEs, putting special emphasis on the solution of the resulting linear systems with multigrid methods. Central to our approach are those classes of structured matrices for which we have developed multigrid methods in Chapter 4.

We discuss elliptic problems of the form

$$-(a_1(x, y)u_x)_x - (a_2(x, y)u_y)_y = h(x, y) \quad , \quad (6.2.1)$$

defined on the unit square ($\Omega = [0, 1]^2$) with nonnegative functions $a_1(x, y)$ and $a_2(x, y)$. Linear systems of Toeplitz or trigonometric algebra type mainly appear in two situations:

- When $a_1(x, y)$ and $a_2(x, y)$ are constant functions, i.e. when the PDE has constant coefficients, finite-difference discretization, for example with a five-point stencil, leads to a Toeplitz, circulant, or DCT-III matrix, depending on the type of boundary condition.

- In the case of variable coefficients $a_1(x, y)$ and $a_2(x, y)$, circulant or tau preconditioners are applied to reduce the condition number of the linear system.

In the remainder of this subsection, we discuss these two cases and apply multigrid methods for the structured linear systems involved.

Anisotropic PDEs with constant coefficients

An elliptic PDE of the form (6.2.1) with constant coefficients $a_1 = a_1(x, y)$ and $a_2 = a_2(x, y)$ is said to be anisotropic if either $0 < a_1 \ll a_2$ or $0 < a_2 \ll a_1$. First, we assume that, on the whole boundary Ω , homogeneous Dirichlet boundary conditions are imposed. In this case, discretization of equation (6.2.1) with a five-point stencil or a nine-point stencil on a uniform grid results in a linear system $A_{\mathbf{n}}x = g$, which belongs to both the two-level Toeplitz and the two-level tau class. In the case of periodic boundary conditions, the resulting matrix $A_{\mathbf{n}}$ is circulant, and in the case of Neumann boundary conditions, $A_{\mathbf{n}}$ is of DCT-III type. Mixed types of structured matrices appear when different types of boundary conditions are imposed on different parts of the boundary. The corresponding generating function to (6.2.1) with constant coefficients is

$$f(x, y) = a_1(1 - \cos(x)) + a_2(1 - \cos(y)) . \quad (6.2.2)$$

For the solution of the linear system with multigrid methods, we use the same heuristic as in Section 4.2. We apply semicoarsening until the system is not anisotropic anymore and then switch to full coarsening. Theorem 12 states that the choice

$$b(x, y) = 1 + \cos(x) \text{ or } b(x, y) = 1 + \cos(y) \quad (6.2.3)$$

(depending on the direction of anisotropy) is sufficiently accurate for the construction of the restriction matrices on all grids. For different values of a_1 and a_2 , our heuristic suggests different numbers of semicoarsening steps. Table 6.1 contains the optimal number of semicoarsening steps and the iteration numbers of the V-cycle for different values of a_1 and a_2 . Since the boundary conditions are assumed to be of Dirichlet type, the resulting matrices have both Toeplitz and tau structure. We use one iteration of the Gauss-Seidel method as pre- and postsmoother.

Discretization of the equation

$$-(a_1(x, y)u_{xx})_{xx} - (a_2(x, y)u_{yy})_{yy} = h(x, y) , \quad (6.2.4)$$

with constant coefficients $a_1 = a_1(x, y)$ and $a_2 = a_2(x, y)$ results in a function of the form

$$f(x, y) = a_1(1 - \cos(x))^2 + a_2(1 - \cos(y))^2 . \quad (6.2.5)$$

The function $b(x, y)$ is chosen to be the square of the function in (6.2.3). Table 6.2 contains the iteration numbers of the six-grid method for different choices of a_1 and a_2 and for matrices $R_{\mathbf{n}}[f]$ of the DCT-III algebra.

	# semi	$\mathbf{n}=(2^6-1)^2$	$\mathbf{n}=(2^7-1)^2$	$\mathbf{n}=(2^8-1)^2$
$a_1 = 0.1, a_2 = 1$	2	10	10	10
$a_1 = 0.01, a_2 = 1$	3	8	8	8
$a_1 = 0.001, a_2 = 1$	5	7	7	7

Table 6.1: Iteration numbers of the six-grid method for $\tau_{\mathbf{n}}[f]$ with $f(x, y)$ from (6.2.2) and different a_1 and a_2

	# semi	$\mathbf{n}=(2^6-1)^2$	$\mathbf{n}=(2^7-1)^2$	$\mathbf{n}=(2^8-1)^2$
$a_1 = 0.1, a_2 = 1$	2	9	9	9
$a_1 = 0.01, a_2 = 1$	3	10	10	10
$a_1 = 0.001, a_2 = 1$	5	10	10	10

Table 6.2: Iteration numbers of the six-grid method for $R_{\mathbf{n}}[f]$ with $f(x, y)$ from (6.2.5) and different a_1 and a_2

Preconditioners from trigonometric matrix algebras

In the following, we assume that the coefficients $a_1(x, y)$ and $a_2(x, y)$ in (6.2.1) and (6.2.4) are nonnegative, but variable in x and y . Then, the resulting matrices $A_{\mathbf{n}}$ have the same sparsity pattern as above, but they do not belong to a structured matrix class that corresponds to generating functions. In the point (x_0, y_0) , the PDE is anisotropic if $a_2(x_0, y_0)$ is significantly smaller or larger than $a_1(x_0, y_0)$.

Circulant preconditioner have been employed for isotropic PDEs of the form (6.2.1) or (6.2.4), see e.g. [83, 22, 103]. Two different ways to construct a circulant preconditioner are proposed in [22]. Both are based on computing the arithmetic means \bar{a}_1 and \bar{a}_2 of all values of $a_1(x, y)$ and $a_2(x, y)$ at the grid points. The first preconditioner C_1 is a one-level circulant matrix of size \mathbf{n} with entries

$$c_1^{(1)} = 2(\bar{a}_1 + \bar{a}_2) + \rho \mathbf{n}^{-1}, \quad c_2^{(1)} = -\bar{a}_1, \quad c_{n_2+1}^{(1)} = -\bar{a}_2,$$

whereas the second preconditioner is a two-level circulant matrix of the form

$$C_2 = C_{a_2} \otimes I + I \otimes C_{a_1}$$

with entries

$$c_1^{(a_1)} = 2\bar{a}_1 + \rho \mathbf{n}^{-1}, \quad c_2^{(a_1)} = -\bar{a}_1, \quad c_1^{(a_2)} = 2\bar{a}_2 + \rho \mathbf{n}^{-1}, \quad c_2^{(a_2)} = -\bar{a}_2, \dots$$

The terms $\rho \mathbf{n}^{-1}$ are added to the main diagonal of the preconditioners in order to minimize the condition number of the resulting preconditioned system. They correspond to the Strang correction for circulant matrices, which was introduced in Section 4.2.5. In [22], it is proved that C_1 reduces the condition of $A_{\mathbf{n}}$ from

	# semi	$\mathbf{n}=(2^6-1)^2$	$\mathbf{n}=(2^7-1)^2$	$\mathbf{n}=(2^8-1)^2$
$a_1 = 0.1, a_2 = 1$	2	7	7	7
$a_1 = 0.01, a_2 = 1$	3	6	6	6
$a_1 = 0.001, a_2 = 1$	5	6	6	6

Table 6.3: Iteration numbers of the six-grid method for $C_{\mathbf{n}}[f]$ with $f(x, y)$ from (6.2.5) and different \bar{a}_1 and \bar{a}_2

$O(\mathbf{n})$ to $O(\sqrt{\mathbf{n}} \log(\mathbf{n}))$ and C_2 to $O(\sqrt{\mathbf{n}})$. Here, we focus on C_2 , and particularly on the case where C_2 is anisotropic, i.e. where $\bar{a}_1 \ll \bar{a}_2$ or $\bar{a}_2 \ll \bar{a}_1$. The matrix C_2 is equal to the Strang correction of $C_{\mathbf{n}}[f]$ corresponding to the function $f(x, y)$ from (6.2.2). Discretization of higher order PDEs may result in functions such as f from (6.2.5) or in functions with zeros of higher order. These circulant linear systems can be solved with the methods from Chapter 4. Table 6.3 contains the number of V-cycle iteration for f from (6.2.5). Six grids are used, and the number of semicoarsening steps depends on the values of \bar{a}_1 and \bar{a}_2 .

In [69], Huckle suggested a skew-circulant preconditioner, which leads to similar results. The advantage of this approach is that the preconditioner works without the extra term $\rho \mathbf{n}^{-1}$. For anisotropic problems, this preconditioner can be applied as well, showing a similar convergence behavior as the circulant one.

Not only circulant and skew-circulant matrices are used as preconditioners for linear systems obtained from discretizing elliptic PDEs. Especially when Dirichlet boundary conditions are imposed on the PDE, the tau class is a more natural choice for the preconditioner. Different tau preconditioners have been suggested in the literature, see e.g. [30, 83]. For numerical results on two-level tau systems, which can be used as preconditioners similar to the circulant examples, we refer to Table 6.1.

6.2.2 Multigrid methods for the Helmholtz equation

Several articles have been published concerning the solution of the Helmholtz equation with multigrid methods, see for example [16, 47, 50, 49]. In this article, we wish to present a different approach to the multigrid solution of the Helmholtz equation with constant coefficients. It is primarily based on certain classes of structured matrices and their strong correspondence to generating functions.

The discretized Helmholtz equation

The 2D Helmholtz equation is a boundary value problem given by

$$\begin{aligned} -\Delta u - k^2 u &= g & \text{on } \Omega \subset \mathbb{R}^2, \\ Bu &= h & \text{on } \Gamma \subset \delta\Omega, \end{aligned} \tag{6.2.6}$$

h	$k = 50$	$k = 100$	$k = 200$	$k = 400$
0.01	1.875	1.500	0	–
0.001	1.999	1.995	1.980	1.920
0.0001	2.000	2.000	1.999	1.999

Table 6.4: Values of ρ , depending on k and h

where k is a constant wavenumber and B an operator on the boundary Γ . Discretization of the Helmholtz equation is done, for example, with a five-point finite difference scheme. Depending on the boundary condition, this results in a sparse and structured matrix A of a certain class. For example, if Dirichlet boundary conditions are imposed on Γ , the matrix belongs to both the two-level Toeplitz and the two-level tau class. For periodic boundary conditions, we obtain a two-level circulant matrix, and for certain Neumann boundary conditions, a two-level DCT-III matrix. Since all these matrix classes are strongly related to generating functions, we wish to apply the theory of generating functions to the solution of the discretized Helmholtz equation.

Since the matrix A is indefinite, we solve the corresponding linear system via normal equations. The generating function f , which corresponds to the matrix $A_{\mathbf{n}}[f] = A^T A$, is of the form (5.0.1), where ρ takes the value $2 - \frac{k^2 h^2}{2}$, with k denoting the wavenumber and h the size of a discretization step (see [50, 49]). If $|\rho| = 2$, $f(x, y)$ has a single isolated zero of order 4, whereas for $|\rho| < 2$, f is zero along a whole curve. This zero curve becomes larger as ρ decreases. Depending on the boundary condition, $A_{\mathbf{n}}[f]$ is the two-level tau, circulant, or DCT-III matrix corresponding to $f(x, y)$.

For these types of boundary conditions and under the assumption of constant coefficients, the methods from Chapter 5 can be applied for the multigrid solution of the discretized systems. In our notation, the parameter ρ measures how indefinite and how difficult-to-solve the linear system is. Usually, the degree of indefiniteness is determined with k in (6.2.6). In many articles (see e.g. [47, 50]), the condition

$$k \cdot h \leq \pi/5 \tag{6.2.7}$$

is proposed for determining the grid size, i.e. the size of the resulting matrix. If $k \cdot h = \pi/5$, ρ takes the value 1.80. This allows between one and two coarsening steps with the method from Section 5.2. Then, the zero curve is too large for a further step, and we have to switch to a splitting technique on coarser grids. If we require that $k \cdot h$ be smaller, then we end up with a larger ρ , which allows more coarsening steps with one coarse grid correction, but the matrix size increases quadratically. For example, halving the value of $k \cdot h$ implies that the size of the resulting linear system is multiplied by four. Table 6.4 displays the values of ρ obtained for different k and h . For a moderate size of k , such as $k = 30$,

ρ	$k = 10$	$k = 20$	$k = 40$
1.99	14	14	14
1.98	15	15	15
1.97	18	18	18

Table 6.5: Iteration numbers of the four-grid method for $R_{\mathbf{n}}[f]$ with $f(x, y)$ from (5.0.1) and different values of k

ρ	$k = 40$	$k = 80$	$k = 120$
1.8	10	10	10

Table 6.6: Iteration numbers of the four-grid method for $C_{\mathbf{n}}[f]$ with $f(x, y)$ from (5.0.1) and different values of k

condition (6.2.7) is satisfied if $\mathbf{n} > 2300$. If $k = 100$, \mathbf{n} must be > 25000 , and for a large k such as 400, we obtain $\mathbf{n} > 400000$.

In the following, we demonstrate numerically that the multigrid methods from Chapter 5 converge optimally if the approximations for the zero curve are accurate enough. In the notation of the Helmholtz equation this means that for a fixed ρ , the number of iterations does not grow when the wavenumber k increases. The first example deals with moderate values of k . Assuming that ρ is chosen between 1.97 and 1.99, we solve the Helmholtz equation for k between 10 and 50. A larger value of k implies that the size of the linear system grows quadratically if ρ is fixed. Recall from Chapter 5 that a value of ρ between 1.97 and 1.99 allows the use of three coarsening steps with a single coarse grid correction. Only if we wish to use more than four grids, we have to apply splitting. Table 6.5 displays the number of iterations of the four-grid method applied to a DCT-III matrix. If we wish to obtain a larger value of ρ , we have to choose a smaller grid size, i.e. we have to use a larger linear system. On the other hand, the number of iterations is smaller, and we could even use more than four grids if $\rho = 1.99$. For larger k , we assume that the matrix size is chosen such that condition (6.2.7) is just satisfied. We solve the Helmholtz equation for values of k between 40 and 120. With $\rho = 1.8$ we can use only one coarsening step before we have to apply splitting. Table 6.6 contains the number of iterations which are obtained for a multigrid method with one coarsening step followed by splitting and two more coarsening steps for each subproblem. Here, we assume that periodic boundary conditions are imposed.

ρ	$k = 10$	$k = 20$	$k = 40$
1.99	12	12	12
1.98	14	14	14
1.97	15	15	15

Table 6.7: Iteration numbers of the four-grid method for $C_{\mathbf{n}}[f]$ with $f(x, y)$ from (5.4.1) and different values of ρ and k

An anisotropic version of the Helmholtz equation

Finally, let us consider an anisotropic version of the Helmholtz equation. This PDE is obtained by replacing the first line of (6.2.6) by

$$-a_1(x, y) \cdot u_{xx} - a_2(x, y) \cdot u_{yy} - k^2 u = g \quad \text{on } \Omega \subset \mathbb{R}^2 .$$

Here, we assume that the coefficients are constant and strictly positive in Ω , i.e. $a_1 = a_1(x, y) > 0$ and $a_2 = a_2(x, y) > 0$. Moreover, we require that either $a_1 \ll a_2$ or $a_2 \ll a_1$. Discretization with a five-point stencil leads to a structured matrix $A_{\mathbf{n}}[f]$ corresponding to a function $f(x, y)$ which is of the form (5.4.1). The constant coefficient case occurs, for example, when the discretized Helmholtz equation with variable coefficients is preconditioned, for example with a two-level circulant matrix. For the solution of the linear system with $A_{\mathbf{n}}[f]$, we use the multigrid methods from Section 5.4. Depending on the value of ρ , we start with some semicoarsening steps followed either by full coarsening with one coarse grid correction or by splitting into several problems on coarser grids. Again, we wish to demonstrate that the optimal convergence behavior of our multigrid method allows us to solve larger systems, i.e. problems with a larger wavenumber k , in the same number of iterations. We start with k of moderate size, i.e. between 10 and 50, and we assume that the anisotropy is significant, i.e. that $\alpha = 0.01$. If we use quite a large value of ρ , we can apply three steps of semicoarsening and then, possibly, splitting, followed by further coarsening. In our numerical tests, we apply a four-grid method with three semicoarsening steps to the BCCB systems corresponding to f from (5.4.1). Table 6.7 contains the iteration numbers for different choices of ρ and different wavenumbers k . For larger values of k , we require that condition (6.2.7) is just satisfied, i.e. that $\rho \approx 1.8$. This is necessary, because otherwise the resulting linear systems are too large for practical computations. This implies that we can apply one semicoarsening step before we have to split into several coarse grid problems. For our numerical tests, we use a four-grid method, with one step of semicoarsening followed by splitting and two more coarsening steps within each subproblem. Again, we assume that $\alpha = 0.01$. Table 6.8 summarizes the iteration numbers for this method, applied to Toeplitz matrices corresponding to f from (5.4.1) with different values of k .

ρ	$k = 40$	$k = 80$	$k = 120$
1.8	12	12	12

Table 6.8: Iteration numbers of the four-grid method for $T_{\mathbf{n}}[f]$ with $f(x, y)$ from (5.0.1) and different values of k

Chapter 7

Conclusions and future work

At the end of this thesis, we would like to summarize the most important results and point to some possible directions for future research. The whole work was devoted to the development of multigrid methods for structured linear systems of equations. The main contributions to current research presented in this dissertation can be divided into three categories: anisotropic systems (Chapter 4), generating functions with zero curves (Chapter 5), and applications (Chapter 6). Moreover, a multigrid method for the DST-III algebra and some minor theoretical results and heuristics on block Toeplitz matrices and on generating functions with multiple zeros have been obtained in Chapter 3.

Anisotropic systems

With the help of generating functions, we have divided anisotropic structured linear systems into two classes, depending on the direction of anisotropy. The following results were obtained:

- For systems with anisotropy along coordinate axes, we have presented a multigrid method based on a suitable combination of semicoarsening and full coarsening steps. For both Toeplitz matrices and matrices belonging to a trigonometric algebra, we have given two-grid convergence proofs. Moreover, we have shown level-independency, which implies convergence of the multigrid method with W -cycles.
- An alternative approach is the combination of full coarsening with line smoothers. Using generating functions, we have proved that the block Jacobi smoother satisfies conditions (3.1.16) and (3.1.17) in the trigonometric algebra case.
- The rigorous treatment of systems with anisotropy along coordinate axes with the help of generating functions allows us to carry over large parts of the results, for Toeplitz and circulant matrices, to the case of anisotropy in other

directions. After having presented the principal heuristic of transforming the coordinate system, we have given a two-grid convergence proof for BTTB systems. For line smoothers, we have shown that the smoothing conditions still hold in the circulant case.

The following aspects seem to be interesting for further investigations in the future:

- For linear systems with anisotropy along coordinate axes, the multigrid method with V-cycles converges optimally in all numerical experiments. Therefore, a multigrid (V-cycle) convergence proof for trigonometric matrix algebras should be possible by extending the results from [3, 2].
- In the circulant case, a multigrid extension of the convergence proofs (both W-cycle and V-cycle) might be given for the case of anisotropy in other directions. At least, this should be possible when anisotropy occurs in an angle of 45° .

Generating functions with zero curves

The development of multigrid methods for linear systems corresponding to generating functions with zero curves has lead to the following results:

- We have presented a Galerkin-based multigrid method and proved optimal convergence of the two-grid method for all trigonometric matrix algebras.
- By approximating the zero curves on coarser grids, we have modified this Galerkin method such that it still converges extremely fast, but also retains the sparsity of the matrices on coarser grids.
- For the case where zero curves become too large on coarser grids, we have introduced a multigrid method based on splitting. Furthermore, we have combined this method with the Galerkin approach.
- We have combined the results of Chapters 4 and 5 in order to solve anisotropic linear systems with zero curves. Again, we have proved two-grid convergence for the Galerkin method and presented two heuristics for the design of efficient multigrid methods with sparse matrices on coarser grids.
- Eventually, we have constructed multigrid preconditioners with slightly different techniques as were used for the development of the solvers.

There is still plenty of room for extending these results. Here are some suggestions:

- The Galerkin-based method applied to Toeplitz systems converges optimally in all experiments. Therefore, an extension of the two-grid proof from trigonometric matrix algebras to Toeplitz matrices seems to be possible.

- An analysis of the multigrid methods from Sections 5.2 and 5.3 with the Local Fourier Analysis (see [115, 112]) seems worthwhile. This could be helpful for the decision how accurate an approximation of the zero curve must be.

Applications

In this thesis, we have presented results for two applications. These are image deblurring problems and the solution of partial differential equations:

- We have applied our multigrid methods to anisotropic image deblurring problems with shift-invariant point spread function. Noise-free pictures are deblurred with a multigrid method consisting of semicoarsening and full coarsening steps. This technique also works well if anisotropy occurs along lines that are not in parallel with one of the coordinate axes. In the presence of noise, we employ Tikhonov or Riley regularization and solve the resulting linear systems with the multigrid methods from Chapter 4.
- Anisotropic structured linear systems appear when anisotropic, elliptic PDEs with constant coefficients are solved or when trigonometric matrix algebra preconditioners are applied to variable coefficient problems. In both cases, the multigrid methods from Chapter 4 are highly efficient. Moreover, we have discussed multigrid solution techniques for the Helmholtz equation with constant coefficients and also for an anisotropic version of the Helmholtz equation.

We suggest the following extensions of our results for future research:

- In addition to Tikhonov and Riley regularization, Donatelli and Serra [42] suggest that multigrid methods be used as regularizers. We expect that this approach can be applied to anisotropic deblurring problems with the same efficiency.
- The treatment of other boundary value problems will be subject of future research. A straightforward extension should be possible for other Toeplitz plus Hankel matrices such as the DST-III matrices. A more difficult task is an adaptation of our methods to complex boundary conditions such as the Sommerfeld radiation boundary conditions [49]. Furthermore, an extension of our methods for the solution of the Helmholtz equation with variable coefficients seems a challenging task for the future.

Bibliography

- [1] G. Ammar and W. Gragg. Superfast solution of real positive definite Toeplitz systems. *SIAM J. Matrix Anal. Appl.*, 9:61–76, 1988.
- [2] A. Arico and M. Donatelli. A V-cycle multigrid for multilevel matrix algebras: proof of optimality and applications. *Numer. Math., to appear*.
- [3] A. Arico, M. Donatelli, and S. Serra-Capizzano. V-cycle optimal convergence for certain (multilevel) structured linear systems. *SIAM J. Mat. Anal. Appl.*, 26:186–214, 2004.
- [4] G. Axelsson and M. Neytcheva. The algebraic multilevel iteration methods - theory and applications. In *Proceedings of the 2nd. Int. Coll. on Numerical Analysis, D. Bainov Ed.*, Bulgaria, 1994. VSP 1994.
- [5] O. Axelsson and V. Barker, editors. *Finite element solution of boundary value problems, theory and computation*. Academic Press, Orlando, FL, 1984.
- [6] F. Di Benedetto. Analysis of preconditioning techniques for ill-conditioned Toeplitz matrices. *SIAM J. Sci. Comp.*, 16:682–697, 1995.
- [7] F. Di Benedetto. Preconditioning of block Toeplitz matrices by sine transforms. *SIAM J. Sci. Comp.*, 18(2):499–515, 1997.
- [8] D. Bini and F. Di Benedetto. A new preconditioner for the parallel solution of positive definite Toeplitz systems. In *Second ACM Symp. on Parallel Algorithms and Architectures*, pages 220–223, Crete, Greece, 1990.
- [9] D. Bini and M. Capovani. Spectra and computational properties of band symmetric Toeplitz matrices. *Linear Algebra Appl.*, 52:99–126, 1983.
- [10] D. Bini and P. Favati. On a matrix algebra related to the discrete Hartley transform. *SIAM J. Matrix Anal. Appl.*, 14:500–507, 1993.
- [11] R. R. Bitmead and B. D. O. Anderson. Asymptotically fast solution of Toeplitz and related systems of equations. *Linear Algebra Appl.*, 34:103–116, 1980.

- [12] J. H. Bramble, J. E. Pasciak, and J. Xu. Parallel multilevel preconditioners. *Math. Comp.*, 55:1–22, 1990.
- [13] A. Brandt. Multi-level adaptive solutions to boundary value problems. *Math. Comput.*, 31:333–390, 1977.
- [14] A. Brandt. Algebraic multigrid theory: The symmetric case. *Appl. Math. Comput.*, 19:23–56, 1986.
- [15] A. Brandt. Rigorous quantitative analysis of multigrid: I. Constant coefficients two-level cycle with L_2 norm. *SIAM J. Numer. Anal.*, 31:1695–1730, 1994.
- [16] A. Brandt and I. Livshits. Wave-ray multigrid method for standing wave equations. *ETNA*, 6:162–181, 1997.
- [17] A. Brandt, S. F. McCormick, and J. Ruge. Algebraic multigrid (AMG) for sparse matrix equations. pages 257–284, 1984.
- [18] R. Brent, F. Gustavson, and D. Yun. Fast solution of Toeplitz systems of equations and computation of Padé approximations. *J. Algo.*, 1:259–295, 1980.
- [19] W. L. Briggs, V. E. Henson, and S. F. McCormick. *A multigrid tutorial*. SIAM, Philadelphia, second edition, 2000.
- [20] R. Chan. Circulant preconditioners for Hermitian Toeplitz systems. *SIAM J. Matrix Anal. Appl.*, 10:542–550, 1989.
- [21] R. Chan. Toeplitz preconditioners for Toeplitz systems with nonnegative generating functions. *IMA J. Numer. Anal.*, 11:333–345, 1991.
- [22] R. Chan and T. Chan. Circulant preconditioners for elliptic problems. *Num. Lin. Alg. Appl.*, 1:77–101, 1992.
- [23] R. Chan, T. Chan, and C. Wong. Cosine transform based preconditioners for total variation deblurring. *IEEE Trans. Image Proc.*, 8:1472–1478, 1999.
- [24] R. Chan, Q. Chang, and H. Sun. Multigrid methods for ill-conditioned Toeplitz systems. *SIAM J. Sci. Comp.*, 19:516–529, 1998.
- [25] R. Chan, W. Ching, and C. Wong. Optimal trigonometric preconditioners for elliptic problems and queueing problems. *SEA Bull. Math*, 20:117–124, 1996.
- [26] R. Chan, X. Jin, and M. Yeung. The circulant operator in the Banach algebra of matrices. *Linear Algebra Appl.*, 149:41–53, 1991.

- [27] R. Chan and M. Ng. Conjugate gradient methods for Toeplitz systems. *SIAM Review*, 38:427–482, 1996.
- [28] R. Chan, S. Serra-Capizzano, and C. Tablino-Possio. Two-grid methods for banded linear systems from DCT III algebra. *Numer. Linear Algebra Appl.*, 12:241–249, 2005.
- [29] R. Chan and P. Tang. Fast band-Toeplitz preconditioners for Hermitian Toeplitz systems. *SIAM J. Sci. Comp.*, 15:164–171, 1994.
- [30] R. Chan and C. Wong. Sine transform based preconditioners for elliptic problems. *Numer. Linear Algebra Appl.*, 4:351–368, 1997.
- [31] R. Chan and M. Yeung. Circulant preconditioners constructed from kernels. *SIAM J. Numer. Anal.*, 29:1093–1103, 1992.
- [32] R. Chan and M. Yeung. Circulant preconditioners for Toeplitz matrices with positive continuous generating function. *Math. Comp.*, 58:233–240, 1992.
- [33] R. H. Chan, M. Donatelli, S. Serra-Capizzano, and C. Tablino-Possio. Application of multigrid techniques to image restoration problems. In F. Luk, editor, *Proceedings to the SPIE Symposium on Advanced Signal Processing: Algorithms, Architectures, and Implementations*, volume 4791, pages 210–221, San Diego, CA, 2002.
- [34] T. Chan. An optimal circulant preconditioner for Toeplitz systems. *SIAM J. Sci. Stat. Comp.*, 9:766–771, 1988.
- [35] T. Chan and J. Olkin. Circulant preconditioners for Toeplitz-block matrices. *Numerical Algorithms*, 6:89–101, 1994.
- [36] P. Davis. *Circulant matrices*. John Wiley and Sons, New York, 1979.
- [37] M. Donatelli. A multigrid method for restoration and regularization of images with Dirichlet boundary conditions. In *Proceedings SPIE, Advanced Signal Processing Algorithms, Architectures and Implementations XIII*, volume 5205, pages 358–368, San Diego, CA, 2003.
- [38] M. Donatelli. *Image deconvolution and multigrid methods*. PhD thesis, University of Milano, November 2005.
- [39] M. Donatelli. A multigrid for image deblurring with Tikhonov regularization. *Numer. Linear Algebra Appl.*, 12:715–729, 2005.
- [40] M. Donatelli, C. Estatico, J. Nagy, L. Perrone, and S. Serra-Capizzano. Anti-reflective boundary conditions and fast 2D deblurring models. In *Proceedings SPIE, Advanced Signal Processing Algorithms, Architectures and Implementations XIII*, volume 5205, pages 380–389, San Diego, CA, 2003.

- [41] M. Donatelli and S. Serra-Capizzano. Anti-reflective boundary conditions and reblurring. *Inverse Problems*, 21:169–182, 2004.
- [42] M. Donatelli and S. Serra-Capizzano. On the regularizing power of multigrid-type algorithms. *SIAM J. Sci. Comput.*, 27:2053–2076, 2006.
- [43] J. Dongarra and F. Sullivan. Top ten algorithms of the century. *Computing in Science and Engineering*, 2:22–23, 2000.
- [44] M. Dryja and O. B. Widlund. Some domain decomposition algorithms for elliptic problems. In L. Hayes and D. Kincaid, editors, *Iterative methods for large linear systems*, pages 273–291, San Diego, CA, 1989. Academic Press.
- [45] M. Dryja and O. B. Widlund. Towards a unified theory of domain decomposition algorithms for elliptic problems. In T. F. Chan, R. Glowinsky, J. Periaux, and O. B. Widlund, editors, *Third international symposium on domain decomposition methods for partial differential equations*, pages 3–21, Philadelphia, 1990. SIAM.
- [46] M. Dryja and O. B. Widlund. Multilevel additive methods for elliptic finite element problems. In W. Hackbusch, editor, *Parallel Algorithms for Partial Differential Equations, Proceedings of the 6th GAMM-Seminar*, Braunschweig, Germany, 1991. Vieweg.
- [47] H. Elman, O. Ernst, and D. O’Leary. A multigrid method enhanced by Krylov subspace iteration for discrete Helmholtz equations. *SIAM J. Sci. Comp.*, 23:1290–1314, 2001.
- [48] H. C. Elman and X. Zhang. Algebraic analysis of the hierarchical basis preconditioner. *SIAM J. Matrix Anal. Appl.*, 16(1):192–205, 1995.
- [49] Y. Erlangga, C. Oosterlee, and C. Vuik. A novel multigrid based preconditioner for heterogeneous Helmholtz problems. *To appear in SIAM J. Sci. Comp.*
- [50] Y. Erlangga, C. Vuik, and C. Oosterlee. On a class of preconditioners for solving the Helmholtz equation. *Appl. Num. Math.*, 50:409–425, 2004.
- [51] C. Di Fiore and P. Zellini. Matrix decompositions using displacement rank and classes of commutative matrix algebras. *Linear Algebra Appl.*, 229:49–99, 1995.
- [52] G. Fiorentino and S. Serra. Multigrid methods for Toeplitz matrices. *Calcolo*, 28:283–305, 1992.
- [53] G. Fiorentino and S. Serra. Multigrid methods for indefinite symmetric Toeplitz matrices. *Calcolo*, 33:223–236, 1996.

- [54] G. Fiorentino and S. Serra. Multigrid methods for symmetric positive definite block Toeplitz matrices with nonnegative generating functions. *SIAM J. Sci. Comp.*, 17:1068–1081, 1996.
- [55] R. Fischer and T. Huckle. Multigrid methods for anisotropic BTTB systems. *Accepted for publication in Lin. Alg. Appl. (Special issue on the 80th birthday of F. L. Bauer)*.
- [56] R. Fischer and T. Huckle. Multigrid solution techniques for anisotropic structured linear systems. *Accepted for publication in IMACS J. Appl. Num. Math. (Special issue on the 7th IMACS conference on iterative methods)*.
- [57] R. Fischer and T. Huckle. Multigrid methods for strongly ill-conditioned structured matrices. *Submitted to ECOMAS Proceedings of the 8th European Multigrid Conference, 2005*.
- [58] G. Golub and C. Van Loan. *Matrix computations*. Johns Hopkins University Press, Baltimore, MD, second edition, 1989.
- [59] G. Golub and J. M. Ortega, editors. *Scientific computing, An introduction with parallel computing*. Academic Press, 1993.
- [60] R. Gonzalez and R. Woods. *Digital image processing*. Addison Wesley, New York, 1992.
- [61] A. Greenbaum. *Iterative methods for solving linear systems*. SIAM, Seattle, Washington, 1997.
- [62] U. Grenander and G. Szegö. *Toeplitz forms and their applications*. Chelsea Publishing, New York, second edition, 1984.
- [63] M. J. Grote and T. Huckle. Parallel preconditioning with sparse approximate inverses. *SIAM J. Sci. Comput.*, 18:838–853, 1997.
- [64] W. Hackbusch. On the convergence of a multi-grid iteration applied to finite element equations. Technical Report 77-8, Institute for Applied Mathematics, University of Cologne, West Germany, 1977.
- [65] W. Hackbusch. Multi-grid convergence theory. In W. Hackbusch and U. Trottenberg, editors, *Multigrid methods*, volume 1982 of *Lecture Notes in mathematics 960*, pages 177–219. Springer, Berlin, 1982.
- [66] W. Hackbusch, editor. *Multigrid methods and applications*. Springer, 1985.
- [67] W. Hackbusch, editor. *Iterative solution of large linear systems of equations*. Springer Verlag, New York, 1994.

- [68] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *J. Res. Nat. Bur. Stand.*, 49:409–436, 1952.
- [69] T. Huckle. A note on skewcirculant preconditiones for elliptic problems. *Num. Alg.*, 2:279–286, 1992.
- [70] T. Huckle. Some aspects of circulant preconditiones. *SIAM J. Sci. Comput.*, 14:531–541, 1993.
- [71] T. Huckle and J. Staudacher. Multigrid methods for block Toeplitz matrices with small size blocks. *To appear in BIT*.
- [72] T. Huckle and J. Staudacher. Multigrid preconditioning and Toeplitz matrices. *ETNA*, 13:81–105, 2002.
- [73] T. Kailath, S. Kung, and M. Morf. Displacement ranks of matrices and linear equations. *J. Math. Appl.*, 68:395–407, 1979.
- [74] T. Kailath and A. H. Sayed, editors. *Fast reliable algorithms for matrices with structure*. SIAM, 1999.
- [75] L. Landweber. An iteration formula for Fredholm integral equations of the first kind. *Amer. J. Math.*, 73:615–624, 1951.
- [76] N. Levinson. The Wiener RMS (Root-Mean-Square) error criterion in filter design and prediction. *J. Math. Phys.*, 25:261–278, 1946.
- [77] C. Van Loan. *Computational frameworks for the Fast Fourier Transform*. SIAM, 1992.
- [78] J. A. Meijerink and H. A. Van der Vorst. An iterative solution for linear systems of which the coefficient matrix is a symmetric M-matrix. *Math. Comp.*, 31:148–162, 1977.
- [79] M. Miranda and P. Tilli. Asymptotic spectra of Hermitian block Toeplitz matrices and preconditioning results. *SIAM J. Matrix Anal. Appl.*, 21:867–881, 2000.
- [80] M. Morf. Doubling algorithms for Toeplitz and related equations. *Proc. IEEE Internat. conf. on Acoustics, Speech and Signal Processing*, pages 954–959, 1980.
- [81] B. Musicus. Levinson and fast Cholesky algorithm for Toeplitz and almost Toeplitz matrices. Technical report, Research Lab. of Electronics, MIT, Cambridge, MA, 1981.
- [82] M. Ng. Band preconditioners for block-Toeplitz-Toeplitz-block systems. *Lin. Alg. Appl.*, 259:307–327, 1997.

- [83] M. K. Ng, editor. *Iterative methods for Toeplitz systems*. Oxford University Press, Oxford, 2004.
- [84] D. Noutsos, S. Serra-Capizzano, and P. Vassalos. Spectral equivalence and matrix algebra preconditioners for multilevel Toeplitz systems: a negative result. *Contemp. Math.*, 323:313–322, 2003.
- [85] D. Noutsos, S. Serra-Capizzano, and P. Vassalos. Matrix algebra preconditioners for multilevel Toeplitz systems do not insure optimal convergence rate. *Theoretical Computer Science*, 315:557–579, 2004.
- [86] D. Noutsos, S. Serra-Capizzano, and P. Vassalos. A preconditioning proposal for ill-conditioned Hermitian two-level Toeplitz systems. *Num. Lin. Alg. Appl.*, 12:231–239, 2005.
- [87] D. Noutsos and P. Vassalos. New band Toeplitz preconditioners for ill-conditioned symmetric positive definite Toeplitz systems. *SIAM J. Matr. Anal. Appl.*, 23:728–743, 2002.
- [88] D. Potts and G. Steidl. Preconditioners for ill-conditioned Toeplitz matrices. *BIT*, 39:513–533, 1999.
- [89] J. D. Riley. Solving systems of linear equations with a positive definite, symmetric, but ill-conditioned matrix. *Math. Tables Aids Comput.*, 9:96–101, 1955.
- [90] J. W. Ruge and K. Stüben. Algebraic multigrid. In *Frontiers in Applied Mathematics: Multigrid Methods*, S. McCormick Ed., pages 73–130, Philadelphia, 1987. SIAM.
- [91] S. Serra. Preconditioning strategies for asymptotically ill-conditioned block Toeplitz systems. *BIT*, 34:597–594, 1994.
- [92] S. Serra. Optimal, quasi-optimal and superlinear band-Toeplitz preconditioners for asymptotically ill-conditioned positive definite Toeplitz systems. *Math. Comp.*, 66:651–665, 1997.
- [93] S. Serra. Asymptotic results on the spectra of block Toeplitz preconditioned matrices. *SIAM J. Matrix Anal. Appl.*, 20:31–44, 1998.
- [94] S. Serra. On the extreme eigenvalues of Hermitian (block) Toeplitz matrices. *Linear Algebra Appl.*, 270:109–129, 1998.
- [95] S. Serra. A Korovkin-type theory for finite Toeplitz operators via matrix algebras. *Numer. Math.*, 82(1):117–142, 1999.

- [96] S. Serra. Spectral and computational analysis of block Toeplitz matrices having nonnegative definite matrix-valued generating functions. *BIT*, 39:152–175, 1999.
- [97] S. Serra. Superlinear PCG methods for symmetric Toeplitz systems. *Math. Comp.*, 68:793–803, 1999.
- [98] S. Serra-Capizzano. Spectral behavior of matrix sequences and discretized boundary value problems. *Lin. Alg. Appl.*, 337:37–78, 2001.
- [99] S. Serra-Capizzano. Convergence analysis of two-grid methods for elliptic Toeplitz and PDEs matrix-sequences. *Numer. Math.*, 92:433–465, 2002.
- [100] S. Serra-Capizzano. Matrix algebra preconditioners for multilevel Toeplitz matrices are not superlinear. *Lin. Alg. Appl.*, 343–344:303–319, 2002.
- [101] S. Serra-Capizzano. A note on anti-reflective boundary conditions and fast deblurring models. *SIAM J. Sci. Comput.*, 25:1307–1325, 2004.
- [102] S. Serra-Capizzano and C. Tablino-Possio. Preliminary remarks on multigrid methods for circulant matrices. In L. Vulkov, J. Waśniewski, and P. Yalamov, editors, *Numerical Analysis and its Applications*, volume 1988 of *Lecture notes in Computer Science 1988*, pages 152–159. Springer-Verlag, Berlin, 2001.
- [103] S. Serra-Capizzano and C. Tablino-Possio. Multigrid methods for multilevel circulant matrices. *SIAM J. Sci. Comp.*, 26(1):55–85, 2005.
- [104] S. Serra-Capizzano and E. E. Tyrtyshnikov. Any circulant-like preconditioner for multilevel matrices is not superlinear. *SIAM J. Matrix Anal. Appl.*, 21(2):431–439, 1999.
- [105] S. Serra-Capizzano and E. E. Tyrtyshnikov. How to prove that a preconditioner cannot be superlinear. *Math. Comp.*, 72:1305–1316, 2003.
- [106] B. Smith, P. Björstadt, and W. Gropp, editors. *Domain decomposition*. Cambridge University Press, 1996.
- [107] G. Strang. A proposal for Toeplitz matrix calculations. *Stud. Appl. Math.*, 74:171–176, 1986.
- [108] H. Sun, R. Chan, and Q. Chang. A note on the convergence of the two-grid method for Toeplitz systems. *Comput. Math. Appl.*, 34:11–18, 1997.
- [109] H. Sun, X. Jin, and Q. Chang. Convergence of the multigrid method for ill-conditioned Block Toeplitz systems. *BIT*, 41:179–190, 2001.

- [110] A. N. Tikhonov. Solution of incorrectly formulated problems and regularization method. *Soviet Math. Dokl.*, 4:1035–1038, 1963.
- [111] W. Trench. An algorithm for the inversion of finite Toeplitz matrices. *SIAM J. Appl. Math.*, 12:515–522, 1964.
- [112] U. Trottenberg, C. Oosterlee, and K. Schüller. *Multigrid*. Academic Press, 2001.
- [113] E. E. Tyrtysnikov. Optimal and superoptimal circulant preconditioners. *SIAM J. Matrix Anal. Appl.*, 13:459–473, 1992.
- [114] E. E. Tyrtysnikov. Circulant preconditioners with unbounded inverses. *Linear Algebra Appl.*, 216:1–24, 1995.
- [115] R. Wienands and W. Joppich, editors. *Practical Fourier analysis for multigrid methods*. CRC Press, Boca Raton, USA, 2004.
- [116] J. Xu. Iterative methods by space decomposition and subspace correction. *SIAM Review*, 34:581–613, 1992.
- [117] H. Yserentant. Hierarchical bases give conjugate gradient type methods a multigrid speed of convergence. *Appl. Math. Comp.*, 19:347–358, 1986.
- [118] S. Zohar. The solution of a Toeplitz set of linear equations. *J. Assoc. Comput. Mach.*, 21:272–276, 1974.