Publication Series of the John von Neumann Institute for Computing (NIC)
NIC Series Volume 41

John von Neumann Institute for Computing (NIC)

Matthias Bolten

Multigrid methods for structured grids and their application in particle simulation

NIC Series Volume 41 ISBN 978-3-9810843-7-5

Die Deutsche Bibliothek – CIP-Cataloguing-in-Publication-Data
A catalogue record for this publication is available from Die Deutsche Bibliothek

Publisher: NIC-Directors
Distributor: NIC-Secretariat
Research Centre Jülich
52425 Jülich
Germany
Internet: www.fz-juelich.de/nic
Printer: Graphische Betriebe, Forschungszentrum Jülich

NIC Series Volume 41 ISBN 978-3-9810843-7-5

Multigrid methods for structured grids and their application in particle simulation

Zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften

am Fachbereich Mathematik der
Bergischen Universität Wuppertal
genehmigte

Dissertation

von

Dipl.-Inf. Matthias Bolten

Tag der mündlichen Prüfung:	8. Juli 2008
Referent:	Prof. Dr. A. Frommer
Korreferent:	Prof. Dr. Dr. Th. Lippert
Korreferent:	Prof. J. Brannick, PhD

Contents

1 Introduction **1**

2 Partial Differential Equations **5**

2.1 Introduction 5

2.1.1 Boundary conditions 6

2.2 Elliptic partial differential equations 7

2.2.1 Prerequisites from functional analysis 7

2.2.2 Prerequisites from Fourier analysis 11

2.2.3 Weak formulation of a PDE 13

2.2.4 Existence and uniqueness of the weak solution 14

2.2.5 Regularity of the solution for PDEs with Dirichlet boundary conditions 16

2.2.6 Construction of the solution for PDEs with open boundary conditions 16

2.2.7 Construction of the solution for PDEs on the torus 19

2.3 Numerical solution 21

2.3.1 Solution of PDEs on the torus or on subsets of $\mathbb{R}^d$ with Dirichlet boundary conditions using finite differences 21

2.3.2 Finite volume discretization-based solution of PDEs defined on $\mathbb{R}^d$ 26

3 Multigrid Methods **43**

3.1 Iterative methods 43

3.1.1 Linear iterative methods 44

3.1.2 Splitting methods 45

3.1.3 Relaxation methods 48

3.2 Geometric Multigrid 49
3.2.1 Motivation 49
3.2.2 Twogrid methods 52
3.2.3 Multigrid methods 58
3.2.4 FAS and FAC 62
3.3 Algebraic Multigrid Theory for Structured Matrices 64
3.3.1 Convergence theory for multigrid methods for hermitian positive definite problems 65
3.3.2 Replacement of the Galerkin operator 72
3.3.3 Application to circulant matrices 81
3.3.4 Circulant matrices 81
3.3.5 Multigrid methods for circulant matrices 82
3.3.6 Replacement of the Galerkin operator for circulant matrices . . . 84
3.3.7 Replacement strategies for the Galerkin operator for circulant matrices with compact stencils 88
3.3.8 Numerical Examples 91
3.4 Parallelization 95
3.4.1 Data distribution for banded matrices 96
3.4.2 Example results on Blue Gene/L and Blue Gene/P 98
3.4.3 Further parallelization issues 102

4 Particle Simulation **103**
4.1 Introduction 103
4.2 Mathematical formulation 104
4.2.1 Open systems 105
4.2.2 Periodic systems 106
4.2.3 Relation to the Poisson equation 107
4.3 Numerical solution 107
4.3.1 Mesh-free methods 107
4.3.2 Mesh-based methods 108
4.4 Meshed continuum method 110
4.4.1 Derivation of the method 111
4.4.2 Point symmetric densities described by B-splines 114
4.4.3 Numerical experiments 115

5 Conclusion **121**

Acknowledgments **123**

List of Figures

2.1 Coarsened grid in 2D 29
2.2 Conservative discretization at the interface in 2D 30
2.3 Cut through computed solution and analytic point-wise error on 64^3 grid . 39
2.4 Behavior of the error of the original method and of the modification . . . 40

3.1 Error of 5-point Laplacian after 0, 1 and 3 iterations of damped Jacobi . . 51
3.2 Damping factors for 1D Laplacian 53
3.3 Algebraically smooth error for mixture of PDE and integral equation . . . 67
3.4 Generating symbols of Galerkin operator, replacement operator, and their ratio 93
3.5 Convergence of multigrid for 5-point Laplacian using Galerkin operator and the replacement operator 93
3.6 Convergence of multigrid for mixed PDE and integral equation using Galerkin operator and the replacement operator 94
3.7 Pattern of 1D nearest neighbor communication 97
3.8 Speedup for the V-cycle and the W-cycle compared 99
3.9 Blue Gene/L speedup and efficiency for 7-point discretization of Laplacian and 128^3 unknowns 99
3.10 Blue Gene/P speedup and efficiency for 7-point discretization of Laplacian and 1024^3 unknowns 101
3.11 Blue Gene/L weak scaling for 7-point Laplacian and $64 \times 128 \times 128$ unknowns per processor 102

4.1 Influence of the width of the charge distribution for various grid spacings for the 7-point discretization of the Laplacian 117
4.2 Influence of the width of the charge distribution for various grid spacings for the compact fourth-order discretization of the Laplacian 118
4.3 Scaling behavior of Algorithm 4.2 119

List of Tables

2.1 Error and timings for different various sizes 39
2.2 Error norms for a 33^3-problem with $h = 1/32$ and various refinements . . 40
2.3 Error norms for a 33^3-problem with $h = 1/32$ and various refinements using the method of Washio and Oosterlee 41

3.1 Convergence Galerkin coarse grid operator 7-point Laplacian 95
3.2 Convergence replacement coarse grid operator 7-point Laplacian 95
3.3 Blue Gene/L timings for 7-point Laplacian and 128^3 unknowns 100
3.4 Blue Gene/P timings for 7-point Laplacian and 1024^3 unknowns 100
3.5 Blue Gene/L weak scaling for 7-point Laplacian and $64 \times 128 \times 128$ unknowns per processor 101

4.1 Error of second-order discretization of calculated distribution's potential for different distribution widths and grid spacings 116
4.2 Error of fourth-order discretization of calculated distribution's potential for different distribution widths and grid spacings 116
4.3 Influence of the width of the charge distribution for various grid spacings for the 7-point discretization of the Laplacian 117
4.4 Influence of the width of the charge distribution for various grid spacings for the compact fourth-order discretization of the Laplacian 118
4.5 Scaling behavior and accuracy of Algorithm 4.2 for randomly distributed particles and compact fourth-order discretization 119
4.6 Relative error of the electrostatic energy of a DNA fragment calculated for various grid spacings using the compact fourth-order discretization 120

Chapter 1

Introduction

This work is focussed on the application of multigrid methods to particle simulation methods. Particle simulation is important for a broad range of scientific fields, like biophysics, astrophysics or plasma physics, to name a few. In these fields computer experiments play an important role, either supporting real experiments or replacing them. The first can significantly reduce costs, e.g. in the pharmaceutic industry, where possible agents can be checked for an effect in advance of real and expensive experiments. The latter has an important role in astrophysics, where most experiments just cannot be carried out in a laboratory. In the cases we are interested in, the interaction of particles can be evaluated by pairwise potentials, where short-ranged potentials, e.g. potentials describing chemical bonds, are easy to be implemented efficiently. But the very important Coulomb potential and the gravitational potential are not short-ranged, thus an intuitive implementation has to evaluate all pairwise interactions, yielding an $\mathcal{O}(N^2)$ algorithm, where N is the number of particles to be simulated. The key to reduce this complexity is the use of approximate algorithms for the evaluation of the long-ranged potentials.

In the Coulomb or gravitational potential case we have a variety of options. One option is the use of tree-codes, that approximate particles that are far away by a bigger pseudo-particle. Furthermore, in the periodic case we have the option of calculating the convolution with the influence function given by the potential in Fourier space. We are exploiting the fact that the Coulomb or gravitational potential is strongly connected to the Poisson equation, i.e. up to a constant the Green's function of the Poisson equation and these potentials are the same. Given this fact, we are able to solve the problem numerically by sampling a special right hand side onto a mesh describing either a torus or a section of the open space and solving the equation numerically. After the solution is available on the mesh, the electrostatic quantities of interest can be obtained from this discrete solution by interpolating it back to the particles and applying a correction scheme. Given these considerations the problem can be reduced to using a fast Poisson solver for the numerical solution of the Poisson equation on the mesh. Multigrid method are known to be very efficient solvers for

the Poisson equation and similar PDEs, so we choose to use Multigrid methods for that purpose.

In the open boundary case the Poisson equation has to be solved in open space. The problem is that this leads to infinitely large systems. The number of grid points can be reduced easily, as far away from the system the solution will change only very little. Washio and Oosterlee [87] were able to provide an error analysis for such a hierarchically coarsened grid. They suggest to calculate a finite subvolume, only, while setting the boundary values to zero, assuming that the induced error can be neglected if the volume is large enough. They did not provide an estimate for this error, though. We extend their method to impose certain boundary conditions at the boundary of the system and provide an estimate for the error of the modified method. This estimate shows that the modified method is of the desired accuracy. Additionally we show that the method is still optimal for a number of refinement steps that can be precomputed easily. The resulting system can be solved using the well-known FAC method, which is an extension of standard geometric multigrid methods for adaptive grids.

For molecular dynamics simulation, the periodic case is of special importance. The solution of the Poisson equation with constant coefficients on an equidistant regular grid using a discretization technique like finite differences leads to circulant matrices. Circulant matrices form a matrix algebra and can be analyzed elegantly. Recently, multigrid methods for circulant matrices have been developed, see e.g. [2, 74]. The theory for these methods is based on a variational property which is fulfilled when the Galerkin operator is used. This operator gets denser when going down to coarser levels, i.e. we end up with a fully filled stencil after a few coarsening steps, even if the original stencil was sparse. Motivated by the fact that this is not necessary in geometric multigrid methods using a rediscretization of the system with finite differences, and motivated by a stencil collapsing technique introduced in [4] we develop necessary conditions for the V-cycle convergence of multigrid methods not using the Galerkin operator but rather a replacement. We apply these theoretical considerations to certain circulant matrices and present schemes for these matrices that fulfill these properties. As a result we obtain very efficient solvers for circulant matrices.

The rest of this work is structured as follows: In Chapter 2 we will cover partial differential equations. After the definition and classification of partial differential equations we will present various results for the existence, uniqueness and regularity of the solution of elliptic partial differential equations. We present different discretization techniques, namely finite differences, compact discretizations of higher order and the finite volume discretization. The chapter closes with an overview of Washio's and Oosterlee's method and the modification to it, as well as with some numerical examples. After that, in Chapter 3 we introduce iterative solvers and multigrid methods. After a short introduction to general iterative methods and geometric multigrid methods including FAS and FAC, we continue with algebraic multigrid theory for structured matrices. As part of this theory we present the new theoretical considerations for non-Galerkin coarse grid operators and the application to circulant matrices. Thereafter, a short overview over the parallelization of multigrid methods

and some results for our parallel code for circulant matrices are presented. Chapter 4 deals with particle simulation. After an introduction to the problem and a brief overview over available methods we give a mathematical formulation of the problem that consistently uses the Poisson equation and which allows the use of multigrid methods for the solution of these problems. We finish this work with a conclusion in Chapter 5.

Chapter 2

Partial Differential Equations

The development of multigrid methods is strongly connected to their application to the solution of partial differential equations. As the simulation of particle systems leads to a partial differential equation as well, in this chapter we will give a short overview over partial differential equations and the associated theory.

2.1 Introduction

Unlike ordinary differential equations which involve univariate functions, partial differential equations involve multivariate functions. In the following, we call an open and connected subset of $\mathbb{R}^d$ a *domain*. By Ω we denote a bounded domain and its boundary by $\partial\Omega$ or Γ. A formal definition of a partial differential equation is given by the following:

Definition 2.1 (Partial differential equation) *Let $\Omega \subset \mathbb{R}^d$. An equation of the form*

$$\mathcal{F}\left(\mathbf{x}, u(\mathbf{x}), \frac{\partial}{\partial x_1}u(\mathbf{x}), \ldots, \frac{\partial}{\partial x_d}u(\mathbf{x}), \frac{\partial^2}{\partial x_1^2}u(\mathbf{x}), \frac{\partial^2}{\partial x_2 \partial x_1}u(\mathbf{x}), \ldots\right) = 0$$

with $\mathbf{x} \in \mathbb{R}^d$ and $u \in C^k(\Omega)$ and where F depends only on $\mathbf{x}$ and the value of u and the partial derivatives of u at $\mathbf{x}$ is called a partial differential equation *or* PDE *for short.*

Partial differential equations are classified by their order k, i.e. the maximum occurring order of the derivatives. Furthermore PDEs are distinguished by the type of linearity. If $\mathcal{F}$ depends only linearly on u and all partial derivatives, i.e. the coefficient functions depend only on $\mathbf{x}$, the PDE is called *linear*. If it depends only linearly on the partial derivatives of highest order but non-linearly on u and all other partial derivatives it is called *semilinear*. It is called *quasilinear*, if the coefficient functions of the partial derivatives of highest degree depend only on lower-order derivatives and u. Otherwise the equation is called a *non-linear PDE*.

Linear partial differential equations are well studied and a number of different numerical methods exist for their solution. Many physical problems, e.g. heat conduction or wave propagation, lead to second-order linear PDEs. These are classified in the following way:

Definition 2.2 (Classification of linear PDEs of second order) *Considering a linear PDE of second order of the form*

$$\mathcal{L}u(\mathbf{x}) = -\sum_{i,j=1}^{d} a_{i,j}(\mathbf{x})\frac{\partial^2}{\partial x_i \partial x_j}u(\mathbf{x}) + \sum_{j=1}^{d} b_j(\mathbf{x})\frac{\partial}{\partial x_j}u(\mathbf{x}) + c(\mathbf{x})u(\mathbf{x}) = f(\mathbf{x}).$$

Depending on the eigenvalues of the coefficient matrix $A = (a_{i,j})_{i,j=1}^{d}$ these PDEs are called:

- elliptic - *all eigenvalues of A have same sign,*
- parabolic - *all eigenvalues of A, except for one vanishing eigenvalue, have same sign,*
- hyperbolic - *all eigenvalues of A have same sign, except for one eigenvalue that has the opposite sign.*

As geometric multigrid methods are optimal methods for certain elliptic PDEs, in the remaining sections we focus on this class of problems.

A PDE by itself usually has multiple solutions. In order to obtain a unique solution, we need boundary conditions or initial conditions, i.e. given values on the domain's boundary or parts of the boundary of the domain. This leads to boundary value problems or initial value problems, respectively.

2.1.1 Boundary conditions

Various different boundary conditions are known in literature. In this work, we are using the following conditions:

- **Open boundary conditions**
  Open boundary conditions are not very common, although they can be handled very elegantly in theory. If a partial differential equation $\mathcal{F}u = f$ is defined on $\Omega = \mathbb{R}^d$, the solution usually is still not unique. Therefore, a value of u can be prescribed for $\mathbf{x} \in \partial\Omega$, which in this case we consider to be the point ∞ in the 1-point compactification of $\Omega = \mathbb{R}^d$. In order for the solution u to have nice analytical properties, e.g. $u \in L_2$, the following condition is usually required:

  $$u(\mathbf{x}) \overset{\|\mathbf{x}\|\to\infty}{\longrightarrow} 0.$$

- **Dirichlet boundary conditions**
  Let the partial differential equation $\mathcal{F}u = f$ be defined on its domain Ω. Boundary conditions of the form

  $$u = g \text{ on } \Gamma$$

are known as Dirichlet boundary conditions. PDEs with Dirichlet boundary conditions are well-analyzed and a lot of theory exists for existence, uniqueness etc. of the solution, especially with respect to the properties of the boundary and to the smoothness of g.

- **Periodic boundary conditions**
  If a partial differential equation is defined on the torus $\mathbb{R}^d/\mathbb{Z}^d$, boundary conditions are not needed. Nevertheless, in this case one often speaks of periodic boundary conditions. PDEs can be analyzed very elegantly and solved efficiently on the torus using Fourier techniques.

2.2 Elliptic partial differential equations

A large class of important stationary problems leads to elliptic PDEs, namely diffusion-like problems like those described by the electrostatic or the gravitational potential. In accordance to the introductory book by Larsson and Thomée [61], which is the basis of this introduction, we study the equation

$$\mathcal{L}u := -a\Delta u + b \cdot \nabla u + cu = f$$

in larger detail. In order to keep the analysis of this equation consistent with all kinds of boundary conditions needed here, we choose a variational formulation. So in the following the domain Ω of the PDE is either equal to or a subset of $\mathbb{R}^d$ or it is the d-dimensional torus $\mathbb{R}^d/\mathbb{Z}^d$ unless noted otherwise. In case that Ω is a subset of $\mathbb{R}^d$, the solution u shall fulfill $u = g$ on Γ and for $\Omega = \mathbb{R}^d$ u shall vanish as $\|\mathbf{x}\|$ goes to infinity.

2.2.1 Prerequisites from functional analysis

In order to handle PDEs formally correctly, we need a few prerequisites from functional analysis, as existence and uniqueness results can be formulated elegantly using function spaces.

Vector spaces

Given an $\mathbb{R}$-vector space V, we can define a linear functional from V to the underlying field $\mathbb{R}$.

Definition 2.3 (Linear functional) *Let V be a vector space over $\mathbb{R}$. A* linear functional *L on V is a function $L : V \to \mathbb{R}$, such that for all $u, v \in V$ and for all $\lambda, \mu \in \mathbb{R}$ we have*

$$L(\lambda u + \mu v) = \lambda L(u) + \mu L(v).$$

It is called bounded *if there exists a constant $c \in \mathbb{R}$, such that for all $v \in V$ we have*

$$\|L(v)\|_V \leq c\|v\|_V.$$

The set of all bounded linear functionals on a vector space V is called the *dual space* (of V) and denoted by V^*. The norm of an element $L \in V^*$ is given by

$$\|L\|_{V^*} = \sup_{u \in V} \frac{\|L(v)\|_V}{\|v\|_V}.$$

Definition 2.4 (Bilinear form) *Let V be a vector space over $\mathbb{R}$. A* bilinear form $a(\cdot, \cdot)$ *on V is a function $a : V \times V \to \mathbb{R}$ such that it is linear in each argument, i.e.*

$$a(\lambda u + \mu v, w) = \lambda a(u, w) + \mu a(v, w),$$
$$a(u, \lambda v + \mu w) = \lambda a(u, v) + \mu a(u, w),$$

for all $u, v, w \in V$ and $\lambda, \mu \in \mathbb{R}$. The bilinear form a is called symmetric*, iff for all $u, v \in V$*

$$a(u, v) = a(v, u),$$

it is named positive definite*, iff for all $u \in V, u \neq 0$ we have*

$$a(u, u) > 0.$$

A symmetric and positive definite bilinear form on V is called scalar product*. Each scalar product induces a norm $\|u\|_a := \sqrt{a(u, u)}$ on V. A vector space with scalar product is called* Hilbert space*, as usual.*

Further on, if V is a Hilbert space with induced norm $\|\cdot\|_V$, a bilinear form a is called coercive*, iff*

$$a(u, u) \geq \alpha\|u\|_V^2$$

for all $u \in V$, where $\alpha > 0$.

Using these definitions, the following theorem states an important property of Hilbert spaces:

Theorem 2.1 (Riesz representation theorem) *Let V be a Hilbert space with scalar product $(\cdot, \cdot)_V$ and induced norm $\|\cdot\|_V$. For each bounded linear functional L on V there exists a unique $u \in V$, such that for all $v \in V$ we have*

$$L(v) = (v, u).$$

Moreover, the norm of the operator can be expressed in terms of the norm of this unique representation:

$$\|L\|_{V^*} = \|u\|_V.$$

Proof. See e.g. [89], pp. 90–91. □

This theorem helps us to prove the following lemma.

Lemma 2.1 *Let V be a Hilbert space. Assume that we are given a symmetric coercive bilinear form a and a bounded linear functional L. Then there exists a unique solution $u \in V$ of*

$$a(u, v) = L(u), \text{ for all } v \in V.$$

Proof. From a being symmetric and coercive it follows that a is symmetric and positive definite. So a is a scalar product on V and (V, a) is a Hilbert space. The linear functional L is bounded on (V, a), so we can apply Theorem 2.1 and the assertion holds true. □

Often, a bilinear form is only coercive, but not symmetric. The following theorem can be seen as a generalization of the Riesz representation theorem that covers this case.

Theorem 2.2 (Lax-Milgram Lemma [62]) *Let V be a Hilbert space, let a be a bounded coercive bilinear form and let L be a bounded linear functional. Then there exists a unique vector $u \in V$, such that*

$$a(u, v) = L(u), \text{ for all } v \in V.$$

Proof. See e.g. [89], pp. 92–93. □

Sobolev spaces

Later on we will present a variational approach for the analysis of partial differential equations that is based on the results introduced in the previous section. In this framework the solution of a PDE has to be a member of a function space that is a Hilbert space. The most natural choice for a solution would be a function in $C^k(\Omega)$ which is defined as follows.

Definition 2.5 *The space $C^k(\Omega)$ is the space of functions that are continuous up to at least all derivatives of order k. Depending on the domain Ω and boundary conditions $C_0^k(\Omega)$ denotes either the set of functions that vanish on the boundary if Ω is a proper subset of $\mathbb{R}^d$, the sets of functions that vanish as the norm of the argument goes to infinity if Ω is equal to $\mathbb{R}^d$, or the set of functions f defined on $\Omega = \mathbb{R}^d/\mathbb{Z}^d$ whose integral vanishes, i.e.*

$$\int_\Omega f \, d\mathbf{x} = 0.$$

Obviously, the spaces $C^k(\Omega)$ and the spaces $C_0^k(\Omega)$ form vector spaces. Equipped with the usual norm of $C^k(\Omega)$, namely

$$\|u\|_{C^k(\Omega)} = \sup_{\mathbf{x} \in \Omega} u(\mathbf{x}),$$

they are not Hilbert spaces, since this norm is not induced by a scalar product. So we need function spaces that are Hilbert spaces. The common Hilbert spaces used for the theory of PDEs are the spaces $L_2(\Omega)$, where the norm

$$\|u\|_{L_2(\Omega)} = \int_\Omega |u(\mathbf{x})|^2 d\mathbf{x}$$

is induced by the scalar product

$$(u, v)_{L_2(\Omega)} = \int_\Omega u(\mathbf{x})v(\mathbf{x})d\mathbf{x}.$$

As the functions in L_2 are in general not differentiable, we generalize the notion of partial derivative. Let therefore either $\Omega \subseteq \mathbb{R}^d$ or $\Omega = \mathbb{R}^d/\mathbb{Z}^d$. First we assume that $u \in C^1(\Omega)$, where $\partial\Omega$ has a piecewise smooth boundary, thus the expression

$$\frac{\partial u}{\partial x_i}$$

is meaningful. For all $\varphi \in C_0^1(\Omega)$, applying integration by parts yields

$$\int_\Omega \frac{\partial u}{\partial x_i}\varphi d\mathbf{x} = \int_{\partial\Omega} u\,\varphi\,\vec{\mathbf{n}}_i\,d\mathbf{s} - \int_\Omega u\frac{\partial\varphi}{\partial x_i}d\mathbf{x}, \tag{2.1}$$

where $\vec{\mathbf{n}}$ is the surface normal of $\partial\Omega$. The first summand vanishes in both cases. For $\Omega \subseteq \mathbb{R}^d$, with $\partial\Omega$ being piecewise smooth in the case $\Omega \subsetneq \mathbb{R}^d$, we have that u vanishes at the boundary or in infinity. For $\Omega = \mathbb{R}^d/\mathbb{Z}^d$ the value on the hyperplane of the d-dimensional unit hypercube that represents the torus is equal to the corresponding value on the opposite boundary and the normal $\vec{\mathbf{n}}$ is the same on both planes, except for the sign. So for all $\varphi \in C_0^1(\Omega)$ we have

$$\int_\Omega \frac{\partial u}{\partial x_i}\varphi d\mathbf{x} = -\int_\Omega u\frac{\partial\varphi}{\partial x_i}d\mathbf{x}.$$

This motivates the definition of the *weak derivative* for functions in L_2:

Definition 2.6 (Weak derivative) *Let $\Omega \subseteq \mathbb{R}^d$ or $\Omega = \mathbb{R}^d/\mathbb{Z}^d$ and let $v \in L_2(\Omega)$. The* weak derivative *of v is defined to be the linear functional*

$$\frac{\partial v}{\partial x_i}(\varphi) = -\int_\Omega v\frac{\partial\varphi}{\partial x_i}d\mathbf{x}\,,$$

for $\varphi \in C_0^1(\Omega)$. We define weak derivatives of higher and mixed order accordingly.

That allows the definition of the Sobolev spaces W_p^k:

Definition 2.7 (Sobolev space) *Let $\Omega \subseteq \mathbb{R}^d$ and open or $\Omega = \mathbb{R}^d/\mathbb{Z}^d$. The* Sobolev space *$W_p^k(\Omega)$ is the space of all function which are in $L_p(\Omega)$ and whose partial derivatives ∂^α up to the order $|\alpha| \leq k, \alpha \in \mathbb{N}$ are in $L_p(\Omega)$ as well. The spaces W_p^k have the norms*

$$\|u\|_{W_p^k} := \left(\sum_{|\alpha|=0}^{k} \int_\Omega |\partial^\alpha u|^p d\mathbf{x} \right)^{1/p}$$

and the half-norms, i.e. positive semi-definite linear functionals,

$$\hat{\|}u\hat{\|}_{W_p^k} := \left(\sum_{|\alpha|=k} \int_\Omega |\partial^\alpha u|^p d\mathbf{x} \right)^{1/p}.$$

Additionally, we denote W_2^k by H^k.

As L_2 is complete, the W_2^k are complete as well. So the spaces H^k form Hilbert spaces with the scalar products

$$(u, v)_{H^k} := \sum_{|\alpha|=0}^{k} \int_\Omega \partial^\alpha u \, \partial^\alpha v \, d\mathbf{x}.$$

It remains to note, that if the spaces $W_{p,0}^1(\Omega)$ are defined analogously to the spaces C_0^k, the half-norms $\hat{\|} \cdot \hat{\|}_{W_{p,0}^1(\Omega)}$ are norms, as they map only constant functions to zero and except for the zero function these are not part of the spaces $W_{p,0}^1(\Omega)$.

2.2.2 Prerequisites from Fourier analysis

As mentioned before, partial differential equations with periodic boundary conditions can be analyzed elegantly on the torus $\mathbb{R}^d/\mathbb{Z}^d$. In order to do so, the right hand side and the solution are expanded into their respective Fourier series. In the following the most important definitions and Lemmata from Fourier analysis are repeated, for a detailed introduction we refer to the books of Körner [59] and González-Velasco [43].

Definition 2.8 (Fourier series) *Let $f \in L^2(\mathbb{R}^d/\mathbb{Z}^d)$. Disregarding convergence the 'symbolic' series*

$$\sum_{\mathbf{k} \in \mathbb{Z}^d} \hat{f}(\mathbf{k}) e^{2\pi i \, \mathbf{k} \cdot \mathbf{x}}$$

with

$$\hat{f}(\mathbf{k}) = \int_{\mathbb{R}^d/\mathbb{Z}^d} f(\mathbf{x}) \, e^{-2\pi i \, \mathbf{k} \cdot \mathbf{x}} \, d\mathbf{x}$$

is called the Fourier series *of f, the $\hat{f}$ are called* Fourier coefficients *of f. For ease of notation the operator $\mathcal{F}$ and its inverse $\mathcal{F}^{-1}$ are defined as*

$$\begin{aligned} \mathcal{F} : L^2(\mathbb{R}^d/\mathbb{Z}^d) &\to l^2, \\ f &\mapsto \mathcal{F}(f) := \hat{f}, \\ \mathcal{F}^{-1} : l^2 &\to L^2(\mathbb{R}^d/\mathbb{Z}^d), \\ \hat{f} &\mapsto \mathcal{F}^{-1}(\hat{f}) := f. \end{aligned}$$

The Fourier series can be defined for functions that are not square-integrable, but for the sake of simplicity it is convenient to stick to that space. One of the most important theorems states the connection between square-integrable functions and their Fourier coefficients:

Theorem 2.3 (Riesz-Fischer theorem) *Let $\{\hat{f}(\mathbf{k})\}_{\mathbf{k}}, \mathbf{k} \in \mathbb{Z}^d$ be absolutely square integrable, i.e.*

$$\sum_{\mathbf{k}\in\mathbb{Z}^d} |\hat{f}(\mathbf{k})|^2 < \infty.$$

Then there exists a function $f \in L^2(\mathbb{R}^d/\mathbb{Z}^d)$, whose Fourier coefficients are these $\hat{f}(\mathbf{k})$.

Proof. For $d = 1$ see e.g. Theorem 6.9 in [43]. We can extend the result to $d > 1$ by d-fold application. □

In many cases a stronger concept of convergence is needed. In order to interchange differentiation and summation of a Fourier sum, uniform convergence is necessary.

Lemma 2.2 *Let $f \in C(\mathbb{R}^d/\mathbb{Z}^d)$ and let*

$$\sum_{\mathbf{k}\in\mathbb{Z}^d} |\hat{f}(\mathbf{k})| < \infty.$$

Then the Fourier series of f converges uniformly to f.

Proof. For $d = 1$ the Fourier series converges uniformly by the convergence criterion of Weierstraß, as the series of the Fourier coefficients of f is an absolutely convergent majorant of the Fourier series. Again, we can extend the result to $d > 1$ by repeated application. □

Now we can prove the following lemma that is necessary in order to analyze partial differential equations on the torus.

Lemma 2.3 *Let $f \in L^2(\mathbb{R}^d/\mathbb{Z}^d)$ with absolutely converging Fourier coefficients and let $\partial/\partial x_j f \in L^2(\mathbb{R}^d/\mathbb{Z}^d), j = 1, \ldots, d$. The Fourier series of the partial derivative of f is given by*

$$\frac{\partial}{\partial x_j} f(\mathbf{x}) = \sum_{\mathbf{k}\in\mathbb{Z}^d} 2\pi\, i\, k_j \hat{f}(\mathbf{k})\, e^{2\pi\, i\, \mathbf{k}\cdot\mathbf{x}}.$$

Proof. By Lemma 2.2 the Fourier series of f is uniformly convergent, so it can be differentiated element-wise, yielding the desired result. □

2.2.3 Weak formulation of a PDE

We consider the partial differential equation

$$\mathcal{L}u := -a\Delta u + b\nabla \cdot u + cu = f \text{ in } \Omega \tag{2.2}$$

for the domain and boundary conditions set to either

$$\Omega \subset \mathbb{R}^d \qquad \text{and} \qquad u(\mathbf{x}) = 0 \text{ for all } \mathbf{x} \in \partial\Omega, \tag{2.2a}$$

$$\Omega = \mathbb{R}^d \qquad \text{and} \qquad u(\mathbf{x}) \overset{\|\mathbf{x}\|\to\infty}{\longrightarrow} 0, \tag{2.2b}$$

$$\Omega = \mathbb{R}^d/\mathbb{Z}^d. \tag{2.2c}$$

Let further

$$a(\mathbf{x}) \geq a_0 > 0 \text{ and } c(\mathbf{x}) - \frac{1}{2}\nabla \cdot b(\mathbf{x}) \geq 0 \text{ for all } \mathbf{x} \in \Omega, \tag{2.3}$$

so the PDE is elliptic. Now we derive the variational formulation in the same way as the weak derivative. Under the assumption that the solution u is in $C^2(\Omega)$ we multiply (2.2) by $v \in C_0^1(\Omega)$ and integrate over the whole domain Ω, yielding

$$\int_\Omega \mathcal{L}u\, v\, d\mathbf{x} = \int_\Omega f\, v\, d\mathbf{u} \text{ for all } v \in C_0^1(\Omega).$$

Applying the first Green's identity gives

$$\int_\Omega (a\nabla u \cdot \nabla v + b \cdot \nabla u\, v + c\, u\, v)\, d\mathbf{x} - \int_{\partial\Omega} a\nabla u \cdot \vec{\mathbf{n}}\, v\, d\mathbf{s} = \int_\Omega f\, v\, d\mathbf{x}.$$

Like in (2.1) the boundary integral vanishes, so that

$$\int_\Omega (a\nabla u \cdot \nabla v + b \cdot \nabla u\, v + c\, u\, v)\, d\mathbf{x} = \int_\Omega f\, v\, d\mathbf{x} \text{ for all } v \in C_0^1.$$

As Larsson and Thomée denote that C_0^1 is dense in H_0^1 for sufficiently smooth boundaries (see p. 248 in [61]), the following holds true as well:

$$\int_\Omega (a\nabla u \cdot \nabla v + b \cdot \nabla u\, v + c\, u\, v)\, d\mathbf{x} = \int_\Omega f\, v\, d\mathbf{x} \text{ for all } v \in H_0^1. \tag{2.4}$$

Therefore one defines:

Definition 2.9 (Weak solution) *Let $\mathcal{F}u = 0$ be a partial differential equations in domain Ω with boundary conditions defined as in* (2.2a)*,* (2.2b) *or* (2.2c)*. A function u fulfilling*

$$\int_\Omega \mathcal{F}u\, v\, d\mathbf{x} = 0 \text{ for all } v \in H_0^1(\Omega)$$

is called a weak solution *of the partial differential equation.*

As the derivation of this definition started with a classical solution of the problem, it is clear that a classical solution is always a weak solution. If a weak solution of the model problem, i.e. if $u \in H_0^1(\Omega)$ fulfills (2.4), is in $C_0(\Omega)$ and if the right hand side f is continuous, then this u is a classical solution. This can be seen by applying Green's first identity in the opposite direction:

$$\begin{aligned}
&\int_\Omega (a\nabla u \cdot \nabla v + b \cdot \nabla u\, v + c\, u\, v)\, d\mathbf{x} = \int_\Omega f\, v\, d\mathbf{x} \text{ for all } v \in H_0^1 \\
\Leftrightarrow \quad &\int_\Omega (a\nabla \cdot \nabla u v + b \cdot \nabla u\, v + c\, u\, v)\, d\mathbf{x} = \int_\Omega f\, v\, d\mathbf{x} \text{ for all } v \in H_0^1 \\
\Leftrightarrow \quad &\int_\Omega \mathcal{F}u\, v\, d\mathbf{x} = \int_\Omega f\, v\, d\mathbf{x} \text{ for all } v \in H_0^1 \\
\Leftrightarrow \quad &\int_\Omega (\mathcal{F}u - f)\; v\, d\mathbf{x} = 0 \text{ for all } v \in H_0^1.
\end{aligned}$$

As both $\mathcal{F}u$ and f are continuos functions, it follows, that their difference also vanishes point-wise.

2.2.4 Existence and uniqueness of the weak solution

Having the definition of a weak solution at hand it is possible to show that the model problem in (2.2) has a unique weak solution. To prove that we need the Poincaré inequality.

Theorem 2.4 (Poincaré inequality) *Let $\Omega \subset \mathbb{R}^d$ be an open and bounded domain. Then there exists a constant c such that for all $v \in H_0^1(\Omega)$ we have*

$$\|v\| \leq c\|\nabla v\|.$$

Proof. For the case $\Omega = [0,1] \times [0,1]$ see the proof of Theorem A.6 in [61]. □

Now we are ready to show the main result of this section.

Theorem 2.5 (Existence and uniqueness of the weak solution) *Let $f \in L^2(\Omega)$, Ω as in* (2.2a), (2.2b) *or* (2.2c)*, and let the coefficient functions of* (2.2) *fulfill the requirements in* (2.3)*. Then there exists a unique weak solution $u \in H_0^1(\Omega)$ that accomplishes* (2.2) *with boundary conditions* (2.2a), (2.2b) *or* (2.2c)*. Moreover, there exists a constant C independent of f, such that*

$$\|u\|_{H^1} \leq C\|f\|_{L^2}.$$

Proof. Define a linear functional

$$L(v) = \int_\Omega f\, v\, d\mathbf{x}$$

and a bilinear form

$$g(u,v) = \int_\Omega (a\nabla u \cdot \nabla v + b \cdot \nabla u\, v + c\, u\, v)\, d\mathbf{x}.$$

The linear functional is bounded, as with help of the Cauchy-Schwarz inequality and the Poincaré inequality we show

$$|L(v)| \leq \|f\|_{L^2}\|v\|_{L^2} \leq \|f\|_{L^2}\|v\|_{H^1} \leq c\|f\|_{L^2}\hat{\|}v\hat{\|}_{H_0^1},$$

and $g(\cdot,\cdot)$ is bounded and coercive in $H_0^1(\Omega)$ as for all $v \in H_0^1(\Omega)$

$$\begin{aligned} g(v,v) &= \int_\Omega (a|\nabla v|^2 + (c - \frac{1}{2}\nabla \cdot b)|v|^2)\, d\mathbf{x} \\ &\geq a_0\|v\|^2_{H^1(\Omega)}. \end{aligned}$$

The spaces H^k form Hilbert spaces, so the Lax-Milgram Lemma (Theorem 2.2) is applicable, i.e. the equation

$$g(u,v) = L(v)$$

has a unique solution for each $v \in H_0^1(\Omega)$. □

Now that we know, that a unique weak solution exists, we want to know if the solution is *regular*, i.e. if it depends continuously on the data of the partial differential equation. The answer to this question heavily depends on the domain and on the boundary conditions of the problem at hand, so in the following it will be treated separately for boundary conditions (2.2a), (2.2b) and (2.2c).

2.2.5 Regularity of the solution for PDEs with Dirichlet boundary conditions

Partial differential equations with Dirichlet boundary conditions are very well analyzed. The associated theory requires tools from functional analysis that are beyond the scope of our brief overview, e.g. it depends on Sobolev inequalities and similar estimates. More details can be found in the books of Friedman [36], Gilbarg and Trudinger [41], Gustafson [47], Jost [58] and various other textbooks on PDEs. For the purpose of this work it is sufficient to know about some of the most important results that can be found in the book of Larsson and Thomée [61]. For problem (2.2) with boundary conditions (2.2a) they note in Chapter 3.7 that it is possible to show that for Γ being either smooth or described by finitely many convex piecewise polynomials, a solution u of (2.2) is in H^2, and that a constant c exists such that

$$\|u\|_{H^2} \leq c\|f\|. \tag{2.5}$$

For the plain Poisson equation

$$-\Delta u = f \tag{2.6}$$

this means that the second derivatives of the solution are bounded by a combination of special second derivatives. Another consequence of (2.5) is that small changes in the right hand side lead to relatively small changes in the solution.

Obviously neither being in $H^1(\Omega)$ nor being in $H^2(\Omega)$ is sufficient for applications from engineering or physics. Larsson and Thomée mention that for Γ being smooth and $f \in H^k$ the weak solution u is in $H^{k+2}(\Omega)$. With the Sobolev inequality (see Theorem A.5 in [61]), we obtain $H^{k+2}(\Omega) \subset C_2(\Omega \cup \Gamma)$ for $k > d/2$. That implies that the solution has the desired properties. Similar results can be obtained for domains whose boundaries are convex polynomial and for four-dimensional hypercubes.

2.2.6 Construction of the solution for PDEs with open boundary conditions

For some partial differential equations with open boundary conditions given by (2.2b) it is possible to construct the solution analytically, so a regularity analysis is not necessary. For that purpose, let $\mathcal{L}$ be as in (2.2) with open boundary conditions as in (2.2b) and $b = 0$, so

$$\begin{aligned} -a\Delta u + cu &= f \text{ in } \mathbb{R}^d, \\ u(\mathbf{x}) &\overset{\|\mathbf{x}\|\to\infty}{\longrightarrow} 0. \end{aligned} \tag{2.7}$$

In order to construct a solution for this problem, we need the so-called fundamental solution.

Definition 2.10 (Fundamental solution) *Let $\mathcal{L}$ be defined as* (2.2) *with open boundary conditions given in* (2.2b)*. A function U that fulfills*

$$\int_{\mathbb{R}^d} U\,\mathcal{L}\,\varphi\,d\mathbf{x} = \varphi(\mathbf{0}) \text{ for all } \varphi \in C_0^\infty(\mathbb{R}^d) \tag{2.8}$$

and that is smooth for $\mathbf{x} \neq \mathbf{0}$, having a singularity at $\mathbf{x} = \mathbf{0}$, such that $U \in L^1(B)$ with $B := \{\, \mathbf{x} \in \mathbb{R}^d \,|\, |\mathbf{x}| < 1 \,\}$ and such that

$$|\partial^\alpha U(\mathbf{x})| \leq C_\alpha |\mathbf{x}|^{2-d-|\alpha|}, \text{ for } |\alpha| > 0, \tag{2.9}$$

is called fundamental solution *of $\mathcal{L}$.*

In order to make the purpose of the fundamental solution more obvious, we need the definition of the Dirac delta distribution.

Definition 2.11 (Dirac delta distribution) *Let $\Omega \subset \mathbb{R}^d$. The* Dirac delta distribution *δ is defined to be a linear functional acting on smooth test functions as*

$$\delta(\varphi) = \varphi(\mathbf{0}) \text{ for all } \varphi \in C_0(\Omega).$$

Using δ as the right hand side f in $\mathcal{L}$ it follows that the fundamental solution fulfills

$$\mathcal{L}\,U = \delta$$

in the weak sense. To proceed we need the definition of the convolution of two functions.

Definition 2.12 (Convolution) *Let $f, g \in L^2(\mathbb{R}^d)$. We define the* convolution *of f and g as*

$$(f * g)(\mathbf{x}) := \int_{\mathbb{R}^d} f(\mathbf{x} - \mathbf{y}) g(\mathbf{y}) d\mathbf{y}.$$

Given the fundamental solution and its motivation, we can construct a solution for $\mathcal{L}$ with open boundary conditions as given by the following theorem.

Theorem 2.6 *Let $\mathcal{L}$ be defined as* (2.2) *with open boundary conditions given in* (2.2b)*, U be a fundamental solution and $f \in C_0^1(\mathbb{R}^d)$. Then the unique solution u given by*

$$u(\mathbf{x}) = (U * f)(\mathbf{x}) = \int_{\mathbb{R}^d} U(\mathbf{x} - \mathbf{y})\, f(\mathbf{y})\, d\mathbf{y}.$$

Proof. Due to (2.8) it holds

$$\int_{\mathbb{R}^d} U(\mathbf{x} - \mathbf{y})\, \mathcal{L}\, \varphi(\mathbf{x})\, d\mathbf{x} = \int_{\mathbb{R}^d} U(\mathbf{z})\, \mathcal{L}\, \varphi(\mathbf{z} + \mathbf{y})\, d\mathbf{z} = \varphi(\mathbf{y}).$$

Using an arbitrary test function $\varphi \in C_0^\infty(\mathbb{R}^d)$ the definition of u gives

$$\begin{aligned}\int_{\mathbb{R}^d} u\,\mathcal{L}\,\varphi\,d\mathbf{x} &= \int_{\mathbb{R}^d}\int_{\mathbb{R}^d} U(\mathbf{x}-\mathbf{y})\,f(\mathbf{y})\,d\mathbf{y}\,\mathcal{L}\,\varphi(\mathbf{x})\,d\mathbf{x}\\ &= \int_{\mathbb{R}^d}\int_{\mathbb{R}^d} U(\mathbf{x}-\mathbf{y})\,\mathcal{L}\,\varphi(\mathbf{x})\,d\mathbf{x}\,f(\mathbf{y})\,d\mathbf{y}\\ &= \int_{\mathbb{R}^d} \varphi(\mathbf{y})\,f(\mathbf{y})\,d\mathbf{y}.\end{aligned}$$

Since $\partial/\partial x_i U \in L^1(\mathbb{R}^d)$ and $\partial/\partial x_j f \in C_0(\mathbb{R}^d) \subset L^1(\mathbb{R}^d)$ the Fourier transformations of these functions exist. Thus their convolution can be carried out in Fourier space and the convolution exists. Furthermore

$$\frac{\partial}{\partial x_i}U * \frac{\partial}{\partial x_j}f = \frac{\partial^2}{\partial x_i \partial x_j}(U*f) = \frac{\partial^2}{\partial x_i \partial x_j}u,$$

see e.g. Proposition 1 on page 156 in [89]. Thus all second partial derivatives of u exist, so by partial integration the following holds

$$\int_{\mathbb{R}^d} u\,\mathcal{L}\,\varphi\,d\mathbf{x} = \int_{\mathbb{R}^d} \mathcal{L}\,u\,\varphi\,d\mathbf{x},$$

thus

$$\int_{\mathbb{R}^d} (\mathcal{L}\,u - f)\,\varphi\,d\mathbf{x} = 0,$$

for all $\varphi \in C_0^\infty$. Therefore $\mathcal{L}\,u = f$. □

We can summarize this theorem as follows: Given a fundamental solution of a partial differential equation with open boundary conditions, the classical solution can be constructed for sufficiently smooth right hand sides f vanishing at infinity. As mentioned before, a classical solution is always a weak solution, which is unique. The solution is also regular, as it depends smoothly on the right hand side.

This section closes with the fundamental solution of a particular partial differential equation, the *Green's function* of the Poisson equation in $\mathbb{R}^3$.

Theorem 2.7 *Let* $U : \mathbb{R}^3 \to \mathbb{R}$,

$$U(\mathbf{x}) = \frac{1}{4\pi|\mathbf{x}|}. \tag{2.10}$$

Then U *is a fundamental solution of*

$$\begin{aligned}-\Delta u &= f \text{ in } \mathbb{R}^3,\\ u(\mathbf{x}) &\overset{\|\mathbf{x}\|\to\infty}{\longrightarrow} 0.\end{aligned}$$

Proof. Differentiation of U at $\mathbf{x} \neq \mathbf{0}$ yields

$$\frac{\partial U}{\partial x_i} = -\frac{x_i}{4\pi|\mathbf{x}|^3} \qquad \frac{\partial^2 U}{\partial x_i^2}, = \frac{3x_i^2 - |\mathbf{x}|^2}{4\pi|\mathbf{x}|^5},$$

so $-\Delta U = 0$ for $\mathbf{x} \neq \mathbf{0}$. Equation (2.9) is fulfilled, as $(d/dr)^\alpha 1/r = cr^{-1-\alpha}$. It remains to show, that (2.8) is valid as well. For that purpose let $\varphi \in C_0^\infty(\mathbb{R}^3)$. We set $\mathbf{n} := \mathbf{x}/|\mathbf{x}|$ and apply Green's second identity:

$$\int\limits_{|\mathbf{x}|>\varepsilon} U(-\Delta\varphi)\, d\mathbf{x} = \int\limits_{|\mathbf{x}|>\varepsilon} (-\Delta U)\, \varphi\, d\mathbf{x} - \int\limits_{|\mathbf{x}|=\varepsilon} \left(\varphi \frac{\partial U}{\partial \mathbf{n}} - \frac{\partial \varphi}{\partial \mathbf{n}} U\right) d\mathbf{s}.$$

Now

$$\int\limits_{|\mathbf{x}|>\varepsilon} (-\Delta U)\, \varphi\, d\mathbf{x} = 0,$$

$$\int\limits_{|\mathbf{x}|=\varepsilon} \varphi \frac{\partial U}{\partial \mathbf{n}}\, d\mathbf{s} = \frac{1}{4\pi\varepsilon^2} \int\limits_{|\mathbf{x}|=\varepsilon} \varphi\, d\mathbf{s} \xrightarrow{\varepsilon\to 0} \varphi(\mathbf{0})$$

and

$$\left| \int\limits_{|\mathbf{x}|=\varepsilon} (-\frac{\partial \varphi}{\partial \mathbf{n}} U)\, d\mathbf{s} \right| = \left| \frac{1}{4\pi\varepsilon} \int\limits_{|\mathbf{x}|=\varepsilon} \frac{\partial \varphi}{\partial \mathbf{n}}\, d\mathbf{s} \right| \leq \varepsilon \|\nabla\varphi\|_{C_0(\mathbb{R}^3)} \xrightarrow{\varepsilon\to 0} 0,$$

so

$$\int\limits_{\mathbb{R}^3} U(-\Delta\varphi)\, d\mathbf{x} = \lim_{\varepsilon\to 0} \int\limits_{|\mathbf{x}|>\varepsilon} U(-\Delta\varphi)\, d\mathbf{x} = \varphi(\mathbf{0}).$$

□

2.2.7 Construction of the solution for PDEs on the torus

As for the partial differential equations in the previous section the solution of partial differential equations on the torus is constructed analytically, though the tools needed differ a lot from the ones used previously. We consider the problem (2.2) with boundary conditions (2.2c), i.e.

$$-a\Delta u + b\nabla \cdot u + cu = f \text{ in } \mathbb{R}^d/\mathbb{Z}^d,$$

with constant coefficients $a, b, c \in \mathbb{R}$ and with $f \in L^2(\mathbb{R}^d/\mathbb{Z}^d)$. The solution of this partial differential equation can be given in terms of its Fourier series as stated in the following theorem.

Theorem 2.8 *Let*

$$-a\Delta u + b\nabla \cdot u + cu = f \text{ in } \mathbb{R}^d/\mathbb{Z}^d,$$

with $f \in L^2(\mathbb{R}^d/\mathbb{Z}^d)$ be a given partial differential equation and let $\hat{f}(\mathbf{k})$ be the Fourier coefficients of the right hand side with

$$\sum_{\mathbf{k}\in\mathbb{Z}^d} |\hat{f}(\mathbf{k})| < \infty.$$

Assuming that either $c \neq 0$ or that the Fourier coefficient $\hat{f}(\mathbf{0})$ vanishes, the solution u can be given in terms of its uniformly convergent Fourier series as

$$u(\mathbf{x}) = \sum_{\mathbf{k}\in\mathbb{Z}^d} \hat{u}(\mathbf{k}) e^{-2\pi\, i\, \mathbf{k}\cdot\mathbf{x}},$$

where

$$\hat{u}(\mathbf{k}) = \frac{\hat{f}(\mathbf{k})}{\sum\limits_{j=1}^{d} a\, k_j^2 + b\, i\, k_j + c}.$$

Proof. The series

$$\sum_{\mathbf{k}\in\mathbb{Z}^d} \hat{u}(\mathbf{k}) e^{-2\pi\, i\, \mathbf{k}\cdot\mathbf{x}}$$

has a convergent majorant series, as

$$|\hat{u}(\mathbf{k})| = \left| \frac{\hat{f}(\mathbf{k})}{\sum\limits_{j=1}^{d} a\, k_j^2 + b\, i\, k_j + c} \right| < c_0 \left| \hat{f}(\mathbf{k}) \right|, \ \forall\, \mathbf{k} \neq \mathbf{0}.$$

Therefore it converges uniformly to u by the convergence criterion of Weierstraß. That allows to analyze the partial differential equation by developing both sides of the equation into the respective Fourier series:

$$\begin{aligned} & \mathcal{F}(-a\Delta u + b\nabla \cdot u + cu)(\mathbf{x}) = \mathcal{F}(f)(\mathbf{x}) \\ \Leftrightarrow \qquad & \sum_{\mathbf{k}\in\mathbb{Z}^d} \widehat{(-a\Delta u + b\nabla \cdot u + cu)}(\mathbf{k}) e^{2\pi\, i\, \mathbf{k}\cdot\mathbf{x}} = \sum_{\mathbf{k}\in\mathbb{Z}^d} \hat{f}(\mathbf{k}) e^{2\pi\, i\, \mathbf{k}\cdot\mathbf{x}}. \end{aligned}$$

Applying Lemma 2.3 twice gives

$$\sum_{\mathbf{k}\in\mathbb{Z}^d} \sum_{j=1}^{d} (a\, k_j^2 + b\, i\, k_j + c)\, \hat{u}(\mathbf{k}) e^{2\pi\, i\, \mathbf{k}\cdot\mathbf{x}} = \sum_{\mathbf{k}\in\mathbb{Z}^d} \hat{f}(\mathbf{k}) e^{2\pi\, i\, \mathbf{k}\cdot\mathbf{x}}.$$

Comparison of coefficients yields:

$$\hat{u}(\mathbf{k}) = \frac{\hat{f}(\mathbf{k})}{\sum_{j=1}^{d} a\, k_j^2 + b\, i\, k_j + c}.$$

Now we have to consider two cases: If $c \neq 0$, this equation is always true. For $c = 0$, we need more, namely $\hat{f}(\mathbf{0}) = \mathbf{0}$, required in the assumptions. □

So under the premises of the previous theorem the classical solution of the partial differential equation can be constructed. It remains to mention that the constructed solution is also regular, as the dependence on the right hand side of the PDE is smooth.

2.3 Numerical solution

We have shown that unique solutions of a PDE of the form (2.2) with Dirichlet boundary conditions (2.2a) or periodic boundary conditions (2.2c) and of a PDE with open boundary conditions as given in (2.7) exist. Furthermore they are regular, so a numerical approximation of the solution is meaningful. Various different methods for the numerical solution of partial differential equations exist. In the following we will examine two methods in larger detail. The first method under investigation will be the discretization using finite differences which is probably the easiest method for the numerical solution of PDEs. After that we will discuss the discretization of PDEs using finite volumes. The first method is perfectly suited for simply shaped domains like cuboids with either Dirichlet or periodic boundary conditions, where in the periodic case the cuboid is just a representative of the torus. When the partial differential equation at hand has constant coefficients, the resulting linear systems are easy to analyze, for details we refer to Chapter 3. The second discretization technique is especially well suited for the numerical solution of the Poisson equation with open boundary conditions. Together with the method an extension of the error analysis of Washio and Oosterlee [87] is presented here.

Various other discretization and solution techniques for PDEs exist, that we do not mention here. One of the most important techniques missing in this work is the finite element method, which is strongly connected to the variational approach that was presented in Section 2.2.3.

2.3.1 Solution of PDEs on the torus or on subsets of $\mathbb{R}^d$ with Dirichlet boundary conditions using finite differences

The use of finite differences for the solution of partial differential equations is straightforward as it is directly connected to the definition of the derivative. To motivate the use of

finite differences for the solution of partial differential equations, we will start with one dimension.

Finite differences in one dimension

The derivative of a function is defined via the difference quotient

$$f'(x) := \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}.$$

Motivated by this definition the discretization of a derivative on an equispaced grid with grid width h can be given by

$$f'(x) \doteq \frac{f(x+h) - f(x)}{h}.$$

Using the Taylor expansion it can be shown that the error involved is of order h, as

$$\begin{aligned} & f(x+h) = f(x) + f'(x)\, h + \mathcal{O}(h^2) \\ \Leftrightarrow \qquad & f'(x) = \frac{f(x+h) - f(x)}{h} + \mathcal{O}(h). \end{aligned}$$

Using Taylor expansion at additional grid points, e.g.

$$f(x+h) = f(x) + f'(x)\, h + \frac{f''(x)}{2}\, h^2 + \frac{f'''(x)}{6}\, h^3 + \mathcal{O}(h^4) \tag{2.11}$$

and

$$f(x-h) = f(x) - f'(x)\, h + \frac{f''(x)}{2}\, h^2 - \frac{f'''(x)}{6}\, h^3 + \mathcal{O}(h^4), \tag{2.12}$$

allows the definition of higher-order approximations of the first derivative, e.g. by subtracting (2.12) from (2.11) and dividing the result by 2 we have

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} + \mathcal{O}(h^2), \tag{2.13}$$

and of approximations of higher derivatives, e.g. the order h^2-approximation of the second derivative given by adding (2.11) and (2.11) given by

$$f''(x) = \frac{f(x-h) - 2\, f(x) + f(x+h)}{h^2} + \mathcal{O}(h^2). \tag{2.14}$$

Higher order approximations can be constructed by using more grid points, e.g. not only $x-h$, x and $x+h$, but $x-2h$, $x+2h$, ...

For the one-dimensional analogue of the Poisson equation with Dirichlet boundary conditions, i.e.

$$-u''(x) = f(x) \text{ for all} x \in \Omega, \qquad u(0) = g_0, \qquad u(1) = g_1,$$

discretization with the approximation in (2.14) with $u_i = u(ih)$, $f_i = f(ih)$ and $h = 1/n$ leads to the linear system

$$\begin{aligned} u_0 &= g0, \\ \frac{1}{h^2}\left(u_{i-1} - 2\,u_i + u_{i+1}\right) &= f_i, \text{ for } i = 1, \ldots, n-1, \\ u_n &= g1. \end{aligned}$$

After elimination of the boundary values this linear system leads to a tridiagonal linear system. Analogously, for periodic boundary conditions we get

$$\begin{aligned} \frac{1}{h^2}\left(u_n - 2\,u_0 + u_1\right) &= f_0, \\ \frac{1}{h^2}\left(u_{i-1} - 2\,u_i + u_{i+1}\right) &= f_i, \text{ for } i = 1, \ldots, n-1 \\ \frac{1}{h^2}\left(u_{n-1} - 2\,u_n + u_0\right) &= f_n. \end{aligned}$$

The resulting system has a singular coefficient matrix that is circulant. Both systems can be solved using multigrid methods. This will be described in Chapter 3.

Finite differences for higher dimensions and the stencil notation

The usage of finite differences for the approximation of derivatives is not limited to one dimension but can easily be extended to more dimensions. The occurring partial derivatives are approximated as before, yielding a linear system that has to be solved in order to obtain the approximate solution of the partial differential equation. So for a second order accurate approximation of the Laplacian in two dimensions we combine (2.14) in x_1- and x_2-direction and obtain

$$\begin{aligned} \Delta u(\mathbf{x}) = \frac{1}{h^2}\,[&u(\mathbf{x} - h\mathbf{e}_1) + u(\mathbf{x} - h\mathbf{e}_2) - 4u(\mathbf{x}) \\ &+ u(\mathbf{x} + h\mathbf{e}_1) + u(\mathbf{x} + h\mathbf{e}_2)] + \mathcal{O}(h^2), \end{aligned} \tag{2.15}$$

where $\mathbf{e}_i$ is the i-th unit vector. For the sake of clarity we introduce the *stencil notation*. In this notation the coefficients belonging to neighboring grid points are written in squared brackets, where the coefficient in the center is belonging to the actual grid point itself. The stencil for (2.15) is

$$\frac{1}{h^2}\begin{bmatrix} & 1 & \\ 1 & -4 & 1 \\ & 1 & \end{bmatrix}. \tag{2.16}$$

The same can be used for higher dimensions, e.g. an approximation of the Laplacian in three dimensions is given by

$$\begin{aligned}\Delta u(\mathbf{x}) = \frac{1}{h^2}\left[u(\mathbf{x}-h\mathbf{e}_1)+u(\mathbf{x}-h\mathbf{e}_2)+u(\mathbf{x}-h\mathbf{e}_3)-6u(\mathbf{x}\right.\\ \left.+u(\mathbf{x}+h\mathbf{e}_1)+u(\mathbf{x}+h\mathbf{e}_2)+u(\mathbf{x}+h\mathbf{e}_3)\right]+\mathcal{O}(h^2),\end{aligned} \tag{2.17}$$

or in stencil notation by

$$\frac{1}{h^2}\begin{bmatrix} & 1 & \\ & & \end{bmatrix} \qquad \frac{1}{h^2}\begin{bmatrix} & 1 & \\ 1 & -6 & 1 \\ & 1 & \end{bmatrix} \qquad \frac{1}{h^2}\begin{bmatrix} & 1 & \\ & & \end{bmatrix}.$$

To simplify the representation we write $u_{i,j,k}$ and $f_{i,j,k}$ to denote the value of u respectively f at the grid point $\mathbf{x}_{i,j,k}$, and we introduce the notation $\partial^2_{x_1} u_{i,j,k}$, which is defined as the central finite difference approximation to the second partial derivative in x-direction, i.e.

$$\partial^2_{x_1} u_{i,j,k} := \frac{u(\mathbf{x}_{i-1,j,k}) - 2u(\mathbf{x}_{i,j,k}) + u(\mathbf{x}_{i+1,j,k})}{h^2}. \tag{2.18}$$

Here $h = \|\mathbf{x}_{i,j,k} - \mathbf{x}_{i-1,j,k}\|_2 = \|\mathbf{x}_{i+1,j,k} - \mathbf{x}_{i,j,k}\|_2$. We define $\partial^2_{x_2} u_{i,j,k}$ and $\partial^2_{x_3} u_{i,j,k}$ in the same manner, so we may write for (2.17):

$$\Delta u_{i,j,k} = \partial^2_{x_1} u_{i,j,k} + \partial^2_{x_2} u_{i,j,k} + \partial^2_{x_3} u_{i,j,k} + \mathcal{O}(h^2).$$

If one orders variables lexicographically, the linear systems that belong to these discretizations are blocked systems, where the occurring blocks can be derived directly from the stencil. So in two dimensions the three diagonals of the block on the main diagonal are given by the row in the center of the stencil and the diagonal entries of the blocks on the secondary diagonals are given by the lower row for the lower diagonal block respectively by the upper row for the upper diagonal block.

Compact discretizations of higher order

We will now continue with compact discretizations of higher order, i.e. discretizations not only taking into account the direct neighbors, but all nearest neighbors. These discretizations are often referred to as *compact* discretizations, as the stencil describing them still has the compact 9-point representation in two dimensions, respectively a 27-point stencil in 3D. Nevertheless, the stencil has more non-zero entries than the original stencil of the discretization of order h^2. The main advantage of these stencils is that they achieve higher order but still only nearest neighbors are needed. This is a nice property especially when considering PDEs with Dirichlet boundary conditions, as the nearest neighbors are always available, which might not be the case for the next layer. Another advantage is the reduced

amount of communication for parallel solvers, which use ghost cells. The approach presented here can be found in the work of Spotz and Carey, who derived the discretization in [75].

To define compact schemes of higher order, we now take a closer look at the error term in (2.17), while still using the notation as in (2.18). For the Poisson equation

$$-\Delta u = f, \tag{2.19}$$

we get

$$\partial_x^2 u_{i,j,k} + \partial_y^2 u_{i,j,k} + \partial_z^2 u_{i,j,k} - \tau_{i,j,k} = f_{i,j,k},$$

with

$$\tau_{i,j,k} = \frac{h^2}{12}\left[\frac{\partial^4 u}{\partial x_1^4} + \frac{\partial^4 u}{\partial x_2^4} + \frac{\partial^4 u}{\partial x_3^4}\right]_{i,j,k} + \frac{h^4}{360}\left[\frac{\partial^6 u}{\partial x_1^6} + \frac{\partial^6 u}{\partial x_2^6} + \frac{\partial^6 u}{\partial x_3^6}\right]_{i,j,k} + \mathcal{O}(h^6). \tag{2.20}$$

Taking the appropriate partial derivatives of (2.19) we get

$$\begin{aligned}
\frac{\partial^4 u}{\partial x_1^4} &= \frac{\partial^2 f}{\partial x_1^2} - \frac{\partial^4 u}{\partial x_1^2 \partial x_2^2} - \frac{\partial^4 u}{\partial x_1^2 \partial x_3^2},\\
\frac{\partial^4 u}{\partial x_2^4} &= \frac{\partial^2 f}{\partial x_2^2} - \frac{\partial^4 u}{\partial x_1^2 \partial x_2^2} - \frac{\partial^4 u}{\partial x_2^2 \partial x_3^2},\\
\frac{\partial^4 u}{\partial x_3^4} &= \frac{\partial^2 f}{\partial x_3^2} - \frac{\partial^4 u}{\partial x_1^2 \partial x_3^2} - \frac{\partial^4 u}{\partial x_2^2 \partial x_3^2}.
\end{aligned}$$

When we substitute these into (2.20), we obtain

$$\begin{aligned}
\tau_{i,j,k} = \frac{h^2}{12}\Delta f_{i,j,k} - \frac{h^2}{6}&\left[\frac{\partial^4 u}{\partial x_1^2 \partial x_2^2} + \frac{\partial^4 u}{\partial x_1^2 \partial x_3^2} + \frac{\partial^4 u}{\partial x_2^2 \partial x_3^2}\right]_{i,j,k}\\
&+ \frac{h^4}{360}\left[\frac{\partial^6 u}{\partial x_1^6} + \frac{\partial^6 u}{\partial x_2^6} + \frac{\partial^6 u}{\partial x_3^6}\right]_{i,j,k} + \mathcal{O}(h^6).
\end{aligned}$$

Now for all terms that are multiplied by h^2 and thus contribute to our h^2 error term, we are able to provide h^2-accurate approximations. Thus the resulting approximation to (2.19) is given by

$$\begin{aligned}
\left[\partial_{x_1}^2 + \partial_{x_2}^2 + \partial_{x_3}^2 + \frac{h^2}{6}(\partial_{x_1}^2\partial_{x_2}^2 + \partial_{x_1}^2\partial_{x_3}^2 + \partial_{x_2}^2\partial_{x_3}^2)\right] u_{i,j,k} =&\\
f_{i,j,k} + \frac{h^2}{12}\left[\partial_{x_1}^2 + \partial_{x_2}^2 + \partial_{x_3}^2\right] f_{i,j,k} + \mathcal{O}(h^4)&
\end{aligned} \tag{2.21}$$

and is h^4-accurate.

Provided that the analytical derivatives of the right hand side f are available, Spotz and Carey derived h^6-accurate approximations in the same manner. For details, we refer to [75]. Their work was recently reviewed and extended by Sutmann and Steffen in [83].

2.3.2 Finite volume discretization-based solution of PDEs defined on $\mathbb{R}^d$

While finite difference methods are easy to understand and to implement for standard geometries leading to equispaced grids, they are hard to deal within the case of unstructured grids as they occur in many engineering applications. One option to avoid the problems related to the use of finite difference methods is the finite volume method.

Finite volume discretization

The purpose of the finite volume method is the same as that of the finite differences, i.e. discretizing a PDE in order to gain a solution of it at defined points, but the derivation of the methods is completely different. Whereas in the finite difference method we started with the discrete points and discretizations of the occurring partial derivatives directly yielding the algebraic equations, in the finite volume method the domain is partitioned into several small volumes and the PDE is rewritten at the interior of these volumes using the divergence theorem. This is a common approach for hyperbolic PDEs, but it is feasible for the solution of the Poisson equation in free space, as well.

For this purpose we consider $\mathcal{L}$ as in (2.2) with $b = -\nabla a$, i.e.

$$-a(\mathbf{x})\Delta u(\mathbf{x}) - \nabla a(\mathbf{x}) \cdot \nabla u(\mathbf{x}) + c(\mathbf{x})u(\mathbf{x}) = f(\mathbf{x}) \text{ for } \mathbf{x} \in \Omega \tag{2.22}$$

with boundary conditions as in (2.2a), (2.2b) or (2.2c). Now we may write

$$-a(\mathbf{x})\Delta u(\mathbf{x}) - \nabla a(\mathbf{x}) \cdot \nabla u(\mathbf{x}) + c(\mathbf{x})u(\mathbf{x}) = -\nabla \cdot (a(\mathbf{x})\nabla u(\mathbf{x})) + c(\mathbf{x})u(\mathbf{x}),$$

yielding (2.22) in *divergence form*

$$-\nabla \cdot (a(\mathbf{x})\nabla u(\mathbf{x})) + c(\mathbf{x})u(\mathbf{x}) = f(\mathbf{x}) \text{ for } \mathbf{x} \in \Omega. \tag{2.23}$$

The domain Ω is partitioned into smaller closed volumes $v_i, i = 1, \ldots, n$, such that

$$\bigcup_{i=1,\ldots,n} v_i = \Omega, \tag{2.24}$$

while

$$v_i \cap v_j = \emptyset \text{ for all } i \neq j. \tag{2.25}$$

By V we denote this partitioning of Ω. For each subvolume $v_i \subset \Omega, i = 1, \ldots, n$ the following holds true

$$\int_{v_i} -\nabla \cdot (a(\mathbf{x})\nabla u(\mathbf{x})) + c(\mathbf{x})u(\mathbf{x})d\mathbf{x} = \int_{v_i} f(\mathbf{x})d\mathbf{x}.$$

Applying Gauß' divergence theorem yields

$$\int_{\partial v_i} -(a(\mathbf{s})\nabla u(\mathbf{s})) \cdot \vec{\mathbf{n}}\, d\mathbf{s} + \int_{v_i} c(\mathbf{x})u(\mathbf{x})d\mathbf{x} = \int_{v_i} f(\mathbf{x})d\mathbf{x}, \tag{2.26}$$

where $\vec{\mathbf{n}}$ is the outer normal of ∂v_i. On the basis of this equation and with the help of finite difference approximations of the gradient a proper discretization of the partial differential equation can be given for domain Ω. The gradient in the boundary integral in (2.26) is called the *flux*. The flux out of one subvolume over the boundary to a neighboring subvolume is equal to the flux over this boundary into that subvolume. This is true for a symmetric discretization of the gradients as well, which by this observation is conservative.

Consider the simple case of equation (2.22) in $d = 2$ dimensions with Dirichlet boundary on the unit square, i.e. $\Omega = [0, 1]^2$. For $h = 1/n$ we define the partitioning

$$V_h = \{v_{h_{i,j}} | v_{h_{i,j}} = [(i-1)h, ih] \times [(j-1)h, jh], i, j = 1, \ldots, n\}.$$

This partitioning fulfills (2.24) and (2.25). We discretize the boundary integral by the value of the gradient in the middle of one side times its length, i.e.

$$\begin{aligned}\int_{\partial v_{h_{i,j}}} (a(\mathbf{s})\nabla u(\mathbf{s})) \cdot \vec{n}\, d\mathbf{s} \doteq h \Bigg(& a((i-1)h, (j-\tfrac{1}{2})h)\frac{\partial u((i-1)h, (j-\frac{1}{2})h)}{\partial x_1} \\ & - a(ih, (j-\tfrac{1}{2})h)\frac{\partial u(ih, (j-\frac{1}{2})h)}{\partial x_1} \\ & + a((i-\tfrac{1}{2})h, (j-1)h)\frac{\partial u((i-\frac{1}{2})h, (j-1)h)}{\partial x_2} \\ & - a((i-\tfrac{1}{2})h, jh)\frac{\partial u((i-\frac{1}{2})h, jh)}{\partial x_2} \Bigg),\end{aligned}$$

and the volume integrals by the value of u and f at the center times the volume, i.e.

$$\int_{v_{h_{i,j}}} c(\mathbf{x})u(\mathbf{x})d\mathbf{x} \doteq h^2 c((i-\tfrac{1}{2})h, (j-\tfrac{1}{2})h)u((i-\tfrac{1}{2})h, (j-\tfrac{1}{2})h)$$

and

$$\int_{v_{h_{i,j}}} f(\mathbf{x}) \doteq h^2 f((i-\tfrac{1}{2})h, (j-\tfrac{1}{2})h).$$

Both quadrature formulas are order h^2 accurate. If we discretize the partial derivatives

using the second order accurate discretization in (2.13), i.e.

$$\begin{aligned}
\frac{\partial u((i-1)h,(j-\frac{1}{2})h)}{\partial x_1} &\doteq \frac{u_{i,j}-u_{i-1,j}}{h},\\
\frac{\partial u(ih,(j-\frac{1}{2})h)}{\partial x_1} &\doteq \frac{u_{i+1,j}-u_{i,j}}{h},\\
\frac{\partial u((i-\frac{1}{2})h,(j-1)h)}{\partial x_2} &\doteq \frac{u_{i,j}-u_{i,j-1}}{h},\\
\text{and} \qquad \frac{\partial u((i-\frac{1}{2})h,jh)}{\partial x_2} &\doteq \frac{u_{i,j+1}-u_{i,j}}{h},
\end{aligned}$$

where $u_{i,j} = u((i-1/2)h,(j-1/2)h)$, for $i,j = 1,\ldots,n$, we obtain for the subvolume centered around $((i-1/2)h,(j-1/2)h)$:

$$(h^2 c_{i,j} - 4)u_{i,j} + u_{i-1,j} + u_{i,j-1} + u_{i,j+1} + u_{i+1,j} = h^2 f_{i,j}. \tag{2.27}$$

Except for the boundary conditions, which either have to be given in terms of the values of u at a distance of $\frac{h}{2}$ away from the boundary or in terms of the normal derivative of u, this yields the same system as the discretization using finite differences. The same is true for higher dimensions. As both the quadrature formulae and the approximation of the first derivatives are second order accurate, the overall accuracy of this method is of order $\mathcal{O}(h^2)$, higher order quadrature formulae and partial derivative discretization can be used yielding higher accuracy. The main advantage of the finite volume method over the finite differences discretization is the potential to discretize a partial differential equation in irregular domains or adaptively as it depends on approximating the flow between two volumes and discretizing the integral over the right hand side, only.

Washio's and Oosterlee's finite volume discretization of the Poisson equation on $\mathbb{R}^d$

In the following we will derive an adaptive discretization for the solution of the Poisson equation with open boundary conditions that is based on a work of Washio and Oosterlee [87]. The following covers the case that the solution of

$$\begin{aligned}
-\Delta u(\mathbf{x}) &= f(\mathbf{x}),\ \mathbf{x} \in \mathbb{R}^3\\
u(\mathbf{x}) &\to 0,\ ||\mathbf{x}|| \to \infty,
\end{aligned} \tag{2.28}$$

is sought for in $\Omega_0 = [-\frac{1}{2},\frac{1}{2}]^3$, only, where supp$(f) \subset \Omega_0$. To solve this problem numerically, we discretize Ω_0 using a regular grid with mesh-width h and Δ using finite volumes, i.e. the 3D analogue of (2.27).

To properly handle the boundary conditions the original grid is extended with the help of a grid extension rate $\alpha \in (1,2)$ in the following way: The grid on the finest level is defined

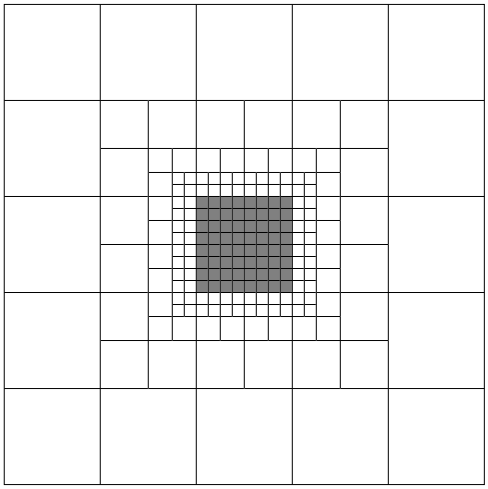

Figure 2.1: Coarsened grid in 2D. Highlighted is the original fine grid, in which the solution is of interest.

to be the discretization of domain Ω_1 with grid-width h_1, where

$$\begin{aligned} \Omega_1 &:= \left[-\frac{\beta_1}{2}, \frac{\beta_1}{2}\right]^3, \\ \beta_1 &\geq \alpha, \\ h_1 &:= h. \end{aligned} \tag{2.29}$$

As a result Ω_1 is just an extension of the original domain Ω. The domain is then extended and the grid is coarsened recursively as

$$\begin{aligned} \Omega_l &:= \left[-\frac{\beta_l}{2}, \frac{\beta_l}{2}\right]^3, \\ \beta_l &\geq \alpha^l, \\ h_l &:= 2^{(l-1)}\, h. \end{aligned} \tag{2.30}$$

The additional parameters β_l are introduced in order to enable the extended grids to have common grid points with the fine grids. Furthermore we define the set of grid points $\mathcal{G}_l$ of level l to be

$$\mathcal{G}_l := \{\, \mathbf{x} \in \Omega_l \mid \mathbf{x} = h_l \mathbf{z}, \mathbf{z} \in \mathbb{Z} \,\}.$$

An example of how a coarsened grid might look like in 2D can be found in Figure 2.1. We remark that Washio and Oosterlee continue the extension and coarsening process up to infinity, which is nice for the analysis of the discretization but not suitable for an actual implementation.

The Laplacian is now discretized on the domains $\Omega_1, \Omega_2 \backslash \Omega_1, \Omega_3 \backslash \Omega_2, \ldots$ using the finite volume method, except for the interfaces. For a complete discretization of $\mathbb{R}^3$ we have to

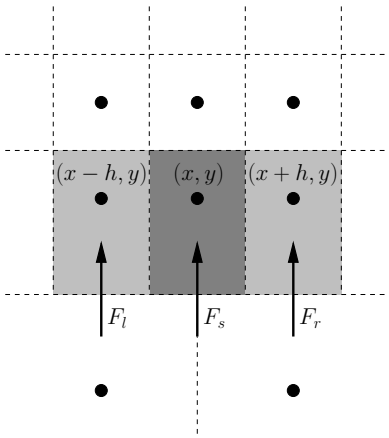

Figure 2.2: Conservative discretization at the interface in 2D.

give a discretization of the problem on the composite grid

$$\mathcal{G} := \mathcal{G}_1 \cup \mathcal{G}_2 \cup \cdots ,$$

including the interfaces between Ω_l and $\Omega_{l+1} \backslash \Omega_l$. Using the finite volume discretization this can be done relatively straightforwardly. As an example consider the two-dimensional discretization using finite volumes at the refinement boundary that is depicted in Fig. 2.2 (the extension to 3D is straightforward). Here the flux F_s can be approximated by interpolating linearly from the left and the right neighbors, i.e.

$$F_s = \frac{1}{2}(F_l + F_r). \tag{2.31}$$

Now we are ready to show that for a suitable grid-extension rate α the error of this method is of the same order as it would be if the whole grid was discretized using the finest grid size.

In order to analyze the error of such a discretization we define the discrete analogue of a Green's function.

Definition 2.13 (Discrete Green's function) *Let Δ_h be a discretization of the Laplace operator on the grid $\{ \mathbf{x} \mid \mathbf{x} = h\, \mathbf{z}, \mathbf{z} \in \mathbb{Z}^3 \}$ and let $\delta_h(\mathbf{x}, \mathbf{y})$ be defined as*

$$\delta_h(\mathbf{x}, \mathbf{y}) := \begin{cases} 1, & \mathbf{x} = \mathbf{y}, \\ 0 & \textit{otherwise}. \end{cases}$$

Then the discrete Green's function is defined by

$$\Delta_h G_h(\mathbf{x}, \mathbf{y}) := \delta_h(\mathbf{x}, \mathbf{y}),$$

where Δ_h is w.r.t. the first argument $\mathbf{x}$, only.

Conforming to the discretization using the partitioning of the domain, we measure the error in terms of the difference to the cell-average of the analytic Green's function, which is derived in Theorem 2.7.

Definition 2.14 (Cell-averaged Green's function) *Let $G(\mathbf{x}, \mathbf{y})$ be the Green's function of the Laplace operator Δ and let $\Omega_{\mathbf{x}}$ be defined as the cube with volume h^3 centered at $\mathbf{x}$, i.e.*

$$\Omega_{\mathbf{x}} := \left\{ \mathbf{y} \;\middle|\; ||\mathbf{x} - \mathbf{y}||_\infty \leq \frac{h}{2} \right\}.$$

The cell-averaged Green's function $\tilde{G}$ is given by

$$\tilde{G}(\mathbf{x}, \mathbf{y}) = \frac{1}{h^3} \int\limits_{\Omega_{\mathbf{x}}} G(\mathbf{z}, \mathbf{y}) d\mathbf{z}.$$

As we chose a conservative discretization, Green's identity holds for the discrete case as well. Thus we obtain

$$\int\limits_{\Omega} [u(\Delta_h v) - (\Delta_h u)v] d\mathbf{x} = \oint\limits_{\partial\Omega} [u(\nabla_h v) - (\nabla_h u)v] \cdot \vec{\mathbf{n}} d\mathbf{s}.$$

Therefore, for the discrete Green's function G_h it holds true that

$$\int\limits_{\Omega} G_h(\mathbf{x}, \mathbf{y})[-\Delta\Psi_h(\mathbf{y})] d\mathbf{y} = \Psi_h(\mathbf{x}) - \oint\limits_{\partial\Omega} [G_h(\mathbf{x}, \mathbf{s})(\nabla_h \Psi_h(\mathbf{s})) - (\nabla_h G_h(\mathbf{x}, \mathbf{s}))\Psi_h(\mathbf{s})] \cdot \vec{\mathbf{n}} d\mathbf{s}, \quad (2.32)$$

for a function Ψ_h. With this observation, we are now ready to provide an error analysis for Washio's and Oosterlee's method.

Theorem 2.9 *Using the described grid coarsening strategy with a grid extension rate $\alpha \geq 2^{2/3}$ the error $e(\mathbf{x}, \mathbf{p})$, defined as*

$$e(\mathbf{x}, \mathbf{p}) := |\tilde{G}(\mathbf{x}, \mathbf{p}) - G_h(\mathbf{x}, \mathbf{p})|, \quad (2.33)$$

is of order h^2 for all $\mathbf{x} \in \mathcal{G}_0$.

Proof. By applying the discrete version of Green's identity, i.e. inserting $e(\mathbf{x}, \mathbf{p})$ into (2.32),

we get

$$e(\mathbf{x},\mathbf{p}) = \Bigg| \int\limits_{\Omega_l} G_h(\mathbf{x},\mathbf{y})[-\Delta_h(\tilde{G}(\mathbf{y},\mathbf{p}) - G_h(\mathbf{y},\mathbf{p}))]\, d\mathbf{y} + \oint\limits_{\partial\Omega_l} \Big[G_h(\mathbf{x},\mathbf{s})\nabla_h(\tilde{G}(\mathbf{s},\mathbf{p}) - G_h(\mathbf{s},\mathbf{p})) - \nabla_h G_h(\mathbf{x},\mathbf{s})(\tilde{G}(\mathbf{s},\mathbf{p}) - G_h(\mathbf{s},\mathbf{p}))\Big] \cdot \vec{\mathbf{n}} d\mathbf{s}\Bigg| \quad (2.34)$$

for any domain Ω_l. Letting $l \to \infty$ the second integral vanishes due to the boundary conditions. So the error due to the integration over the finest grid is bounded by

$$|e_0(\mathbf{x},\mathbf{p})| \leq c_0 h^2.$$

Washio and Oosterlee showed in [87] that the error e_1 due to the region outside the finest grid, but not including the non-cubic-cells is bounded by

$$|e_1(\mathbf{x},\mathbf{p})| \leq c_1 \frac{\alpha^3 - 1}{1 - 2^2/\alpha^3} \frac{h^2}{d_{\mathbf{x}} d_{\mathbf{p}}^5}$$

and that the error e_2 due to the non-cubic cells is bounded by

$$|e_2(\mathbf{x},\mathbf{p})| \leq c_2 \frac{1}{1 - 2^2/\alpha^3} \frac{h^2}{d_{\mathbf{x}} d_{\mathbf{x}}^4},$$

where $d_{\mathbf{x}}$ and $d_{\mathbf{p}}$ are the minimum distances from the boundary of the finest grid of $\mathbf{x}$ and $\mathbf{p}$, respectively. The proof depends on the fact that there exist constants $c_k, (k = 0, 1, 2, \dots)$, such that

$$|\Delta_y^{k-m} \Delta_p^m G(\mathbf{y},\mathbf{p})| \leq \frac{c_k}{|\mathbf{y} - \mathbf{p}|^{k+1}},\ (m \leq k),$$

where Δ_y and Δ_p act on $\mathbf{y}$ and $\mathbf{p}$, respectively. For further details we refer to [87]. Overall, the first integral can thus be estimated as:

$$\left| \int\limits_{\Omega_l} G_h(\mathbf{x},\mathbf{y})[-\Delta_h(\tilde{G}(\mathbf{y},\mathbf{p}) - G_h(\mathbf{y},\mathbf{p}))]\, d\mathbf{y} \right| \leq e_0 + e_1 + e_2 = \mathcal{O}(h^2).$$

The limit $l \to \infty$ yields the desired result. □

Modification of Washio's and Oosterlee's method

Although the proposed method of Washio and Oosterlee is of the right accuracy it does not provide a practical numerical scheme as the error analysis only holds for infinitely many refinement levels. In practice the refinement process is stopped at an arbitrary but finite number of refinements, but we cannot be sure, that the error produced by this alteration is of the required accuracy. To tackle this problem, we have two options:

1. Estimate the error induced by stopping the refinement process at a given level.
2. Provide a modification of the method which does not exhibit this problem.

As the first would strongly depend on the number of refinements and the grid size of the finest mesh, we decided to use the latter approach. The extension was published by the author in [8]. For our purpose we define $l_{\max}$ to be the index of the maximum coarsening level and we denote the discretized domain by $\Omega_{l_{\max}}$. At that level the Dirichlet boundary conditions of the original problem are imposed, i.e.

$$u(\mathbf{x}_\partial) = \frac{1}{4\pi} \int\limits_{\Omega} \frac{f(\mathbf{x})}{||\mathbf{y} - \mathbf{x}_\partial||_2} d\mathbf{y} \text{ for } \mathbf{x}_\partial \in \partial\Omega_{l_{\max}}. \tag{2.35}$$

So the boundary conditions of the Dirichlet problem that is solved numerically are set with the help of the fundamental solution. We immediately obtain the new problem to solve:

$$\begin{aligned} \Delta u(\mathbf{x}) &= f(\mathbf{x}),\ \mathbf{x} \in \Omega,\ \operatorname{supp}(f) \subset \Omega \subset \mathbb{R}^3, \\ u(\mathbf{x}_\partial) &= \frac{1}{4\pi} \int\limits_{\Omega} \frac{f(\mathbf{x})}{||\mathbf{y} - \mathbf{x}_\partial||_2} d\mathbf{y} \text{ for } \mathbf{x}_\partial \in \partial\Omega. \end{aligned} \tag{2.36}$$

The solution of this Dirichlet problem, which can be interpreted as a slice of the original problem with open boundary conditions, is the same as the solution of the original problem in that region, as stated by the following lemma:

Lemma 2.4 *Let $f \in C_0(\mathbb{R}^3) \cap L^2(\mathbb{R}^3)$ with $\operatorname{supp}(f) \subsetneq \mathbb{R}^3$ and let u be the solution of* (2.28) *with that right hand side f. Then u also is the unique solution of* (2.36) *in any bounded domain $\Omega \supset \operatorname{supp}(f)$.*

Proof. Let Ω be any domain that is a superset of $\operatorname{supp}(f)$. With Theorem 2.6 u fulfills (2.36) for all $\mathbf{x}_\partial \in \partial\Omega$. Uniqueness follows from Theorem 2.5. □

Imposing the boundary conditions with the help of the continuous problem does not yield the same solution as solving the discrete problem on the unrestricted domain. So as an extension to Theorem 2.9 we have to provide an error estimate for this step as well.

Theorem 2.10 *Assume that the discrete Green's function can be bounded by*

$$G_h(\mathbf{x}, \mathbf{p}) \leq \frac{1}{4\pi} \left[\frac{1}{||\mathbf{x} - \mathbf{p}||_2} + \frac{c_1}{||\mathbf{x} - \mathbf{p}||_2^3} \right].$$

Using the described grid coarsening strategy with a grid extension rate $\alpha \geq 2^{2/3}$ up to an arbitrary level $l_{max} \in \mathbb{N}$ and setting the boundary conditions at that level as in (2.35) *the error $e(\mathbf{x}, \mathbf{p})$, defined in* (2.33)*, is of order h^2 for all $\mathbf{x} \in \mathcal{G}_0$.*

Proof. For an arbitrary domain Ω_l the estimate of the volume integral in (2.34) holds as in the proof of Theorem 2.9. It remains to estimate the value of the surface integral. For that purpose let d be the minimum distance of a point of the original domain Ω_0 to the boundary of the domain discretized using the coarsest grid. As both, $\mathbf{x}$ and $\mathbf{p}$ are inside of the original domain, we can estimate the second integral:

$$\begin{aligned}
&\left| \oint_{\partial\Omega_{l_{\max}}} \left[G_h(\mathbf{x},\mathbf{s})\nabla_h(\tilde{G}(\mathbf{s},\mathbf{p}) - G_h(\mathbf{s},\mathbf{p})) - \nabla_h G_h(\mathbf{x},\mathbf{s})(\tilde{G}(\mathbf{s},\mathbf{p}) - G_h(\mathbf{s},\mathbf{p})) \right] \cdot \vec{\mathbf{n}} d\mathbf{s} \right| \\
&\qquad \leq \alpha^{3l_{\max}} \max_{\mathbf{s}\in\partial\Omega_{l_{\max}}} \left[\left| G_h(\mathbf{x},\mathbf{s})\nabla_h(\tilde{G}(\mathbf{s},\mathbf{p}) - G_h(\mathbf{s},\mathbf{p})) \cdot \vec{\mathbf{n}} \right| + \right. \\
&\qquad\qquad \left. \left| \nabla_h G_h(\mathbf{x},\mathbf{s}) \cdot \vec{\mathbf{n}} \, (\tilde{G}(\mathbf{s},\mathbf{p}) - G_h(\mathbf{s},\mathbf{p})) \right| \right] \\
&\qquad \leq \alpha^{3l_{\max}} \left[\left(\left| \frac{1}{4\pi}\frac{1}{d} \right| + \left| \frac{c_1 h_{l_{\max}}^2}{d^3} \right| \right) \left| \frac{3c_1 h_{l_{\max}}^2}{d^4} \right| + \right. \\
&\qquad\qquad \left. \left(\left| \frac{1}{4\pi}\frac{1}{d^2} \right| + \left| \frac{3c_1 h_{l_{\max}}^2}{d^4} \right| \right) \left| \frac{c_1 h_{l_{\max}}^2}{d^3} \right| \right] \\
&\qquad = \alpha^{3l_{\max}} \left[\left| \frac{1}{4\pi}\frac{3c_1 h_{l_{\max}}^2}{d^5} \right| + \left| \frac{3c_1^2 h_{l_{\max}}^4}{d^6} \right| + \left| \frac{1}{4\pi}\frac{3c_1 h_{l_{\max}}^2}{d^6} \right| + \left| \frac{3c_1^2 h_{l_{\max}}^4}{d^7} \right| \right].
\end{aligned}$$

Obviously, for $\alpha \geq 2^{2/3}$ we can estimate d as

$$d = \frac{\alpha^l - 1}{2} \geq \frac{\alpha^l}{4}$$

and for $h_{l_{\max}}$ we have

$$h_{l_{\max}} = 2^{(l-1)} h_1 = 2^{(l-1)} h.$$

So we get

$$\begin{aligned}
&\alpha^{3l_{\max}} \left[\left| \frac{1}{4\pi}\frac{3c_1 h_{l_{\max}}^2}{d^5} \right| + \left| \frac{3c_1^2 h_{l_{\max}}^4}{d^6} \right| + \left| \frac{1}{4\pi}\frac{3c_1 h_{l_{\max}}^2}{d^6} \right| + \left| \frac{3c_1^2 h_{l_{\max}}^4}{d^7} \right| \right] \\
&\qquad \leq \alpha^{3l_{\max}} \left[\left| \frac{1}{4\pi}\frac{3072 c_1 h^2}{\alpha^5} \left(\frac{2^2}{\alpha^5} \right)^{(l-1)} \right| + \left| \frac{12288 c_1^2 h^4}{\alpha^6} \left(\frac{2^4}{\alpha^6} \right)^{(l-1)} \right| + \right. \\
&\qquad\qquad \left. \left| \frac{1}{4\pi}\frac{12288 c_1 h^2}{\alpha^6} \left(\frac{2^2}{\alpha^6} \right)^{(l-1)} \right| + \left| \frac{49152 c_1^2 h^4}{\alpha^7} \left(\frac{2^4}{\alpha^7} \right)^{(l-1)} \right| \right].
\end{aligned}$$

This is order h^2 for $\alpha > 2^{2/3}$. □

Remark 2.1 *The assumption that the discrete Green's function G_h is bounded, i.e.*

$$G_h(\mathbf{x}, \mathbf{p}) \leq \frac{1}{4\pi} \left[\frac{1}{||\mathbf{x} - \mathbf{p}||_2} + \frac{c_1}{||\mathbf{x} - \mathbf{p}||_2^3} \right],$$

is justified in the delight of an asymptotic expansion of the five-point discretization of the Laplacian given by Burkhart in [16]. This expansion is missing terms of even powers, so our assumption is fulfilled.

Implementation of the grid coarsening

As noted in the definition of the different domains in (2.29) and (2.30) we introduced additional parameters β_l to simplify letting the different domains have common grid points. In the following we assume that the original domain $\Omega_0 = [-\frac{1}{2}, \frac{1}{2}]^3$ is discretized using grid spacing $h = 2^{-m}$, where $m > 2$. Therefore the domain of interest consists of 2^{3m} grid points. Furthermore we want to double the grid spacing h_l on each coarsening level, i.e.

$$h_l = 2^{-m+l-1}.$$

We define the domain on refinement level l as

$$\Omega_l := \left[-\frac{\beta_l}{2}, \frac{\beta_l}{2} \right]^3,$$

where β_l is the length of domain Ω_l. So for a conservative discretization of the flux as in (2.31) we need that the new domain has at least length

$$\beta_l \geq \beta_{l-1} + 4h_l. \tag{2.37}$$

For the error analysis to hold we need a grid extension rate of at least α^l, i.e. the length has to fulfill

$$\beta_l \geq \alpha^l \tag{2.38}$$

on each level l. To fit (2.37) and (2.38) and to simplify the implementation we choose $\beta_1 = 2$ and

$$\beta_l := \max \left(\beta_{l-1} + 4h_l, 2^{\lceil \log_2(\alpha^l) \rceil} \right) \text{ for } l > 1.$$

On level l we now have

$$n_l = \frac{2^{\lceil \log_2 \alpha^l \rceil)}}{2^{-k+l-1}}$$

grid points in each direction and the grid points on that level are given by

$$\mathbf{x}_{i,j,k}^l = h_l(i - \tfrac{n_l - 1}{2}, j - \tfrac{n_l - 1}{2}, k - \tfrac{n_l - 1}{2})^T,\ i, j, k = 0, \ldots, n_l - 1.$$

With that choice we can always reduce the problem to a coarse discretization with 9^3 unknowns in a finite number of coarsening steps, as stated by the following lemma.

Lemma 2.5 *Let $2^{2/3} \leq \alpha < 2$, $m \geq 3$ and let the hierarchical coarsening be defined by*

$$\begin{aligned} &\beta_0 = 2, \\ &\beta_l = \max\left(\beta_{l-1} + 4h_l, 2^{\lceil \log_2(\alpha^l) \rceil}\right) \text{ for } l > 1, \\ &\Omega_l = \left[-\frac{\beta_l}{2}, \frac{\beta_l}{2}\right]^3 \text{ and} \\ &h_l = 2^{-m+l-1}. \end{aligned} \tag{2.39}$$

Then we have that only 9 grid points are present in each direction on level

$$l_{max} := \left\lceil \frac{2-m}{\log_2 \alpha - 1} \right\rceil + 1 \tag{2.40}$$

and on all subsequent levels.

Proof. We start assuming

$$\beta_l = 2^{\lceil \log_2(\alpha^l) \rceil} \text{ for } l > 1,$$

neglecting the formation of the maximum in (2.39). Using that definition a level l at which only 9^3 or fewer grid points are left is reached when

$$\begin{aligned} & 2^{\lceil \log_2(\alpha^l) \rceil} \leq 8h_l \\ \Leftrightarrow \quad & \frac{2^{\lceil \log_2(\alpha^l) \rceil}}{2^{-m+l+1}} \leq 8 \\ \Leftrightarrow \quad & 2^{\lceil l \log_2(\alpha) - l + m + 1 \rceil} \leq 8 \\ \Leftrightarrow \quad & l(\log_2(\alpha) - 1) + m + 1 \leq 3 \\ \Leftrightarrow \quad & l \geq \frac{2-m}{\log_2(\alpha) - 1} \end{aligned}$$

If we show for levels below or equal to such an l that $\beta_{l-1} + 4h_l$ is not always larger than $2^{\lceil l \log_2(\alpha) \rceil}$ we have shown the first part of the proposition. For that purpose we note that

$$\begin{aligned} \beta_1 &= 2^{\lceil \log_2(\alpha) \rceil} = 2 \text{ and} \\ \beta_2 &= \beta_1 + 4h_2. \end{aligned}$$

Now for any level $l+1$ with

$$\begin{aligned} \beta_{l-1} &= 2^{\lceil (l-1) \log_2(\alpha) \rceil} \text{ and} \\ \beta_l &= \beta_{l-1} + 4h_l \end{aligned}$$

it holds true that

$$\beta_{l+1} = 2^{\lceil (l+1) \log_2(\alpha) \rceil} \Leftrightarrow \beta_l + 4h_{l+1} \leq 2^{\lceil (l+1) \log_2(\alpha) \rceil},$$

as

$$\begin{aligned}\beta_l + 4h_{l+1} &= \beta_{l-1} + 4h_l + 8h_l \\ &= 2^{\lceil (l-1)\log_2(\alpha)\rceil} + 12 \cdot 2^{-m+l-1}.\end{aligned}$$

With (2.40) for any level $l \leq l_{\text{max}}$ we have

$$l \leq \left\lceil \frac{2-m}{\log_2 \alpha - 1} \right\rceil + 1,$$

such that

$$12 \cdot 2^{-m+l-1} \leq 3 \cdot 2^{l-2+(l-1)(\log_2(\alpha)-1)+1} \leq 3 \cdot 2^{\lceil (l-1)\log_2(\alpha)\rceil}.$$

So we get

$$\begin{aligned}\beta_l + 4h_{l+1} &\leq 2^{\lceil (l-1)\log_2(\alpha)\rceil} + 3 \cdot 2^{\lceil (l-1)\log_2(\alpha)\rceil} \\ &= 2^{\lceil (l-1)\log_2(\alpha)\rceil + 2} \\ &\leq 2^{\lceil (l+1)\log_2(\alpha)\rceil}.\end{aligned}$$

To summarize: Up to level l_{max} each time β_l is equal to $\beta_{l-1} + 4h_l$, on the following refinement level we have $\beta_{l+1} = 2^{\lceil (l+1)\log_2(\alpha)\rceil}$. It remains to show that once refinement level l_{max} is reached, all subsequent levels possess 9^3 grid points as well. This is easy to see for any level l possessing 9^3 grid points, as after doubling the grid spacing 5 grid points are left in the domain Ω_l. Thus adding $4h_{l+1}$ in each direction doubles the length resulting in a domain eight times as big with 9 grid points in each direction. This length is the maximum, as $\alpha < 2$. So the next refinement level still possesses 9 grid points in each direction. □

The modified method not only has the same order of the discretization error than the original method with refinement up to infinitely many levels but also only a finite number of refinement steps depending linearly on the number of unknowns on the finest discretization level is necessary to reduce the problem to 9^3 grid points. As a consequence only $\mathcal{O}(N)$, where N is the total number of unknowns on the finest level, steps are required to impose the boundary conditions on the coarsest refinement level. Now it remains to show that the number of grid points grows linearly with the number of grid points of the innermost box.

Lemma 2.6 *Let α, m, β_l, Ω_l and h_l, $l = 1, \ldots, l_{max}$ be defined as in Lemma 2.5. Then the total number of grid points on all grids separately depends linearly on the number of grid points inside of the original domain Ω_0, namely $(2^m + 1)^3$.*

Proof. We show the assertion by induction over m. We set d to the maximum of the total number of grid points divided by 2^{3m} for $m = 3$ and $\frac{64}{7}c$, where c is the number of additional levels when we go from m to $m + 1$, i.e.

$$d := \max\left(\frac{1}{2^{3m}} \sum_{l=1}^{l_{\max(m)}} \#\mathcal{G}_l \Big|_{m=3}, \frac{64}{7}c \right),$$

with

$$l_{\max(m)} := \left\lceil \frac{2-m}{\log_2 \alpha - 1} \right\rceil + 1.$$

We like to note that c is a constant, as $l_{\max(m)}$ is bounded by a linear function in m. Now we show that

$$\sum_{l=1}^{l_{\max(m)}} \#\mathcal{G}_l \leq d2^{3m}, \tag{2.41}$$

which is obviously true for $m = 3$. Assume (2.41) holds for some $m \in \mathbb{N}$, then it holds for $m+1$ as well, since:

$$\begin{aligned}\sum_{l=1}^{l_{\max(m+1)}} \#\mathcal{G}_l &= \sum_{l=1}^{l_{\max(m)}} \#\mathcal{G}_l + \sum_{l=\max(m)+1}^{l_{\max(m+1)}} \#\mathcal{G}_l \\ &\leq d2^{3m} + \sum_{l=\max(m)+1}^{l_{\max(m+1)}} \#\mathcal{G}_l\end{aligned}$$

Obviously, the number of grid points on each of the c additional levels going from m to $m+1$ is bounded by 2^{3m+6}, yielding

$$\begin{aligned}\sum_{l=1}^{l_{\max(m+1)}} \#\mathcal{G}_l &\leq d2^{3m} + c2^{3m+6} \\ &= (\frac{d}{8} + 8c)2^{3(m+1)} \\ &\leq d\,2^{3(m+1)}.\end{aligned}$$

□

Combining the results we have shown that we have constructed an optimal method of the desired accuracy in the following sense: The number of arithmetical operations per unknown on the finest level is bounded from above by a constant and the number of coarsening steps is predetermined by the size of the finest grid. At the same time the order of the reached accuracy is not influenced by the number of coarsening steps, but depends on the grid spacing on the finest level, only.

Comparison of the unmodified method and our modification

We like to conclude this chapter with a numerical comparison of the original method by Washio and Oosterlee and our method. For that purpose we implemented the method in C,

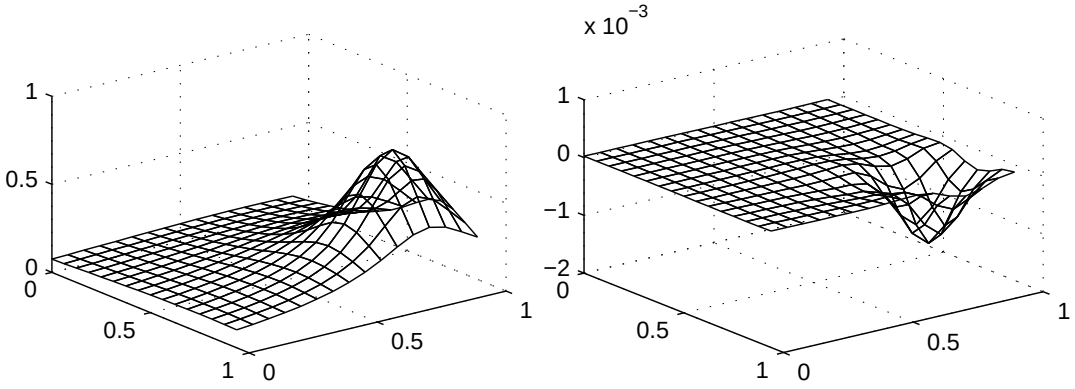

Figure 2.3: A cut through the computed solution of the test case and its analytic point-wise error on a 64^3 grid. Every fourth grid point is plotted.

$\#\mathcal{G}_0$	h	#refinements	$\|\|\mathbf{u}-\mathbf{u}^*\|\|_\infty$	$\|\|\mathbf{u}-\mathbf{u}^*\|\|_2/\#\mathcal{G}_0$	time
17^3	$1/16$	8	$2.110010 \cdot 10^{-2}$	$2.162535 \cdot 10^{-5}$	1.66 s
33^3	$1/32$	11	$5.078421 \cdot 10^{-3}$	$1.810825 \cdot 10^{-6}$	12.84 s
65^3	$1/64$	14	$1.251313 \cdot 10^{-3}$	$1.580911 \cdot 10^{-7}$	104.61 s
129^3	$1/128$	17	$3.112553 \cdot 10^{-4}$	$1.392736 \cdot 10^{-8}$	909.64 s

Table 2.1: Error and timings for different various sizes. The ∞-norm of the error decreases as predicted and the method scales linearly with the number of grid points.

using the FAC method introduced later in Chapter 3.2.4 as a solver for the resulting linear system. The performance was measured on a machine with an 1.7 GHz Power4+ CPU. The grid extension rate α was set to $1.6 > 2^{2/3}$ and for practical reasons β has been chosen as

$$\beta := \lceil 2^{\lceil \log_2(\alpha^l) \rceil} \rceil.$$

We used a point symmetric density described by a translated cubic B-Spline as defined later in Chapter 4 as right hand side f. So the exact solution u^* to the problem is known analytically. The computed solution on a 64^3 grid and the error of this test case can be found in Figure 2.3. Timings and error norms for various grid sizes are shown in Table 2.1. Obviously, the method scales linearly and the ∞-norm of the error decreases as expected.

As it can be seen in Table 2.2 the number of refinement steps does not influence the method's accuracy, although the timings vary a lot. This is a consequence of the reduction of the number of boundary points, when the number of refinement steps is increased. We ran the same test using the original method presented in [87], thus not setting the boundary values to the values of the continuous problem. The results in Table 2.3 and Fig. 2.4 show that this method behaves as expected: Increasing the number of grid refinements increases the accuracy of the method up to the same level than our modification.

The presented method is a useful extension of Washio's and Oosterlee's method. The number of coarsening steps is known a priori and using a multigrid solver for the solution

#refinements	$\#\mathcal{G}_{l_{\max}}$	$\|\|\mathbf{u}-\mathbf{u}^*\|\|_\infty$	$\|\|\mathbf{u}-\mathbf{u}^*\|\|_2/\#\mathcal{G}_0$
2	65^3	$5.089194 \cdot 10^{-3}$	$2.023222 \cdot 10^{-6}$
3	65^3	$5.085428 \cdot 10^{-3}$	$1.857736 \cdot 10^{-6}$
4	37^3	$5.066483 \cdot 10^{-3}$	$1.927579 \cdot 10^{-6}$
5	33^3	$5.063288 \cdot 10^{-3}$	$1.840964 \cdot 10^{-6}$
6	33^3	$5.079554 \cdot 10^{-3}$	$1.815541 \cdot 10^{-6}$
7	21^3	$5.067220 \cdot 10^{-3}$	$1.815151 \cdot 10^{-6}$
8	17^3	$5.070326 \cdot 10^{-3}$	$1.811852 \cdot 10^{-6}$
9	17^3	$5.084148 \cdot 10^{-3}$	$1.812722 \cdot 10^{-6}$
10	13^3	$5.084021 \cdot 10^{-3}$	$1.812488 \cdot 10^{-6}$
11	9^3	$5.078421 \cdot 10^{-3}$	$1.810825 \cdot 10^{-6}$
12	9^3	$5.084541 \cdot 10^{-3}$	$1.812455 \cdot 10^{-6}$
13	9^3	$5.088087 \cdot 10^{-3}$	$1.813763 \cdot 10^{-6}$
14	9^3	$5.089895 \cdot 10^{-3}$	$1.814523 \cdot 10^{-6}$
15	9^3	$5.090805 \cdot 10^{-3}$	$1.814928 \cdot 10^{-6}$
16	9^3	$5.091260 \cdot 10^{-3}$	$1.815137 \cdot 10^{-6}$
17	9^3	$5.091489 \cdot 10^{-3}$	$1.815244 \cdot 10^{-6}$
18	9^3	$5.091603 \cdot 10^{-3}$	$1.815297 \cdot 10^{-6}$
19	9^3	$5.091660 \cdot 10^{-3}$	$1.815324 \cdot 10^{-6}$
20	9^3	$5.091688 \cdot 10^{-3}$	$1.815337 \cdot 10^{-6}$

Table 2.2: Error norms for a 33^3-problem with $h = 1/32$ and various refinements. The error of the method is only marginally affected by the number of refinement steps.

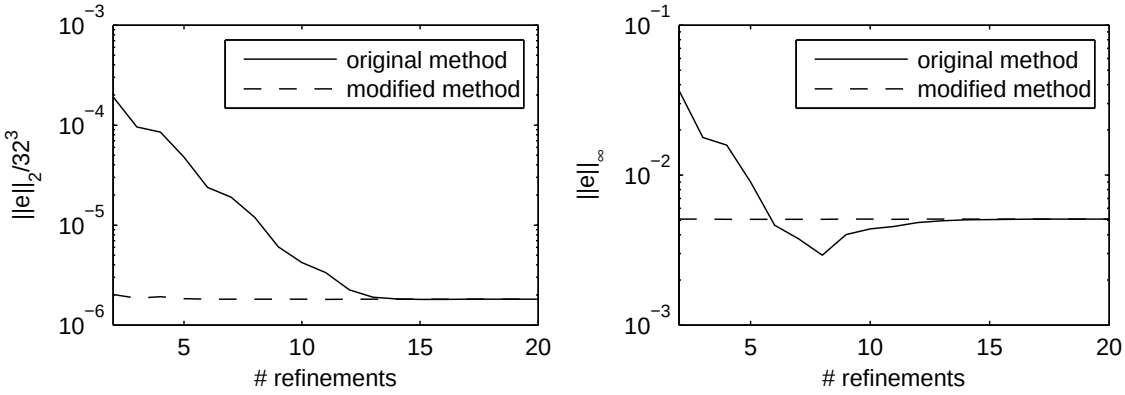

Figure 2.4: Behavior of the error of the original method and of the modification. Using the original method both, the error in the 2-norm and in the ∞-norm, depend heavily on the number of grid refinements. The accuracy converges to the accuracy of our modification, that is almost independent of the number of refinements.

#refinements	$\#\mathcal{G}_{l_{\max}}$	$\|\|\mathbf{u}-\mathbf{u}^*\|\|_\infty$	$\|\|\mathbf{u}-\mathbf{u}^*\|\|_2/\#\mathcal{G}_0$
2	65^3	$3.653597 \cdot 10^{-2}$	$1.915066 \cdot 10^{-4}$
3	65^3	$1.783026 \cdot 10^{-2}$	$9.573523 \cdot 10^{-5}$
4	37^3	$1.584892 \cdot 10^{-2}$	$8.508257 \cdot 10^{-5}$
5	33^3	$8.995442 \cdot 10^{-3}$	$4.779984 \cdot 10^{-5}$
6	33^3	$4.631318 \cdot 10^{-3}$	$2.383245 \cdot 10^{-5}$
7	21^3	$3.762046 \cdot 10^{-3}$	$1.905511 \cdot 10^{-5}$
8	17^3	$2.929352 \cdot 10^{-3}$	$1.192743 \cdot 10^{-5}$
9	17^3	$4.014153 \cdot 10^{-3}$	$6.073405 \cdot 10^{-6}$
10	13^3	$4.375166 \cdot 10^{-3}$	$4.211756 \cdot 10^{-6}$
11	9^3	$4.554064 \cdot 10^{-3}$	$3.346295 \cdot 10^{-6}$
12	9^3	$4.821822 \cdot 10^{-3}$	$2.248768 \cdot 10^{-6}$
13	9^3	$4.956727 \cdot 10^{-3}$	$1.902828 \cdot 10^{-6}$
14	9^3	$5.024221 \cdot 10^{-3}$	$1.821841 \cdot 10^{-6}$
15	9^3	$5.057969 \cdot 10^{-3}$	$1.809017 \cdot 10^{-6}$
16	9^3	$5.074843 \cdot 10^{-3}$	$1.809788 \cdot 10^{-6}$
17	9^3	$5.083280 \cdot 10^{-3}$	$1.811972 \cdot 10^{-6}$
18	9^3	$5.087498 \cdot 10^{-3}$	$1.813512 \cdot 10^{-6}$
19	9^3	$5.089608 \cdot 10^{-3}$	$1.814394 \cdot 10^{-6}$
20	9^3	$5.090662 \cdot 10^{-3}$	$1.814863 \cdot 10^{-6}$

Table 2.3: Error norms for a 33^3-problem with $h = 1/32$ and various refinements using the method of Washio and Oosterlee. The error of the method heavily depends on number of refinement steps, reaching the same accuracy as the modified method.

of the linear system the computational cost grows linearly with the number of unknowns as intended by Washio and Oosterlee. The error analysis presented shows that independent of the number of refinement steps the method is of the desired order of accuracy. In contrast to that the original method lacks this independence, as the error analysis is based on the assumption that infinitely many coarsening steps are carried out. In practice this number is an additional parameter that has to be provided by the user. As clearly seen in the numerical examples, the accuracy of the original method depends on the number of coarsening steps.

Chapter 3

Multigrid Methods

3.1 Iterative methods

In the following we are interested in the solution of linear systems using iterative methods. For that purpose let $A \in \mathbb{R}^{n\times n}, n \in \mathbb{N}$, regular and let $\mathbf{b} \in \mathbb{R}^n$. Later on, we will use the field $\mathbb{C}$ instead of $\mathbb{R}$, as it simplifies representation. We are interested in the solution $\mathbf{x} \in \mathbb{R}^n$ of linear systems of the form

$$A\mathbf{x} = \mathbf{b}. \tag{3.1}$$

A lot of different methods exist to solve this system directly or iteratively. Examples for direct solution methods are Gaussian elimination or the Cholesky decomposition. Besides roundoff errors and memory requirements the main drawback of direct solvers is their high arithmetical complexity, e.g. the Gaussian elimination is of order $\mathcal{O}(n^3)$ if one cannot exploit the sparsity of A. In this work we are interested in iterative methods, the arithmetic complexity of which should be significantly smaller. This short introduction to iterative methods is based on the books by Meister [67] and Hackbusch [54], for further details we refer their works. In our case (3.1) is solved using an iterative method ϕ.

Definition 3.1 *An* iterative method *is a mapping*

$$\phi : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n.$$

In the following we denote by $\mathbf{x}^{(0)} \in \mathbb{R}^n$ the initial approximation. The new iterate $\mathbf{x}^{(k+1)}$ is computed with the help of $\mathbf{x}^{(k)}$ and $\mathbf{b}$ as

$$\mathbf{x}^{(k+1)} = \phi(\mathbf{x}^{(k)}, \mathbf{b}).$$

We demand from a numerical method that it converges against the solution of the system and that the solution of the system is a fixed point of the method.

Definition 3.2 *An iterative method ϕ is called* consistent with A *iff for all* $\mathbf{b} \in \mathbb{R}^n$ $A^{-1}\mathbf{b}$ *is a fixed point of* $\phi(\cdot, \mathbf{b})$. *It is called* convergent *iff for all* $\mathbf{b} \in \mathbb{R}^n$ *and for all initial approximations* $\mathbf{x}^{(0)} \in \mathbb{R}^n$ *the sequence* $\{\mathbf{x}^{(k)}\}_{k=0}^{\infty}$ *has the limit* $A^{-1}\mathbf{b}$.

Both consistency and convergence are necessary conditions for an iterative method to be a meaningful method.

3.1.1 Linear iterative methods

Definition 3.3 *An iterative method ϕ is called a* linear iterative method *iff there exist matrices* $M, N \in \mathbb{R}^{n\times n}$ *such that*

$$\phi(\mathbf{x}, \mathbf{b}) = M\mathbf{x} + N\mathbf{b}.$$

The matrix M is called iteration matrix.

For a linear iterative method necessary and sufficient conditions for consistency and convergence can be given as stated by the following two theorems.

Theorem 3.1 *A linear iterative method ϕ is consistent iff we can write*

$$M = I - NA.$$

Proof. Let $\mathbf{x}^* = A^{-1}\mathbf{b}$. Assume that $\mathbf{x}^*$ is a fixed point of $\phi(\cdot, \mathbf{b})$, so we have

$$\mathbf{x}^* = \phi(\mathbf{x}^*, \mathbf{b}) = M\mathbf{x}^* + N\mathbf{b} = (M + NA)\mathbf{x}^*.$$

This is the case for all $\mathbf{b} \in \mathbb{R}^n$, i.e. for all $\mathbf{x}^* \in \mathbb{R}^n$, iff $I = M + NA$. □

Theorem 3.2 *A linear iterative method ϕ is convergent iff the spectral radius of the iteration matrix is bounded from above by* 1, *i.e.*

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

Proof. See e.g. the proof of Theorem 3.2.7 in [54]. □

For the analysis of multigrid methods which use linear iterative methods as smoothers the following lemma is helpful.

Lemma 3.1 *The k-th iterate of the linear iterative method ϕ can be written as*

$$\mathbf{x}^k = M^k\mathbf{x}^{(0)} + \sum_{l=0}^{k-1} M^l N\mathbf{b}. \tag{3.2}$$

Proof. We prove the statement by induction. For $k = 1$ equation (3.2) holds. Assume that (3.2) holds for $k - 1$. Inserting the definitions yields

$$\mathbf{x}^{(k)} = M\left(M^{k-1}\mathbf{x}^{(0)} + \sum_{l=0}^{k-2} M^l N\mathbf{b}\right) + N\mathbf{b} = M^k\mathbf{x}^{(0)} + \sum_{l=0}^{k-1} M^l N\mathbf{b}$$

□

So applying k iterations of a linear iterative method results in multiplying the current approximation by the k-th power of the method's iteration matrix and adding a modification of the right hand side. Starting with a zero approximation we can give an explicit formula for $\mathbf{x}^{(k)}$.

Lemma 3.2 *Let $\phi : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ be a consistent linear iteration method with iteration matrix $M = I - NA$. If we start with a zero approximation for the solution of $A\mathbf{x} = \mathbf{b}$, we can write the k-th iterate as*

$$\mathbf{x}^{(k)} = (I - M^k)A^{-1}\mathbf{b}.$$

Proof. For A we can write $A = N^{-1}(I - M) \Leftrightarrow A^{-1} = (I - M)^{-1}N$. As $\mathbf{x}^{(0)} = \mathbf{0}$, we have

$$\begin{aligned}\mathbf{x}^{(k)} &= (I + M + M^2 + \cdots + M^{k-1})N\mathbf{b} \\ &= (I - M^k)(I - M)^{-1}N\mathbf{b} \\ &= (I - M^k)A^{-1}\mathbf{b}.\end{aligned}$$

□

3.1.2 Splitting methods

From Definition 3.3 it is not obvious how to choose either M or N. One way to construct a linear iterative method is the splitting of the matrix A, i.e. with regular $B \in \mathbb{R}^{n\times n}$ we write

$$A = B + (A - B).$$

Now we can define the iterative method ϕ by

$$\phi(\mathbf{x}, \mathbf{b}) = B^{-1}(B - A)\mathbf{x} + B^{-1}\mathbf{b} = (I - B^{-1}A)\mathbf{x} + B^{-1}\mathbf{b},$$

thus we set $M = (I - B^{-1}A)$ and $N = B^{-1}$. Obviously the defined iterative method is consistent, as B is regular. The key idea is to define B to be similar to A and easy to invert. One of the first ideas is to set B to the product of the identity and an arbitrary value, resulting in the Richardson method.

Definition 3.4 *Let $\theta > 0$. Then the* Richardson method *is defined as the linear iterative method*

$$\phi_{Richardson,\theta}(\mathbf{x}, \mathbf{b}) = (I - \theta A)\mathbf{x} + \theta\mathbf{b}.$$

Theorem 3.3 *Let A be symmetric and positive definite, let λ_{min} be the smallest and let λ_{max} be the largest eigenvalue of A. Then the Richardson method converges iff $\theta \in (0, 2/\lambda_{max})$ and the convergence rate is*

$$\rho(M_{Richardson,\theta}) = \max\{|1 - \theta\lambda_{min}|, |1 - \theta\lambda_{max}|\}. \tag{3.3}$$

Proof. Let λ_A be an eigenvalue of A, then obviously $1-\theta\lambda$ is an eigenvalue of $M_{\text{Richardson},\theta}$. As the function $1 - \theta\lambda$ has no local maxima, we immediately obtain (3.3). Now assume that $\theta \in (0, 2/\lambda_{\text{max}})$, so we have

$$-1 < 1 - \theta\lambda_{\text{max}} \leq 1 - \theta\lambda_{\text{min}} < 1,$$

so $\rho(M_{\text{Richardson},\theta}) < 1$. This shows sufficiency. To show necessity we assume $\rho(M_{\text{Richardson},\theta}) < 1$. With

$$1 > \rho(M_{\text{Richardson},\theta}) \geq |1 - \theta\lambda_{\text{max}}| \geq 1 - \theta\lambda_{\text{max}}$$

we have $\theta > 0$, from

$$-1 < \rho(M_{\text{Richardson},\theta}) \leq -|1 - \theta\lambda_{\text{max}}| \leq 1 - \theta\lambda_{\text{max}}$$

we obtain $\theta < 2/\lambda_{\text{max}}$. □

Another well-known splitting method is the Jacobi method, where B is chosen to contain only the diagonal of A.

Definition 3.5 *Let $A = D + L + U$, where D is the matrix containing only the diagonal of A, L contains only the lower triangular part and U only the upper triangular part. The* Jacobi method *is the linear iterative method given by*

$$\phi_{Jacobi}(\mathbf{x}, \mathbf{b}) = -D^{-1}(L + U)\mathbf{x} + D^{-1}\mathbf{b}.$$

Its iteration matrix is denoted by $M_{Jacobi} = -D^{-1}(L + U)$.

A number of convergence criterions exists. We just would like to mention the criterion for positive definite matrices. Here and in the following, for A and B being two symmetric matrices the expression "$A > B$" denotes that $A - B$ is symmetric and positive definite, "$A \geq B$" denotes that $A - B$ is symmetric and positive semi-definite. For some symmetric and positive definite matrix C and matrices D and E that are such that CD and CE are positive definite, by "$D >_C E$" and "$D \geq_C E$" we denote that $CD - CE$ is symmetric and positive definite and symmetric and positive semi-definite, respectively.

Theorem 3.4 *Let both A be symmetric positive definite and let the relation*

$$2D > A > 0$$

hold. Then the Jacobi method converges and its convergence rate is given by

$$\rho(M_{Jacobi}) = \|M_{Jacobi}\|_A = \|M_{Jacobi}\|_D < 1.$$

Proof. Obviously we have

$$2D > A > 0 \Leftrightarrow 2I > \underbrace{D^{-\frac{1}{2}} A D^{-\frac{1}{2}}}_{=:A'} > 0.$$

So $\sigma(A') \subset (0, 2)$. Now the matrix

$$M' := I - A' = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}} = D^{\frac{1}{2}} M_{\text{Jacobi}} D^{-\frac{1}{2}}$$

is similar to M_{Jacobi}, so

$$\sigma(M_{\text{Jacobi}}) = \sigma(M') \subset (-1, 1).$$

Additionally

$$\rho(M_{\text{Jacobi}}) = \rho(M') = \|M'\|_2 = \|D^{-\frac{1}{2}} M' D^{\frac{1}{2}}\|_D = \|M_{\text{Jacobi}}\|_D.$$

Using the similar symmetric matrix $A^{\frac{1}{2}} M_{\text{Jacobi}} A^{-\frac{1}{2}}$, we obtain

$$\rho(M_{\text{Jacobi}}) = \rho(A^{\frac{1}{2}} M_{\text{Jacobi}} A^{-\frac{1}{2}}) = \|A^{\frac{1}{2}} M_{\text{Jacobi}} A^{-\frac{1}{2}}\|_2 = \|M_{\text{Jacobi}}\|_A.$$

□

Remark 3.1 *Writing the Jacobi method component-wise yields*

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{\substack{j=1 \\ j \neq i}}^{n} a_{ij} x_j^{(k)} \right).$$

Using not only components of the old iterate $\mathbf{x}^{(k)}$ but the available components of the new iterate $\mathbf{x}^{(k+1)}$ results in

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} \right),$$

which is the component-wise version of the Gauss-Seidel method. In matrix form it reads

$$\phi_{GS}(\mathbf{x}, \mathbf{b}) = -(D + L)^{-1} U \mathbf{x} + (D + L)^{-1} \mathbf{b}.$$

3.1.3 Relaxation methods

The new iterate of a linear iteration method can be written in terms of the residual vector $\mathbf{r} := \mathbf{b} - A\mathbf{x}$ as

$$\mathbf{x}^{(k+1)} = (I - NA)\mathbf{x}^{(k)} + N\mathbf{b} = \mathbf{x}^{(k)} + N(\mathbf{b} - A\mathbf{x}^{(k)}) = \mathbf{x}^{(k)} + N\mathbf{r}^{(k)}.$$

By weighting the correction we get

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \omega N\mathbf{r}^{(k)},$$

resulting in a new linear iterative method. The additional parameter ω allows us to optimize the spectral radius of the original method's iteration matrix. By introducing the parameter to the Jacobi method we get the JOR method.

Definition 3.6 *Let A, D, L and U be as in Definition 3.5. The* Jacobi overrelaxation method *or for short* JOR *is the linear iterative method given by*

$$\phi_{JOR,\omega}(\mathbf{x}, \mathbf{b}) = \mathbf{x} - \omega D^{-1}(A\mathbf{x} + \mathbf{b}).$$

For the JOR method we can formulate the following convergence criterion.

Theorem 3.5 *Let A be symmetric and and positive definite and let ω fulfill*

$$0 < \omega < 2/\rho(D^{-1}A). \tag{3.4}$$

Then the JOR method converges, and its convergence rate is given by

$$\rho(M_{JOR,\omega}) = \|M_{JOR,\omega}\|_A = \|M_{JOR,\omega}\|_D < 1.$$

Proof. We have $(D^{-1}A)^{-1} \geq 1/\rho(D^{-1}A)I$. Thus, with condition (3.4) we have

$$0 < \omega I < 2/\rho(D^{-1}A)I \leq 2(D^{-1}A)^{-1} = 2A^{-1}D.$$

This in turn implies

$$0 < \omega A < 2D.$$

The rest of the proof proceeds like the proof of Theorem 3.4, which states the convergence criterion for the Jacobi method. □

Remark 3.2 *The Gauss-Seidel method mentioned in Remark 3.1 can be extended by a relaxation parameter in a similar way as the Jacobi method, the main difference being the component-wise introduction of ω. The resulting method is the well-known SOR method.*

Linear iterative methods like the Richardson method or the Jacobi and JOR method, respectively, are easy to analyze and to implement. The convergence rate directly depends on the eigenvalues of the original system. For example the eigenvalues of the iteration matrix of the Richardson method are given as

$$\lambda_{M_{\text{Richardson},\theta}} = 1 - \theta\lambda_A,$$

where λ_A is an eigenvalue of A. So for an ill-conditioned system with an eigenvalue close to zero the convergence rate of the Richardson iteration will be smaller than one, but very close to it. As a consequence it will be unsatisfactory. Other methods like Jacobi and Gauss-Seidel and their relaxed variants behave in the same way. This is in contrast to Multigrid methods which do not share this downside for certain classes of matrices.

3.2 Geometric Multigrid

Multigrid methods are optimal, i.e. $\mathcal{O}(n)$, methods for the solution of certain linear systems arising from the discretization of elliptic PDEs. Additionally, they are efficient, i.e. the constant factor that is multiplied with the leading n-term is small. Multigrid is a universal principle that can be applied to a wide range of elliptic problems, e.g. problems with non-constant coefficients, different discretizations, etc. and to non-elliptic problems as well. The origins of multigrid go back to the workings of Fedorenko [30, 31], who analyzed the convergence of a multigrid method solving a discretized elliptic PDE of second order with Dirichlet boundary conditions. Further on Bakhvalov [6] is to be named, who mentioned the use of nested iterations in order to improve the initial approximation. Brandt used the ideas contained in these papers in his work on adaptive rediscretization and showed their practical efficiency [11]. Later, he published a very detailed work on multigrid methods [12]. Simultaneous to these developments, Hackbusch worked on multigrid methods for the solution of elliptic PDEs as well [48, 49, 50, 51], putting particular emphasize on mathematical rigor.

We stick to the standard model problem and definitions as most introductory multigrid books that are much more detailed, see e.g. [15, 84].

3.2.1 Motivation

As aforementioned, iterative methods like the Richardson method or the Jacobi method converge very slowly for ill-conditioned systems. We want to analyze this effect a little more in detail. Although this observation can be made for a large class of problems, we restrict ourselves to the Poisson equation (2.5) with Dirichlet boundary conditions (2.2a) on the unit square, where u vanishes on the boundary, i.e.

$$-\Delta u(\mathbf{x}) = f(\mathbf{x}) \text{ for } \mathbf{x} \in [0,1]^2$$

and

$$u(\mathbf{x}) = 0 \text{ for } x_1 = 0 \vee x_1 = 1 \vee x_2 = 0 \vee x_2 = 1.$$

Now we discretize the domain using $n+1$ points in each direction, where $n = 2^k$. Using the 5-point formula (2.15) for the discretization of the Laplacian we get

$$\frac{1}{h_k^2}(4u_{i,j} - u_{i-1,j} - u_{i,j-1} - u_{i+1,j} - u_{i,j+1}) = f_{i,j}$$

with $h_k = 2^{-k}$ for $i, j = 1, \ldots, n_k$ and for $i = 0 \vee i = n \vee j = 0 \vee j = n$ we have

$$(u_k)_{i,j} = 0.$$

This results in the linear system

$$L_k \mathbf{u}_k = \mathbf{f}_k, \tag{3.5}$$

where $L_k \in \mathbb{R}^{\mathbf{n}_k \times \mathbf{n}_k}, \mathbf{n}_k = (2^k - 1)^2$ and $\mathbf{u}_k, \mathbf{f}_k \in \mathbb{R}^{\mathbf{n}_k}$. To determine the convergence factor of the Richardson method and the Jacobi method, we need the eigenvalues of L_k. One easily verifies that the vectors $\varphi_{l,m}^{(k)}$ with the components

$$(\varphi_{l,m}^{(k)})_{i,j} = \sin(l\pi ih)\sin(m\pi jh), \text{ for } i, j, l, m = 1, \ldots, n_k \tag{3.6}$$

are the eigenvectors of L_k. The associated eigenvalues are

$$\lambda_{l,m}^{(k)} = 4 - 2\cos(l\pi h) - 2\cos(m\pi h), \text{ for } l, m = 1, \ldots, n_k.$$

So the smallest eigenvalue of L_k is

$$\lambda_{\min}^{(k)} = 4(1 - \cos(\pi h_k)).$$

By Theorem 3.3 and Theorem 3.5 we easily find that the convergence rate for the Richardson method is

$$\rho(M_{\text{Richardson},\theta}) = 1 - \theta(1 - \cos(\pi h_k)),$$

and for the Jacobi method we get

$$\rho(M_{\text{Jacobi}}) = \cos(\pi h_k).$$

Therefore both methods converge slowly for large k, which is not surprising, as the system is asymptotically ill-conditioned.

As the entries on the main diagonal of the coefficient matrix are constant, for this problem the Richardson method is equivalent to the damped Jacobi method. In the following we will cover the Jacobi method in larger detail. Looking more closely at the convergence rate of the Jacobi method for the different eigenvectors, we find that it depends on the associated eigenvalue. If we represent the error

$$\mathbf{e}_k = \mathbf{u}_k^* - \mathbf{u}_k,$$

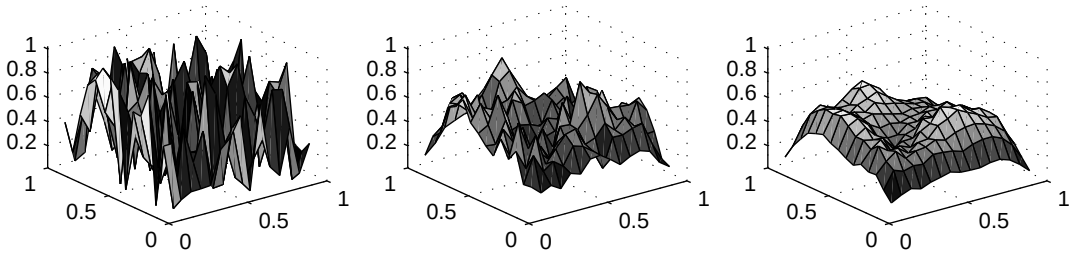

Figure 3.1: Error of an arbitrarily chosen initial approximation and right hand side of the Laplacian discretized on the unit square using 15^2 grid points before and after application of one and three iterations of a damped Jacobi method with $\omega = 4/5$.

where $\mathbf{u}_k^*$ is the exact solution in terms of the eigenvectors, we can immediately determine, which parts of the error are reduced efficiently and which are not. We find out that the part belonging to the eigenvalue $\lambda_{l,m}^{(k)}$ is damped by a factor of $|\frac{1}{2}(\cos(l\pi h_k)\cos(m\pi h_k))|$. So the parts belonging to eigenvalues with indices l and m somewhere in the middle between 1 and n_k are damped efficiently, while parts belonging to eigenvalues with extreme indices are hardly damped at all. Now we analyze, which parts of the error are damped by the JOR method. We obtain that the part belonging to the eigenvalue with index l, m is damped by a factor of

$$\left|1 - \frac{\omega}{2}(2 - \cos(l\pi h_k) - \cos(m\pi h_k))\right|.$$

So for an $\omega < 1$ we can achieve that parts of the error belonging to eigenvalues with large indices l and m are damped efficiently by a factor of at least $|1 - 2\omega|$. The parts of the error belonging to eigenvalues with small indices are still damped very inefficiently, as they are at least asymptotically not damped at all. Now, we observe that the eigenvectors (3.6) belonging to eigenvalues with high coefficients l and m are geometrically highly oscillatory. This means that high frequency parts are damped very efficiently by the Jacobi method, while low frequencies are damped much slower. The error is becoming smooth after only a few iterations of the Jacobi method. This is the fundamental observation that lead to the development of multigrid methods. This behavior can be easily verified by plotting the error before and after applying a few iterations of the Jacobi method, c.f. Figure 3.1.

Another fundamental observation that has to be made in order to construct a twogrid method is that a smooth error is well-represented on a coarser grid. That means a smaller number of grid points is sufficient. Given a current approximation $\mathbf{u}_k$ to the solution of (3.5), we can compute the residual $\mathbf{r}_k$ as

$$\mathbf{r}_k = \mathbf{f}_k - L_k\mathbf{u}_k.$$

The actual iterate can then be updated by adding the approximate solution $\mathbf{e}_k$ of the defect equation

$$L_k\mathbf{e}_k = \mathbf{r}_k. \tag{3.7}$$

This approximate solution can be obtained from the coarse grid, as it is well represented on that level.

On this coarser grid the low frequency components of the finer grid can be differentiated into low and high frequency components, again. The Jacobi method still has the smoothing property on this level, resulting in a very efficient damping of the high frequency parts of the error, which have been low frequency parts on the fine grid. As a consequence, a recursive application of the twogrid idea is possible, leading to a multigrid method.

Now, we will continue to formally define the twogrid and multigrid methods.

3.2.2 Twogrid methods

Twogrid methods consist of three main ingredients: the smoother, the restriction and prolongation operators, and the coarse grid correction operator.

Smoothers

Essentially, all iterative methods that smooth the error in a geometrical sense, i.e. damp the high frequency components efficiently and independently of h, are possible smoothers for a twogrid method. The most common smoothers are the damped JOR method and the SOR method, as defined in Theorem 3.6 and Remark 3.2. We will now give formal definitions of high and low frequencies and of the smoothing factor of the JOR method for (3.5).

Definition 3.7 *Let L_k be defined as in* (3.5)*. An eigenvector $\varphi_{l,m}^{(k)}$ given in* (3.6) *is called*

$$\begin{aligned} &\textit{low frequency,} && \textit{if } \max(l,m) < (n_k+1)/2, \\ &\textit{high frequency,} && \textit{if } (n_k+1)/2 \leq \max(l,m). \end{aligned}$$

Definition 3.8 *Let L_k be defined as in* (3.5) *and let*

$$\chi_{l,m}^{(k)}(\omega) := 1 - \frac{\omega}{2}(2 - \cos(l\pi h) - \cos(m\pi h))$$

be the factor by which the eigenvector $\varphi_{l,m}^{(k)}$ is damped by the JOR method. Then the smoothing factor $\mu_k(\omega)$ *of the JOR method is defined as*

$$\mu_k(\omega) := \max\{|\chi_{l,m}^{(k)}(\omega)| : (n_k+1)/2 \leq \max(l,m) \leq n_k\},$$

i.e. the worst factor by which a high frequency is damped. Further we define its suprenum over k as

$$\mu(\omega) = \sup_{k\in\mathbb{N}} \mu_k(\omega).$$

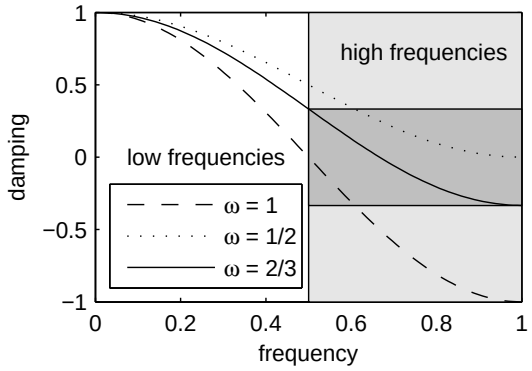

Figure 3.2: Damping factors $\chi_{l,m}$ for $h \to 0$ of the JOR method for the 1D analogon to our model problem for different relaxation parameters ω. The choice $\omega = 2/3$ is optimal and all high frequency components are damped by a factor of at least $1/3$.

Thus the relaxation parameter is optimal if we choose ω as the minimizer of $\mu(\omega)$. In our case $\omega = 4/5$ is optimal. For the 1D analogon of our problem the choice $\omega = 2/3$ is optimal, as depicted in Figure 3.2.

Remark 3.3 *The eigenvectors of the iteration matrix of the Gauss-Seidel and SOR methods are not the same as the eigenvectors of L_k. So the analysis of these smoothers is more involved, requiring other tools as presented here, e.g. the local Fourier analysis (LFA). For details see [84].*

In the following we do not restrict ourselves to the JOR method as a smoother, but we just assume that some appropriate smoothing method S was chosen. S is a linear iterative method, although other methods have been used as smoothers in multigrid methods. To simplify the representation we define $(\phi_{\mathrm{S}}^{(k)})^{\nu}$ to represent ν iterations of the smoothing method on the grid with grid spacing h_k. This is possible, as due to Lemma 3.1 $\nu \geq 1$ iterations of one linear iterative method define another linear iterative method.

Restriction and prolongation operators

So far, we have not mentioned how to transfer the residual from the fine grid to the coarse grid and the result of the solution of the defect equation (3.7) on the coarse grid back to the fine grid. In the following we assume that the grid spacing is doubled on the coarse grid. So counting only the unknowns but not the boundary points, we have only $(n_k + 1)/2 - 1$ variables in each direction on the coarse grid, while we have n_k variables on the fine grid. Under this assumption reasonable operators can be defined.

We begin with the restriction operators. To simplify the representation we use the stencil notation introduced in section 2.3.1. The meaning of a stencil for a restriction operator is that its elements define by which extent the elements of the fine grid contribute to the value on the coarse grid. The point at the center is the fine grid point that corresponds to the

current coarse grid point. To emphasize that the operator maps a vector from the fine grid using grid width h_k to the coarse grid with width h_{k-1} we add the k to the right bottom of the stencil and the $k-1$ to the right top.

Definition 3.9 *The* injection operator *is given by the stencil*

$$\begin{bmatrix} & & \\ & 1 & \\ & & \end{bmatrix}_k^{k-1}.$$

The injection operator is the most easy to implement operator and the computationally least expensive one, as only copying is involved. No floating point operations are needed. Due to this fact, it is an option for optimizing the computational cost of a multigrid cycle. Alternatively, in order to improve the representation of the error on the coarse grid we can distribute the values of a non-coarse grid point to its neighbors, which are part of the coarse grid, resulting in the *full-weighting operator*.

Definition 3.10 *By the stencil*

$$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}_k^{k-1}$$

we define the full-weighting operator.

A cheaper variant of the full-weighting operator is the *half-weighting operator*, which does not take grid points into account that have no neighbors belonging to the coarse grid in x- or y-direction.

Definition 3.11 *The* half-weighting operator *is given by the stencil*

$$\frac{1}{8}\begin{bmatrix} & 1 & \\ 1 & 4 & 1 \\ & 1 & \end{bmatrix}_k^{k-1}.$$

Of course, one can define three dimensional versions of these operators as well.

For prolongation we define the *bilinear interpolation*. In order to emphasize that it works in the opposite direction as the restriction, we denote its stencil with open brackets, i.e. $]\cdot[$, and we add the k and $k-1$ in reverse order. Intuitively this accentuates that the prolongation operator *gives* to the fine grid, while the restriction operator *takes* from the fine grid. With the help of the stencil notation we immediately obtain which share of a coarse grid point is distributed to which fine grid point. Again, the center point is the fine grid point that corresponds to the coarse grid point.

Definition 3.12 *The* bilinear interpolation operator *is given by the stencil*

$$\frac{1}{4}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}^{k}_{k-1}.$$

We would like to denote that the bilinear interpolation operator is the adjoint of the full-weighting operator up to a constant factor. This is an important feature in the context of the variational formulation of the multigrid theory that will be described later.

We denote the matrix representation of the restriction operator from the grid with grid spacing h_k to the grid with spacing h_{k-1} by $I_k^{k-1} \in \mathbb{R}^{\mathbf{n}_k \times \mathbf{n}_{k-1}}$. Analogously the matrix representation of the prolongation operator is denoted by $I_{k-1}^k \in \mathbb{R}^{\mathbf{n}_{k-1} \times \mathbf{n}_k}$.

Coarse grid correction operator

As the error is represented on the coarse grid reasonably well, the defect equation (3.7) is solved on the coarse grid. This is done by the coarse grid correction operator. The coarse grid correction consists of the following steps:

1. Compute residual: $\mathbf{r}_k \leftarrow \mathbf{f}_k - L_k \mathbf{u}_k$
2. Restrict residual: $\mathbf{r}_{k-1} \leftarrow I_k^{k-1} \mathbf{r}_k$
3. Solve defect equation: $\mathbf{e}_{k-1} \leftarrow L_{k-1}^{-1} \mathbf{r}_{k-1}$
4. Prolongate correction: $\mathbf{e}_k \leftarrow I_{k-1}^k \mathbf{e}_{k-1}$
5. Correct current approximation: $\mathbf{x}_k \leftarrow \mathbf{x}_k + \mathbf{e}_k$

Using this description we can define the coarse grid correction as a linear iterative method.

Definition 3.13 *Let L^k and L^{k-1} be two discretizations of the model problem as defined above. Let further I_k^{k-1} be a restriction operator and I_{k-1}^k be a prolongation operator. Then the coarse grid correction is defined as*

$$\phi_{CGC}^{(k)}(\mathbf{u}_k, \mathbf{f}_k) = \mathbf{u}_k + I_{k-1}^k L_{k-1}^{-1} I_k^{k-1} (\mathbf{f}_k - L_k \mathbf{u}_k).$$

An immediate consequence of this definition is the fact that the iteration matrix of the coarse grid correction is given by

$$T_k = I - I_{k-1}^k L_{k-1}^{-1} I_k^{k-1} L_k. \tag{3.8}$$

Remark 3.4 *The coarse grid correction is consistent with the linear system $L_k \mathbf{u}_k = \mathbf{f}_k$, but it is not convergent, as some eigenvalues are equal to one. The rank of the prolongation is at most $\mathbf{n}_{k-1}$.*

The twogrid cycle

Combining the smoother with the coarse grid correction yields the twogrid cycle.

Definition 3.14 *Let $\phi_S^{(k)}$ be an iterative method that smoothes the high frequencies of the error and let $\nu_1, \nu_2 \in \mathbb{N}$ be the number of presmoothing respectively postsmoothing iterations. Assume that $\phi_{CGC}^{(k)}$ is the coarse grid correction. Then the twogrid cycle with ν_1 presmoothing iterations and ν_2 postsmoothing iterations is given by*

$$\phi_{TGM}^{(k)}(\mathbf{u}_k, \mathbf{f}_k) = (\phi_S^{(k)})^{\nu_2}(\phi_{CGC}^{(k)}((\phi_S^{(k)})^{\nu_1}(\mathbf{u}_k, \mathbf{f}_k), \mathbf{f}_k), \mathbf{f}_k).$$

By this definition we obtain the iteration matrix of the twogrid cycle. Given the iteration matrix S_k of the smoother and the iteration matrix T_k of the coarse grid correction in (3.8) we obtain the iteration matrix

$$S_k^{\nu_2} T_k S_k^{\nu_1} = S_k^{\nu_2}(I - I_{k-1}^k L_{k-1}^{-1} I_k^{k-1} L_k) S_k^{\nu_1}.$$

The twogrid cycle in algorithmic form can be found in Algorithm 3.1.

Algorithm 3.1 Twogrid cycle $\mathbf{u}_k \leftarrow \phi_{\text{TGM}}^{(k)}(\mathbf{u}_k, \mathbf{f}_k)$

$\mathbf{u}_k \leftarrow (\phi_{\text{S}}^{(k)})^{\nu_1}(\mathbf{u}_k, \mathbf{f}_k)$
$\mathbf{r}_k \leftarrow \mathbf{f}_k - L_k \mathbf{u}_k$
$\mathbf{r}_{k-1} \leftarrow I_k^{k-1} \mathbf{r}_k$
$\mathbf{e}_{k-1} \leftarrow (L^{k-1})^{-1} \mathbf{r}_{k-1}$
$\mathbf{e}_k \leftarrow I_{k-1}^k \mathbf{e}_{k-1}$
$\mathbf{u}_k \leftarrow \mathbf{u}_k + \mathbf{e}_k$
$\mathbf{u}_k \leftarrow (\phi_{\text{S}}^{(k)})^{\nu_2}(\mathbf{u}_k, \mathbf{f}_k)$

Convergence of the two-grid cycle

There are various ways to prove convergence of the two-grid cycle in different settings. We will outline Hackbusch's proving technique here, as it is closely related to the proofs for algebraic multigrid convergence presented later. Other proof techniques include the use of Fourier transforms or the interpretation of multigrid methods as subspace correction methods. For an overview over these approaches we refer to the book of Trottenberg, Oosterlee and Schüller [84]. Hackbusch provides two properties that together give a sufficient criterion for the convergence of the twogrid method. These are the smoothing property and the approximation property.

The smoothing property is motivated by the fact that the error is smoothed as seen before. We have seen that the high frequencies are the eigenvectors belonging to the large eigenvalues. As a consequence we measure the smoothness of the error in terms of the L_k^2-norm.

So an iterative method $\phi_S^{(k)}$ is a good smoother if the L_k^2-norm of an arbitrary vector $\mathbf{e}_k$ after one iteration step is sufficiently smaller than before, i.e. if

$$\|S_k\mathbf{e}_k\|_{L_k^2} = \|L_kS_k\mathbf{e}_k\|_2 < \|L_k\mathbf{e}_k\|_2 = \|\mathbf{e}_k\|_{L_k^2}.$$

This motivates the following definition.

Definition 3.15 (Smoothing property) *An iterative method ϕ_S^k with iteration matrix S_k fulfills the* smoothing property*, if there exists a function $\eta(\nu)$, such that*

$$\|L_kS_k^\nu\|_2 \leq \eta(\nu)\|L_k\|_2 \text{ for all } 0 \leq \nu \leq \infty \text{ with } k \geq 0,$$
$$\lim_{\nu\to\infty} \eta(\nu) = 0.$$

It can be shown that for our model problem that the Richardson method [54] and the damped JOR method [52] satisfy the smoothing property with $\eta(\nu) = \nu^\nu/(\nu+1)^{\nu+1}$ and $\eta(\nu) = c/(\nu+\frac{1}{2})$, respectively.

Since the inverse of the operator is approximated on the coarse level, the approximation property is defined as a measure for the quality of this approximation.

Definition 3.16 (Approximation property) *Let I_{k-1}^k and I_k^{k-1} be the interpolation and restriction operators and let L^k be the discretization of the underlying partial differential equation as defined above. The twogrid method using these operators is said to fulfill the* approximation property*, if there exists a constant c, such that for all $k \in \mathbb{N}$ we have*

$$\|L_k^{-1} - I_{k-1}^k L_{k-1}^{-1} I_k^{k-1}\|_2 \leq \frac{c}{\|L_k\|_2}.$$

Various problems arising from the discretization of partial differential equations fulfill the approximation property, for details we refer to the work of Hackbusch [48, 49, 50, 51, 52, 53, 54].

Given the smoothing and the approximation property the twogrid method converges, as stated by the following theorem.

Theorem 3.6 *Let the twogrid method $\phi_{TGM,\nu,0}^{(k)}$ with ν presmoothing iterations of the iterative method $\phi_S^{(k)}$ fulfill the smoothing and the approximation property. Then for all $0 < \zeta < 1$ there exists a lower bound $\tilde{\nu}$, such that for all $\nu > \tilde{\nu}$ and for all $h < h_{max}$ we have*

$$\|T_kS_k^\nu\|_2 \leq c\eta(\nu) \leq \zeta.$$

Proof. Choose $\tilde{\nu}$ such that $\eta(\nu) \leq \frac{\zeta}{c}$ for all $\nu > \tilde{\nu}$. Then we have

$$\begin{aligned}
\|T_kS_k^\nu\|_2 &= \|(I - I_{k-1}^k L_{k-1}^{-1} I_k^{k-1} L_k)S_k^\nu\|_2 \\
&= \|(L_k^{-1} - I_{k-1}^k L_{k-1}^{-1} I_k^{k-1})L_kS_k^\nu\|_2 \\
&\leq \|(L_k^{-1} - I_{k-1}^k L_{k-1}^{-1} I_k^{k-1})\|_2 \|L_kS_k^\nu\|_2 \\
&\leq \frac{c}{\|L_k\|_2}\eta(\nu)\|L_k\|_2 = c\eta(\nu) \leq \zeta.
\end{aligned}$$

□

It is sufficient to analyze either pre- or post-smoothing here, as for two-grid methods the spectra of two methods having a different number ν_1 of pre-smoothing iterations and another number ν_2 of post-smoothing iterations but having the same sums $\nu_1 + \nu_2$ coincide, c.f. Lemma 4.4 in [74].

Now that we have defined everything we need for the twogrid method, and that we have given an overview over one proving technique for the convergence of the two grid method, we are ready to apply the same idea recursively, leading to multigrid methods.

3.2.3 Multigrid methods

The twogrid cycle provides a very efficient iterative method for the solution of linear systems arising in the discretization of partial differential equations. The most important feature is the h-independent convergence factor, a feature not provided by the previously considered methods. On the other hand the exact solution of the system on the coarse grid is needed to achieve that behavior. The direct solution on the coarse level is still very expensive, so iterative methods should be used to solve that system. Simple solvers like JOR still expose the same problem on the coarse grid as on the fine grid, although the problem is not as severe, since the smallest eigenvalue is larger on coarser grids. So we use γ iterations of a twogrid method on the coarse grid again, to solve the defect equation. This is a consistent application of the twogrid idea, leading to multigrid methods if applied recursively. On the coarsest level reached, a direct solver is used to solve the system. This coarsest level may contain one unknown only, so the direct solution on that system is computationally cheap. The multigrid cycle can then be defined recursively

Definition 3.17 *Let $\phi_S^{(k)}$ be a linear iterative method with iteration matrix S_k smoothing the high frequencies. Let $\nu_1, \nu_2 \in \mathbb{N}$ be the number of pre- and postsmoothing iterations and let $\gamma \in \mathbb{N}$ be the number of multigrid cycles used to solve* (3.7)*. Then the multigrid cycle is defined as*

$$\phi_{MGM}^{(0)}(\mathbf{u}_0, \mathbf{f}_0) = L_0^{-1}\mathbf{f}_0$$

for $k = 0$ and

$$\phi_{MGM}^{(k)}(\mathbf{u}_k, \mathbf{f}_k) = (\phi_S^{(k)})^{\nu_2}((\phi_S^{(k)})^{\nu_1}(\mathbf{u}_k, \mathbf{f}_k) + I_{k-1}^k((\phi_{MGM}^{(k-1)})^{\gamma}(\mathbf{0}, I_k^{k-1}(\mathbf{f}_k - L_k\mathbf{u}_k))), \mathbf{f}_k)$$

for $k = 1, 2, \ldots$.

With the help of Lemma 3.2 we immediately obtain the recursive definition of the iteration matrix M_k of the multigrid cycle.

$$M_k = \begin{cases} 0 & \text{for } k = 0 \\ S_k^{\nu_2}(I - I_{k-1}^k(I - (M_{k-1})^{\gamma})L_{k-1}^{-1}I_k^{k-1}L_k)S_k^{\nu_1} & \text{for } k = 1, 2, \ldots \end{cases}.$$

Given Definition 3.17 we can extend Algorithm 3.1 to Algorithm 3.2 for the multigrid cycle.

Algorithm 3.2 Multigrid cycle $\mathbf{u}_k \leftarrow \phi^{(k)}_{\text{MGM}}(\mathbf{u}_k, \mathbf{f}_k)$

$\mathbf{u}_k \leftarrow (\phi^{(k)}_{\text{S}})^{\nu_1}(\mathbf{u}_k, \mathbf{f}_k)$
$\mathbf{r}_k \leftarrow \mathbf{f}_k - L_k\mathbf{u}_k$
$\mathbf{r}_{k-1} \leftarrow I^{k-1}_k \mathbf{r}_k$
$\mathbf{e}_{k-1} \leftarrow \mathbf{0}$
if $k-1=0$ **then**
  $\mathbf{e}_0 \leftarrow L_0^{-1}\mathbf{r}_0$
else
  for $i=1$ to γ **do**
    $\mathbf{e}_{k-1} \leftarrow \phi^{(k-1)}_{\text{MGM}}(\mathbf{e}_{k-1}, \mathbf{r}_{k-1})$
  end for
end if
$\mathbf{e}_k \leftarrow I^k_{k-1}\mathbf{e}_{k-1}$
$\mathbf{u}_k \leftarrow \mathbf{u}_k + \mathbf{e}_k$
$\mathbf{u}_k \leftarrow (\phi^{(k)}_{\text{S}})^{\nu_2}(\mathbf{u}_k, \mathbf{f}_k)$

V-cycles and W-cycles

Depending on how often we apply the twogrid cycle to solve the defect equation (3.7), we get different types of multigrid cycles. They are named according to the following definition.

Definition 3.18 *Depending on the number γ of multigrid cycles recursively used to solve the defect equation* (3.7) *on the coarse grid, the multigrid cycle is called* V-cycle, *for $\gamma = 1$ or* W-cycle *for $\gamma = 2$. We denote the V-cycle multigrid operator by $\phi^{(k)}_{\text{V}}$ and the W-cycle operator by $\phi^{(k)}_{\text{W}}$.*

Computational complexity

We will now discuss the computational complexity of different values of γ according to [84], especially of the V- and the W-cycles. We will stick to our standard 2D problem, i.e. we assume that the grid spacing is doubled on each level. Now we can derive the number of arithmetical operations for each multigrid cycle. We define W_k to be the number of arithmetical operations needed for a multigrid cycle starting on level k. Further on we define $\tilde{W}_k$ to be the number of arithmetical operations needed on level k, excluding the solution of the defect equation using the recursive application of the multigrid cycle. Thus we get

$$W_1 = \tilde{W}_1 + W_0 \qquad\qquad W_{k+1} = \tilde{W}_{k+1} + W_k,\ k = 1, 2, \ldots.$$

From that we obtain

$$W_k = \sum_{l=1}^{k-1} \gamma^{k-l} \tilde{W}_l + \gamma^{k-1} W_0 \tag{3.9}$$

Again we let $\mathbf{n}_k$ be the number of unknowns on level k. Neglecting boundary effects we have that $\mathbf{n}_k = \frac{1}{4}\mathbf{n}_{k+1}$. For the work on each level excluding the solution of the defect equation we have $\tilde{W}_k \leq c\mathbf{n}_k$, where c is a small constant independent of $\mathbf{n}_k$. So from (3.9) we get

$$\begin{aligned} W_k &= \sum_{l=1}^{k-1} \gamma^{k-l} \tilde{W}_l + \gamma^{k-1} W_0 \\ &\leq \sum_{l=1}^{k-1} \gamma^{k-l} \left(\frac{1}{4}\right)^{k-l} c\mathbf{n}_k + \gamma^{k-1} W_0 \\ &= c\mathbf{n}_k \sum_{l=1}^{k-1} \left(\frac{\gamma}{4}\right)^{l} + \gamma^{k-1} W_0 \end{aligned}$$

The last summand grows logarithmically with the number of unknowns on the finest grid, the first summand is a geometric series, so we can subsume

$$W_k \leq \begin{cases} \frac{4}{3} c\mathbf{n}_k + \mathcal{O}(\log \mathbf{n}_k) & \text{for } \gamma = 1, \\ 2c\mathbf{n}_k + \mathcal{O}(\log \mathbf{n}_k) & \text{for } \gamma = 2, \\ 4c\mathbf{n}_k + \mathcal{O}(\log \mathbf{n}_k) & \text{for } \gamma = 3. \end{cases}$$

For $\gamma = 4$ the work on each level is constant, as the number of unknowns is quartered up to boundary effects but we spend 4 cycles on each level, so the advantage of quartering the number of unknowns is lost. As the number of levels is an order $\log(\mathbf{n}_k)$-term, we then have a complexity of $\mathcal{O}(\mathbf{n}_k \log \mathbf{n}_k)$. We like to conclude mentioning that the computational complexity depends on the reduction r of the number of unknowns going from level k to level $k-1$, on the complexity c_k per unknown, which may grow while going to a coarser level, and on the number of recursive applications of multigrid cycles γ. As long $\gamma r c_k < 1$ we have linear complexity.

Convergence of the W-cycle

Now we have that the twogrid method converges and that one multigrid cycle is computationally optimal, it remains to show that a multigrid method converges where the convergence rate is bounded from above by a bound that is independent of the number of unknowns. A multigrid method can be interpreted as a twogrid method, where the defect equation (3.7) is solved only approximately. This approximate solution is calculated using a multigrid method, which is an iterative method. Under the assumption, that the twogrid

convergence rate is bounded for all grid spacings and that the involved prolongation, restriction and smoothing operators are bounded as well, we can derive that the multigrid method converges uniformly for $\gamma \geq 2$, i.e. that independent of the number of unknowns the convergence rate is bounded from above.

Theorem 3.7 *Let*

$$\|S_k^{\nu_2} T_k S_k^{\nu_1}\|_* \leq \sigma, \qquad \|S_k^{\nu_2} I_{k-1}^k\|_* \|L_{k-1}^{-1} I_k^{k-1} L_k S_k^{\nu_1}\|_* \leq c$$

hold uniformly for all grid spacings h for some norm $\|\cdot\|_*$. Then the $*$-norm of the iteration matrix M_k is bounded by η^k, where η^k is defined recursively as

$$\eta_0 = \sigma, \qquad \eta_k = \sigma + c\eta_{k-1}^\gamma \quad (k = 1, 2, \dots), \tag{3.10}$$

where $c, \sigma > 0$. For $\gamma = 2$ and

$$4c\sigma \leq 1$$

the $*$-norm of the iteration matrix M_k is bounded from above by

$$\|M_k\|_* \leq \eta = \frac{1}{2c}(1 - \sqrt{1 - 4c\sigma}) \leq 2\sigma,$$

so for $\sigma < \frac{1}{2}$ the method converges with a uniformly bounded convergence rate.

Proof. First we show that the norm of the iteration matrix is bound by η_k as defined in (3.10). We have

$$\begin{aligned}
\|M_k\|_* &= \|S_k^{\nu_2}(I - I_{k-1}^k(I - M_{k-1}^\gamma)L_{k-1}^{-1} I_k^{k-1} L_k) S_k^{\nu_1}\|_* \\
&= \|S_k^{\nu_2}(I - I_{k-1}^k L_{k-1}^{-1} I_k^{k-1} L_k) S_k^{\nu_1} + S_k^{\nu_2} I_{k-1}^k M_{k-1}^\gamma L_{k-1}^{-1} I_k^{k-1} L_k S_k^{\nu_1}\|_* \\
&\leq \|S_k^{\nu_2}(I - I_{k-1}^k L_{k-1}^{-1} I_k^{k-1} L_k) S_k^{\nu_1}\|_* + \|S_k^{\nu_2} I_{k-1}^k M_{k-1}^\gamma L_{k-1}^{-1} I_k^{k-1} L_k S_k^{\nu_1}\|_* \\
&\leq \|S_k^{\nu_2}(I - I_{k-1}^k L_{k-1}^{-1} I_k^{k-1} L_k) S_k^{\nu_1}\|_* + \|S_k^{\nu_2} I_{k-1}^k\|_* \|M_{k-1}^\gamma\|_* \|L_{k-1}^{-1} I_k^{k-1} L_k S_k^{\nu_1}\|_* \\
&= \sigma + c\eta_{k-1}^\gamma.
\end{aligned}$$

Now $\gamma = 2$ and forming the limit yields $\eta = \sigma + c\eta^2$. So for $4c\sigma \leq 1$ we have

$$\eta = \frac{1}{2c}(1 - \sqrt{1 - 4c\sigma}) = \frac{1 - \sqrt{1 - 4c\sigma}}{4c\sigma} 2\sigma \leq 2\sigma,$$

since

$$\begin{aligned}
& 1 - 4c\sigma \leq \sqrt{1 - 4c\sigma} \\
\Leftrightarrow \quad & 1 - \sqrt{1 - 4c\sigma} \leq 4c\sigma \\
\Leftrightarrow \quad & \frac{1 - \sqrt{1 - 4c\sigma}}{4c\sigma} \leq 1.
\end{aligned}$$

Obviously $\eta_0 = \sigma \leq (1 - \sqrt{1 - 4c\sigma})/(2c)$. Assuming that $\eta_{k-1} \leq (1 - \sqrt{1 - 4c\sigma})/(2c)$ we have

$$\begin{aligned}
\eta_k &\leq \sigma + c\eta_{k-1}^2 \\
&\leq \sigma + c\left(\frac{1 - \sqrt{1 - 4c\sigma}}{2c}\right)^2 \\
&= \sigma + \frac{1}{4c}(1 - 2\sqrt{1 - 4c\sigma} + 1 - 4c\sigma) \\
&= \frac{1 - \sqrt{1 - 4c\sigma}}{2c}.
\end{aligned}$$

So $(1 - \sqrt{1 - 4c\sigma})/(2c)$ is an upper bound for the convergence rate of the W-cycle. □

So under a few additional assumptions the convergence of the multigrid method is a consequence of the convergence of the twogrid method. The convergence of the V-cycle requires more advanced techniques of proof. As we will present an algebraic proof of the convergence of the V-cycle later, for proofs that are more related to geometric multigrid we refer to the work of Braess and Hackbusch [10] and to the book of Trottenberg, Oosterlee and Schüller [84].

3.2.4 FAS and FAC

While multigrid methods originally have been developed for the use of linear problems, they have been adopted to non-linear problems as well. We will not deal with non-linear problems here, but we need some ideas from the full approximate storage approach in order to motivate a multigrid technique that efficiently solves problems with local grid refinements. This will allow us to define a fast multigrid method for the solution of the system resulting from the hierarchical grid refinement introduced in section 2.3.2. When dealing with non-linear problems the solution of the defect equation (3.7) is not feasible, as the correction carried out later directly depends on the linearity of the operator, i.e. we make use of

$$\mathbf{u}_k^* = L_k^{-1} L_k(\mathbf{u}_k + (\mathbf{u}_k^* - \mathbf{u}_k)) = \mathbf{u}_k + L_k^{-1}(\mathbf{f}_k - L_k\mathbf{u}_k) = \mathbf{u}_k + L_k^{-1}\mathbf{r}_k.$$

This is obviously not possible for the solution of non-linear problems. To avoid this, we rather transfer the current approximation to the coarse level. We compute a new approximate solution on the coarse level using the restricted current approximation as a start value. The right hand side is constructed as the sum of the current restricted fine level residual and the operator applied to the restricted current fine level approximation. Then we subtract the restricted fine level solution from the new coarse level solution in order to get a correction. That correction is then transferred to the fine level and added to the current approximate

solution on that level. That way we avoided using the linearity of the operator, nevertheless the resulting method is equivalent to the unmodified multigrid cycle for linear operators. So we define the full approximate storage cycle in accordance to the multigrid cycle.

Definition 3.19 *Let $\phi_S^{(k)}$ be an iterative method smoothing the high frequencies. Let $\nu_1, \nu_2 \in \mathbb{N}$ be the number of pre- and postsmoothing iterations and let $\gamma \in \mathbb{N}$ be the number of recursive calls used to solve the coarse level system. Then the* full approximate storage cycle *or* FAS cycle *is defined as*

$$\phi_{FAS}^{(0)}(\mathbf{u}_0, \mathbf{f}_0) = L_0^{-1}\mathbf{f}_0$$

for $k = 0$ and

$$\begin{aligned}\phi_{FAS}^{(k)}(\mathbf{u}_k, \mathbf{f}_k) = (\phi_S^{(k)})^{\nu_2}((\phi_S^{(k)})^{\nu_1}(\mathbf{u}_k, \mathbf{f}_k) + I_{k-1}^{k}((\phi_{FAS}^{(2h)})^{\gamma}(I_k^{k-1}(\phi_S^{(k)})^{\nu_1}(\mathbf{u}_k, \mathbf{f}_k),\\ I_k^{k-1}(\mathbf{f}_k - L_k\mathbf{u}_k) + L_{k-1}I_k^{k-1}(\phi_S^{(k)})^{\nu_1}(\mathbf{u}_k, \mathbf{f}_k)) - I_k^{k-1}(\phi_S^{(k)})^{\nu_1}(\mathbf{u}_k, \mathbf{f}_k)), \mathbf{f}_k)\end{aligned}$$

for $k = 1, 2, \ldots$.

The implementation can be found in Algorithm 3.3.

Algorithm 3.3 FAS cycle $\mathbf{u}_k \leftarrow \phi_{\text{FAS}}^{(k)}(\mathbf{u}_k, \mathbf{f}_k)$

$\mathbf{u}_k \leftarrow (\phi_{\text{S}}^{(k)})^{\nu_1}(\mathbf{u}_k, \mathbf{f}_k)$
$\mathbf{d}_k \leftarrow \mathbf{f}_k - L_k\mathbf{u}_k$
$\mathbf{d}_{k-1} \leftarrow I_k^{k-1}\mathbf{d}_k$
$\mathbf{u}_{k-1} \leftarrow I_k^{k-1}\mathbf{u}_k$
$\mathbf{f}_{k-1} \leftarrow \mathbf{d}_{k-1} + L_{k-1}\mathbf{u}_{k-1}$
$\mathbf{v}_{k-1} \leftarrow \mathbf{u}_{k-1}$
if $k - 1 = 0$ **then**
  $\mathbf{v}_0 \leftarrow L_0^{-1}\mathbf{f}_0$
else
  for $i = 1$ to γ **do**
    $\mathbf{v}_{k-1} \leftarrow \phi_{\text{FAS}}^{(2h)}(\mathbf{v}_{k-1}, \mathbf{f}_{k-1})$
  end for
end if
$\mathbf{v}_{k-1} \leftarrow \mathbf{v}_{k-1} - \mathbf{u}_{k-1}$
$\mathbf{v}_k \leftarrow I_{k-1}^{k}\mathbf{v}_{k-1}$
$\mathbf{u}_k \leftarrow \mathbf{u}_k + \mathbf{v}_k$
$\mathbf{u}_k \leftarrow (\phi_{\text{S}}^{(k)})^{\nu_2}(\mathbf{u}_k, \mathbf{f}_k)$

In Section 2.3.2 we extended the hierarchical refined grid discretization for the solution of the Poisson equation in free space. The multigrid method just developed is directly applicable to solve the system. If that approach is chosen to solve the system, on each level the composite grid up to that level would have to be used. We notice that the parts that

are not refined will not benefit a lot from the solution on a finer level, as they are already treated properly on the lower levels. So we only apply the smoother on the finer levels to that part of the grid that is discretized using the current finest grid size. The only remaining question is then how to treat the correction. In the standard V-cycle the defect equation is solved on the coarse level, so Dirichlet zero boundary conditions are used. This is not an option as parts of the information on the current approximation is contained in the coarse grid approximation, only. So we use the FAS cycle, i.e. we transfer our current residual plus the discretized operator applied to our current approximation to the coarse level and solve the system there. The correction is then formed as described above and our current approximation is updated. This technique is an application of McCormick's *fast adaptive composite grid method* (FAC) [66, 65]. Washio and Oosterlee used the *multilevel adaptive technique* (MLAT) by Brandt [11, 13] that involves high order interpolation constructed from the discretization at the interface in their work [87]. A more general approach to adaptive multigrid methods can be found in the work of Rüde [68].

3.3 Algebraic Multigrid Theory for Structured Matrices

While geometric multigrid methods are easy to develop for problems arising from partial differential equations with simple geometries, it can be very hard to generate a grid hierarchy for more complex geometries. The problem is to find coarser levels for the multigrid method. While in most cases it is easy to provide a finer discretization for a given geometry which is already discretized, it can be very hard to find a reasonable coarser discretization. Therefore the problem on the coarsest level might still be too expensive to be solved directly. Another problem exists when geometry information is not available at all, which might be the case if multigrid should be used as a black box solver, for example in a commercial code, or when the underlying problem is not geometric at all. To tackle these problems algebraic multigrid methods, or AMG methods for short, have been developed as black box multigrid solvers. Unlike in geometric multigrid methods, in algebraic multigrid methods the smoother is fixed and the coarsening process is fully automatic, i.e. given a matrix the interpolation and restriction operators are constructed such that the resulting method converges. Due to the construction of the coarser levels the algebraic multigrid methods can be split into a setup phase and a solution phase. One of the main concerns by AMG critics is the setup phase, as it can be quite expensive. Additionally, the coarse level construction is hard to parallelize. Nevertheless AMG allows the use of multigrid methods where it would not be possible at all to use a geometric multigrid method. The standard algebraic multigrid theory is valid for M-matrices. Introductions to algebraic multigrid can be found in the book chapter by Ruge and Stüben [69], in the appendix written by Stüben [78] or in his reports [77, 76].

The rest of this section is structured as follows: We will first give an overview over the convergence theory for hermitian positive definite problems. After that we will present

some theory regarding the replacement of the Galerkin operator that has been developed during the work leading to this thesis. Finally, we will present multigrid methods for matrices from matrix algebras and the application of the new theory to the circulant case.

3.3.1 Convergence theory for multigrid methods for hermitian positive definite problems

The following presentation of the convergence theory is similar to the one in the book chapter of Ruge and Stüben [69], parts are clarified in the introduction of Stüben [78]. Their theory is based on the works of Brandt [14], Mandel [63], McCormick [64] and others.

Basic definitions and results

While in the presentation of geometric multigrid methods we denoted the matrices by L, the right hand sides by $\mathbf{f}$ and the solutions by $\mathbf{u}$ as they are connected to partial differential equations, we will now use A, $\mathbf{b}$ and $\mathbf{x}$, respectively, again to underline, that the presented theory is not only applicable to problems resulting from the discretization of partial differential equations, but rather applicable to classes of problems, where only the algebraic properties of the associated system matrices are of interest. We are interested in the solution of the system

$$A\mathbf{x} = \mathbf{b},$$

$\mathbf{A} \in \mathbb{C}^{n\times n}$ hermitian and positive definite and $\mathbf{x}, \mathbf{b} \in \mathbb{C}^n$ using a multigrid method. For that purpose we assume that a sequence of systems of equations

$$A_k\mathbf{x}_k = \mathbf{b}_k,$$

with the corresponding sequences of dimensions $\{n_k\}_{k=1}^{k_{\max}}, n_k \in \mathbb{N}$, system matrices $\{A_k\}_{k=1}^{k_{\max}}, A_k \in \mathbb{C}^{n_k\times n_k}$, hermitian and positive definite, right hand side vectors $\{\mathbf{b}_k\}_{k=1}^{k_{\max}}, \mathbf{b}_k \in \mathbb{C}^{n_k}$ and solution vectors $\{\mathbf{x}_k\}_{k=1}^{k_{\max}}, \mathbf{x}_k \in \mathbb{C}^{n_k}$ exists, where

$$A_{k_{\max}} = A, \qquad \mathbf{x}_{k_{\max}} = \mathbf{x}, \qquad \mathbf{b}_{k_{\max}} = \mathbf{b}.$$

Furthermore we assume the existence of prolongation operators

$$P_k \in \mathbb{C}^{n_k\times n_{k-1}}, k = 1, \ldots, k_{\max}$$

and restriction operators

$$R_k \in \mathbb{C}^{n_{k-1}\times n_k}, k = 1, \ldots, k_{\max}.$$

Besides these transfer operators we let ϕ_S^k be a linear iterative method with iteration matrix M_S that is used as a smoother. In analogy to Definition 3.13 we define the coarse grid correction.

Definition 3.20 *Let $A_k \in \mathbb{C}^{n_k \times n_k}, A_{k-1} \in \mathbb{C}^{n_{k-1} \times n_{k-1}}$ be two system matrices, let $P_k \in \mathbb{C}^{n_k \times n_{k-1}}$ be the prolongation operator from level $k-1$ to level k and let $R_k \in \mathbb{C}^{n_{k-1} \times n_k}$ be the restriction operator from level k to level $k-1$. Then the coarse grid correction is defined as*

$$\phi_{CGC}^{(k)}(\mathbf{x}_k, \mathbf{b}_k) = \mathbf{x}_k + P_k A_{k-1}^{-1} R_k(\mathbf{b}_k - A_k \mathbf{x}_k).$$

The iteration matrix T_k is given by

$$T_k = I - P_k A_{k-1}^{-1} R_k A_k. \tag{3.11}$$

In the same fashion we define the twogrid method and the multigrid method on the basis of Definition 3.14 and 3.17, respectively, depending on the definition of the coarse grid correction just given.

Definition 3.21 *Let $\phi_S^{(k)}$ be a linear iterative method that is used as a smoother and let $\nu_1, \nu_2 \in \mathbb{N}$ be the number of presmoothing respectively postsmoothing iterations. Assume that $\phi_{CGC}^{(k)}$ is the coarse grid correction. Then the twogrid cycle with ν_1 presmoothing iterations and ν_2 postsmoothing iterations is given by*

$$\phi_{TGM}^{(k)}(\mathbf{x}_k, \mathbf{b}_k) = (\phi_S^{(k)})^{\nu_2}(\phi_{CGC}^{(k)}((\phi_S^{(k)})^{\nu_1}(\mathbf{x}_k, \mathbf{b}_k), \mathbf{b}_k), \mathbf{b}_k).$$

Definition 3.22 *Let $\phi_S^{(k)}$ be an iterative method used as a smoother. Let $\nu_1, \nu_2 \in \mathbb{N}$ be the number of pre- and postsmoothing iterations and let $\gamma \in \mathbb{N}$ be the number of multigrid cycles used to solve the defect equation*

$$A_k \mathbf{e}_k = \mathbf{r}_k.$$

Then the multigrid cycle is defined as

$$\phi_{MGM}^{(0)}(\mathbf{x}_0, \mathbf{b}_0) = A_0^{-1} \mathbf{b}_0$$

for $k = 0$ and

$$\phi_{MGM}^{(k)}(\mathbf{x}_k, \mathbf{b}_k) = (\phi_S^{(k)})^{\nu_2}((\phi_S^{(k)})^{\nu_1}(\mathbf{x}_k, \mathbf{b}_k) + P_k((\phi_{MGM}^{(k-1)})^{\gamma}(\mathbf{0}, R_k(\mathbf{b}_k - A_k \mathbf{x}_k))), \mathbf{b}_k)$$

for $k = 1, \ldots, k_{max}$.

In analogy to the Definition 3.15 and 3.16, we define the smoothing property and the approximation property. For that purpose we need an arbitrary norm that has to be the same in both definitions. That norm will be denoted by $\|\cdot\|_*$. In the classical work of Ruge and Stüben the energy norm with respect to $A_k \operatorname{diag}(A_k)^{-1} A_k$ is used. Aricò and Donatelli noted in [2] that this choice is not necessary, as long as the same norm is used in both properties.

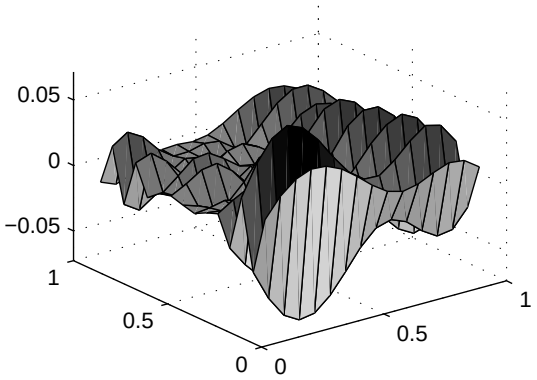

Figure 3.3: Algebraically smooth error of a mixture of a differential equation in x-direction and an integral equation in y-direction after application of 10 iteration of the JOR method with $\omega = 4/5$.

Definition 3.23 *An iterative method $\phi_S^{(k)}$ with iteration matrix S_k fulfills the* smoothing property *if there exists an $\alpha > 0$ such that for all $\mathbf{e}_k \in \mathbb{C}^{n_k}$ it holds*

$$\|S_k \mathbf{e}_k\|_{A_k}^2 \leq \|\mathbf{e}_k\|_{A_k}^2 - \alpha \|\mathbf{e}_k\|_*^2. \tag{3.12}$$

We like to note that this definition of smoothness does not necessarily mean that an error is geometrically smooth. As an example consider a problem similar to the model problem that is discretized on the unit square and described by the stencil

$$\begin{bmatrix} & 1 & \\ -1 & 4 & -1 \\ & 1 & \end{bmatrix}.$$

A plot of the error of the JOR-method after a couple of iterations can be be found in Figure 3.3. Although the error is smooth regarding the previous definition, it is geometrically highly oscillatory, so we prefer to call the error *algebraically smooth*. An error that is algebraically smooth fulfills the property that the $*$-norm of the error is small compared to the A_k-norm. We now continue with the definition of the approximation property.

Definition 3.24 *Let T_k be the iteration matrix of the coarse grid correction $\phi_{CGC}^{(k)}$. If there exists a β for all $\mathbf{e}_k \in \mathbb{C}^{n_k}$ such that*

$$\|T_k \mathbf{e}_k\|_{A_k}^2 \leq \beta \|\mathbf{e}_k\|_*^2, \tag{3.13}$$

then $\phi_{CGC}^{(k)}$ fulfills the approximation property.

Combining the smoothing and the approximation property yields the convergence of the twogrid method using postsmoothing, only, as stated by the following lemma.

Lemma 3.3 *Let $\phi_S^{(k)}$ be an iterative method with iteration matrix S_k fulfilling the smoothing property with some norm $\|\cdot\|^*$ and let $\phi_{CGC}^{(k)}$ be the coarse grid correction fulfilling the approximation property using the same norm, denoting its iteration matrix by T_k and let $T_k \leq_{A_k} I$ hold. Then we have*

$$\beta \geq \alpha$$

and

$$\|S_k T_k\|_{A_k}^2 \leq \sqrt{1-\alpha/\beta}\|T_k\|_{A_k}^2.$$

Proof.

$$\begin{aligned}\|S_k T_k \mathbf{e}_k\|_{A_k}^2 &\leq \|T_k\mathbf{e}_k\|_{A_k}^2 - \alpha\|T_k\mathbf{e}_k\|_*^2 \\ &\leq \|T_k\mathbf{e}_k\|_{A_k}^2 - \alpha/\beta\|T_k^2\mathbf{e}_k\|_{A_k}^2 \\ &\leq (1-\alpha/\beta)\|\mathbf{e}_k\|_{A_k}^2\end{aligned}$$

This proves $\beta \geq \alpha$. □

So for $\sqrt{1-\alpha/\beta} < 1$ we have a convergent twogrid method.

Variational property of the coarse grid correction using the Galerkin operator on the coarser level

For the theoretical considerations we first consider the Galerkin operator as the operator on the coarse grid, only. It is given by the following definition.

Definition 3.25 *Let $A_k \in \mathbb{C}^{n_k \times n_k}$ be the system matrix of level k, P_k the related projection operator and R_k the related restriction operator. Then we define the* Galerkin operator *as*

$$A_{k-1} = R_k A_k P_k.$$

In the following we are only treating hermitian matrices and we define the projection to be the adjoint of the restriction, i.e.

$$P_k = R_k^H.$$

Methods using the Galerkin operator on the coarser level have some nice properties, since due to the use of the Galerkin operator the iteration matrix T_k of the coarse grid correction is an A_k-orthogonal projector.

Definition 3.26 *Let $A \in \mathbb{C}^{n\times n}$ be a hermitian positive definite matrix. Then a matrix $Q \in \mathbb{C}^{n\times n}$ is called A-*orthogonal projector*, if Q is symmetric with respect to the scalar product induced by A, i.e. for all $\mathbf{x}, \mathbf{y} \in \mathbb{C}^n$ we have*

$$\langle Q\mathbf{x}, \mathbf{y}\rangle_A = \mathbf{x}^H Q^H A\mathbf{y} = \mathbf{x}AQ\mathbf{y} = \langle \mathbf{x}, Q\mathbf{y}\rangle_A,$$

and if $Q^2 = Q$.

T_k with the Galerkin operator on the coarse level and the adjoint of the restriction operator as prolongation operator is an A_k-orthogonal projector.

Lemma 3.4 *Let $A_k \in \mathbb{C}^{n_k \times n_k}$ be an hermitian positive definite matrix. Then T_k as given by* (3.11) *with the Galerkin operator on the coarse level and the adjoint of the restriction operator as prolongation operator is an A_k-orthogonal projector. Further we have*

$$\operatorname{ran}(I - T_k) = \operatorname{ran}(P_k). \tag{3.14}$$

Proof. Equation (3.14) is obvious for a projection having full rank. Regarding the first part we have

$$\begin{aligned}
T_k^2 &= (I - P_k A_{k-1}^{-1} R_k A_k)^2 \\
&= I - P_k A_{k-1}^{-1} R_k A_k - P_k A_{k-1}^{-1} R_k A_k \\
&\quad + P_k A_{k-1}^{-1} R_k A_k P_k A_{k-1}^{-1} R_k A_k \\
&= I - P_k A_{k-1}^{-1} R_k A_k - P_k A_{k-1}^{-1} R_k A_k \\
&\quad + P_k A_{k-1}^{-1} R_k A_k P_k (R_k A_k P_k)^{-1} R_k A_k \\
&= I - P_k A_{k-1}^{-1} R_k A_k - P_k A_{k-1}^{-1} R_k A_k + P_k A_{k-1}^{-1} R_k A_k \\
&= I - P_k A_{k-1}^{-1} R_k A_k \\
&= T_k.
\end{aligned}$$

Now for all $\mathbf{x}, \mathbf{y} \in \mathbb{C}^{n_k}$

$$\begin{aligned}
\mathbf{x}^H T_k^H A_k \mathbf{y} &= \mathbf{x}^H (I - A_k P_k A_{k-1}^{-1} R_k) A_k \mathbf{y} \\
&= \mathbf{x}^H A_k (I - P_k A_{k-1}^{-1} R_k A_k) \mathbf{y} \\
&= \mathbf{x}^H A_k T_k \mathbf{y},
\end{aligned}$$

which completes the proof. □

We like to recall some properties of orthogonal projectors:

Lemma 3.5 *Let $A \in \mathbb{C}^{n \times n}$ be a hermitian positive definite matrix and let $Q \in \mathbb{C}^{n \times n}$ be an A-orthogonal projector. Then the following holds true:*

1. $\operatorname{ran}(Q) \perp_A \operatorname{ran}(I - Q)$.
2. *For all $\mathbf{u} \in \operatorname{ran}(Q)$ and for all $\mathbf{u} \in \operatorname{ran}(I - Q)$ it holds $\|\mathbf{u} + \mathbf{v}\|_A^2 = \|\mathbf{u}\|_A^2 + \|\mathbf{v}\|_A^2$.*
3. $\|Q\|_A = 1$.
4. *For all $\mathbf{u} \in \mathbb{C}^n$ we have $\|Q\mathbf{u}\|_A^2 = \min\limits_{\mathbf{v} \in \operatorname{ran}(I-Q)} \|\mathbf{u} - \mathbf{v}\|_A^2$.*

Proof. The first statement holds, as for all $\mathbf{u}, \mathbf{v} \in \mathbb{C}$ we have

$$\langle Q\mathbf{u}, (I - Q)\mathbf{v} \rangle_A = \langle \mathbf{u}, Q(I - Q)\mathbf{v} \rangle_A = \langle \mathbf{u}, \mathbf{0} \rangle_A = 0,$$

the second statement is an immediate consequence of this observation. For the third statement we have

$$\|Q\|_A^2 = \sup_{\mathbf{u}\neq\mathbf{0}} \frac{\|Q\mathbf{u}\|_A^2}{\|\mathbf{u}\|_A^2} = \sup_{\mathbf{u}\neq\mathbf{0}} \frac{\|Q\mathbf{u}\|_A^2}{\|Q\mathbf{u}\|_A^2 + \|(I-Q)\mathbf{u}\|_A^2} \leq 1.$$

Choosing $\mathbf{u} \in \mathrm{ran}(Q)$ yields $\|Q\|_A = 1$. For the last statement the following holds true:

$$\begin{aligned}
\min_{\mathbf{v}\in\mathrm{ran}(I-Q)} \|\mathbf{u}-\mathbf{v}\|_A^2 &= \min_{\mathbf{v}\in\mathrm{ran}(I-Q)} \|Q\mathbf{u} + (I-Q)\mathbf{u} - \mathbf{v}\|_A^2 \\
&= \min_{\mathbf{v}\in\mathrm{ran}(I-Q)} \|Q\mathbf{u} - \mathbf{v}\|_A^2 \\
&= \min_{\mathbf{v}\in\mathrm{ran}(I-Q)} (\|Q\mathbf{u}\|_A^2 + \|\mathbf{v}\|_A^2) \\
&= \|Q\mathbf{u}\|_A^2.
\end{aligned}$$

□

A consequence of these basic properties of the coarse grid correction is that it fulfills a variational property regarding $\mathrm{ran}(P_k)$, i.e. minimizes the A-norm of the error with respect to all variations in $\mathrm{ran}(P_k)$, as due to the last statement of the previous lemma we have for all $\mathbf{e}_k \in \mathbb{C}_{n_k}$

$$\|T_k\mathbf{e}_k\|_{A_k}^2 = \min_{\mathbf{e}_{k-1}\in\mathrm{ran}(P_k)} \|\mathbf{e}_k - \mathbf{e}_{k-1}\|_{A_k}^2.$$

For methods involving the Galerkin operator on the coarse grid the Lemma 3.3 holds as $\|T_k\|_{A_k} = 1$, so the two-grid method converges. We now carry over the convergence result to the multigrid case.

Theorem 3.8 *Let T_k be the coarse grid correction with iteration matrix T_k, using the Galerkin operator $A_{k-1} = R_k A_k P_k$ on the coarser level and the adjoint of the restriction as prolongation, i.e. $P_k = R_k^H$. Now we assume a coarse grid correction $\bar{\phi}_{CGC}^{(k)}$ where we solve the defect equation not directly, but rather with a linear iterative method*

$$\bar{\phi}^{(k-1)}(\mathbf{x}_{k-1}, \mathbf{b}_{k-1}) = \bar{M}_{k-1}\mathbf{x}_{k-1} + \bar{N}_{k-1}\mathbf{x}_{k-1},$$

using zero as start approximation and assume furthermore that

$$\bar{\eta} := \|I - \bar{N}_{k-1}A_{k-1}\|_{A_{k-1}} < 1, \tag{3.15}$$

that $\phi_S^{(k)}$ fulfills the smoothing property (3.12) *and that $\phi_{CGC}^{(k)}$ fulfills the approximation property* (3.13)*. Then the (post-smoothing) two grid method using the modified coarse grid correction $\bar{T}_k$ using the zero initial approximation, i.e.*

$$\bar{T}_k = I - P_k\bar{N}_{k-1}R_kA_k,$$

converges with convergence factor of at most $\max\{\bar{\eta}, \sqrt{1-\delta}\}$*, i.e.*

$$\|S_k^{\nu_2}\bar{T}_k\mathbf{e}_k\|_{A_k} \leq \max\{\bar{\eta}, \sqrt{1-\delta}\}\|\mathbf{e}_k\|_{A_k},$$

where $\delta = \alpha/\beta$ *with* α *and* β *from the smoothing and approximation property.*

Proof. Given a fine level error $\mathbf{e}_k$ we define the coarse level defects as

$$\begin{aligned} A_{k-1}\mathbf{d}_{k-1} &= R_k A_k \mathbf{e}_k \\ \text{respectively} \qquad \bar{\mathbf{d}}_{k-1} &= \bar{N}_{k-1} R_k A_k \mathbf{e}_k. \end{aligned}$$

Thus with (3.15) for the error of the approximate defect we can write

$$\begin{aligned} \|\mathbf{d}_{k-1} - \bar{\mathbf{d}}_{k-1}\|_{A_{k-1}} &= \|A_{k-1}^{-1}R_kA_k\mathbf{e}_k - \bar{N}_{k-1}R_kA_k\mathbf{e}_k\|_{A_{k-1}} \\ &= \|A_{k-1}^{-1}R_kA_k\mathbf{e}_k - \bar{N}_{k-1}A_{k-1}A_{k-1}^{-1}R_kA_k\mathbf{e}_k\|_{A_{k-1}} \\ &= \|(I - \bar{N}_{k-1}A_{k-1})A_{k-1}^{-1}R_kA_k\mathbf{e}_k\|_{A_{k-1}} \\ &\leq \|I - \bar{N}_{k-1}A_{k-1}\|_{A_{k-1}}\|A_{k-1}^{-1}R_kA_k\mathbf{e}_k\|_{A_{k-1}} \\ &= \bar{\eta}\|\mathbf{d}_{k-1}\|_{A_{k-1}} \end{aligned}$$

Now we may write for the error after a modified coarse grid correction step:

$$\begin{aligned} \bar{T}_k\mathbf{e}_k &= \mathbf{e}_k - P_k\bar{\mathbf{d}}_{k-1} \\ &= \mathbf{e}_k - P_k\mathbf{d}_{k-1} + P_k(\mathbf{d}_{k-1} - \bar{\mathbf{d}}_{k-1}) \\ &= T_k\mathbf{e}_k + P_k(\mathbf{d}_{k-1} - \bar{\mathbf{d}}_{k-1}). \end{aligned}$$

As $\|P_k \cdot\|_{A_k} = \|\cdot\|_{A_{k-1}}$ we can estimate $\|P_k(\mathbf{d}_{k-1} - \bar{\mathbf{d}}_{k-1})\|_{A_k} \leq \bar{\eta}\|P_k\mathbf{d}_{k-1}\|_{A_k}$. Using the A_k-orthogonality of $\mathrm{ran}(T_k)$ and $\mathrm{ran}(P_k)$ we thus get:

$$\begin{aligned} \|\bar{T}_k\mathbf{e}_k\|_{A_k}^2 &= \|T_k\mathbf{e}^k\|_{A_k}^2 + \|P_k(\mathbf{d}_{k-1} - \bar{\mathbf{d}}_{k-1})\|_{A_k}^2 \\ &\leq \|T_k\mathbf{e}_k\|_{A_k}^2 + \bar{\eta}^2\|P_k\mathbf{d}_{k-1}\|_{A_k}^2. \end{aligned}$$

So using $P_k\mathbf{d}_{k-1} = (I - T_k)\mathbf{e}_k$ together with the A_k-orthogonality leads to

$$\|\bar{T}_k\mathbf{e}_k\|_{A_k}^2 \leq \|T_k\mathbf{e}_k\|_{A_k}^2 + \bar{\eta}^2(\|\mathbf{e}_k\|_{A_k}^2 - \|T_k\mathbf{e}_k\|_{A^k}^2).$$

Now we observe that

$$\begin{aligned} T_k\bar{T}_k &= (I - P_kA_{k-1}^{-1}R_kA_k)(I - P_k\bar{N}_{k-1}R_kA_k) \\ &= I - P_kN_{k-1}R_kA_k - P_kA_{k-1}^{-1}R_kA_k \\ &\quad + P_kA_{k-1}^{-1}R_kA_kP_kN_{k-1}R_kA_k \\ &= I - P_kA_{k-1}^{-1}R_kA_k \\ &= T_k \end{aligned}$$

and that $\|T_k\|_{A_k} = 1$. Similar to the proof of Lemma 3.3 we now write

$$\begin{aligned}
\|S_k\bar{T}_k\mathbf{e}_k\|_{A_k}^2 &\leq \|\bar{T}_k\mathbf{e}_k\|_{A_k}^2 - \alpha\|\bar{T}_k\mathbf{e}_k\|_*^2 \\
&\leq \|\bar{T}_k\mathbf{e}_k\|_{A_k}^2 - \alpha/\beta\|T_k\bar{T}_k\mathbf{e}_k\|_{A_k}^2 \\
&\leq \|T_k\mathbf{e}_k\|_{A_k}^2 + \bar{\eta}^2(\|\mathbf{e}_k\|_{A_k}^2 - \|T_k\mathbf{e}_k\|_{A^k}^2) - \alpha/\beta\|T_k\mathbf{e}_k\|_{A_k}^2 \\
&= (1 - \bar{\eta}^2 - \alpha/\beta)\|T_k\mathbf{e}_k\|_{A_k}^2 + \bar{\eta}^2\|\mathbf{e}_k\|_{A_k}^2 \\
&\leq \max\{(1 - \alpha/\beta), \bar{\eta}^2\}\|\mathbf{e}_k\|_{A_k}^2
\end{aligned}$$

□

Recursive application of this theorem yields convergence of multigrid methods using the Galerkin operator on the coarser levels. In that case η is the convergence rate of the method on the coarse level, thus the overall convergence rate is bounded by $\sqrt{1-\delta}$, as on the coarsest level the convergence rate is 0.

3.3.2 Replacement of the Galerkin operator

Besides its nice properties, the Galerkin operator has one main downside. As it is essentially formed by prolongating the residual to the fine level, applying the fine level operator there and restricting the result back to the coarse level, its application can be very expensive per unknown. As an example consider the following: Assume that the model problem is discretized using the 5-point discretization from (2.15), yielding the stencil (2.16), i.e.

$$\frac{1}{h^2}\begin{bmatrix} & 1 & \\ 1 & -4 & 1 \\ & 1 & \end{bmatrix}.$$

Now we construct a twogrid method utilizing the full-weighting operator given in Definition 3.10 for restriction and using the bilinear interpolation from Definition 3.12 as prolongation. Instead of rediscretizing the problem using the new grid spacing $2h$, we now use the Galerkin operator, yielding the following stencil representation on the coarse level

$$\frac{1}{h^2}\begin{bmatrix} \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \\ \frac{1}{8} & -\frac{3}{4} & \frac{1}{8} \\ \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \end{bmatrix}.$$

So the Galerkin operator on the coarse level has nine entries, compared to five entries on the fine level or using a coarse rediscretization. Numerical experiments show that the convergence of the method using the Galerkin operator is sightly better than the use of the rediscretization, but not enough to justify the additional cost. We like to emphasize

that this example is a best case scenario, as the drawback of the Galerkin operator will be even more pronounced in higher dimensions or for stencils involving more neighbors than only the next ones. For unstructured grids the problem can get even worse, as after a few levels we might end up with an operator that is not sparse anymore. For our purpose we are interested in reducing the computational time for structured matrices, only. For that purpose in the following we will present sufficient conditions for replacements of the Galerkin operator on the coarse grid, presumably resembling the sparsity pattern of the original matrix and the describing stencils, respectively.

We can subsume that we are interested in not using the Galerkin operator $A_{k-1} = R_k A_k R_k^H$ on the coarse level but rather an approximation $\hat{A}_{k-1}$. The convergence of the two grid method stated by the following lemma is an immediate consequence of Theorem 3.8 above.

Lemma 3.6 *Let A_k, R_k and T_k be defined as in Theorem 3.8 fulfilling the smoothing property and the approximation property, cf. Definition 3.23 and Definition 3.24, and let $\hat{T}$ be defined as $\bar{T}$ in Theorem 3.8 with $\tilde{N}_{k-1} = \hat{A}_{k-1}^{-1}$. Assume that*

$$\eta := \|I - \hat{A}_{k-1}^{-1} A_{k-1}\|_{A_{k-1}} < 1.$$

Then the (post-smoothing) two grid method using the approximation $\hat{A}_{k-1}$ of the Galerkin operator converges with a convergence bounded from above by $\max\{\eta, \sqrt{1-\delta}\}$.

As a consequence, in order to optimize the twogrid method we have to minimize

$$\eta = \|I - \hat{A}_{k-1}^{-1} A_{k-1}\|_{A_{k-1}} = \|A_{k-1}^{\frac{1}{2}}(I - \hat{A}_{k-1}^{-1} A_{k-1})\|_2.$$

under appropriate restriction given, for example, by a sparsity pattern imposed on $\hat{A}_{k-1}$. For application of the method we are interested in multigrid convergence rather than in twogrid convergence. Thus we need to analyze the convergence if the altered system is not solved directly but rather by a multigrid method itself, i.e we solve

$$\hat{A}_{k-1}\mathbf{d}_{k-1} = R_k A + k\mathbf{e}_k \tag{3.16}$$

using the multigrid method, which is the iterative method $\tilde{\phi}$ given by

$$\tilde{\phi}_{k-1}(\mathbf{x}_{k-1}, \mathbf{b}_{k-1}) = \tilde{M}_{k-1}\mathbf{x}_{k-1} + \tilde{N}_{k-1}\mathbf{b}_{k-1}$$

with initial zero approximation, i.e. we use $\tilde{N}_{k-1}$ as an approximate inverse of $\hat{A}_{k-1}$, which itself is an approximation of A_{k-1}^{-1}. Assume that

$$\hat{\eta} := \|I - \hat{A}_{k-1}^{-1} A_{k-1}\|_{A_{k-1}} < 1$$

and that the iterative method $\tilde{\phi}_{k-1}$ used to solve the modified defect equation converges with a convergence rate of at most $\tilde{\eta}$ in the $\hat{A}_{k-1}$-norm. More precisely, assume that

$$\tilde{\eta} := \mu\|I - \tilde{N}_{k-1}\hat{A}_{k-1}\|_{\hat{A}_{k-1}} < 1,$$

where $\mu > 0$ is the constant of the upper bound of the A_{k-1}-norm in terms of the $\hat{A}_{k-1}$-norm, i.e.

$$\|B\|_{A_{k-1}} \leq \mu \|B\|_{\hat{A}_{k-1}},$$

which exists due to the equivalence of norms. Since we want to apply Theorem 3.8 we only analyze $\|I - \tilde{N}_{k-1}A_{k-1}\|_{A_{k-1}}$. We have

$$\begin{aligned}
\|I - \tilde{N}_{k-1}A_{k-1}\|_{A_{k-1}} &= \|I - \tilde{N}_{k-1}\hat{A}_{k-1}\hat{A}_{k-1}^{-1}A_{k-1}\|_{A_{k-1}} \\
&= \|(I - \tilde{N}_{k-1}\hat{A}_{k-1})\hat{A}_{k-1}^{-1}A_{k-1} + (I - \hat{A}_{k-1}^{-1}A_{k-1})\|_{A_{k-1}} \\
&\leq \|I - \tilde{N}_{k-1}\hat{A}_{k-1}\|_{A_{k-1}}\|\hat{A}_{k-1}^{-1}A_{k-1}\|_{A_{k-1}} + \|I - \hat{A}_{k-1}^{-1}A_{k-1})\|_{A_{k-1}} \\
&\leq \mu\|I - \tilde{N}_{k-1}\hat{A}_{k-1}\|_{\hat{A}_{k-1}}\|\hat{A}_{k-1}^{-1}A_{k-1}\|_{A_{k-1}} + \|I - \hat{A}_{k-1}^{-1}A_{k-1})\|_{A_{k-1}} \\
&\leq \tilde{\eta}\|\hat{A}_{k-1}^{-1}A_{k-1}\|_{A_{k-1}} + \hat{\eta}
\end{aligned}$$

This is smaller than 1 if

$$\|\hat{A}_{k-1}^{-1}A_{k-1}\|_{A_{k-1}} \leq \frac{1-\hat{\eta}}{\tilde{\eta}},$$

which can always be fulfilled if $\hat{A}_{k-1}$ is sufficiently close to A_{k-1}, because then $\hat{\eta} \to 0$ as $\|\hat{A}_{k-1}^{-1}A_{k-1}\|_{A_{k-1}} \to 1$. For uniform multigrid convergence we need more, namely

$$\|I - \tilde{N}_{k-1}A_{k-1}\|_{A_{k-1}} \leq \max\{\tilde{\eta}, \sqrt{1-\delta}\}. \tag{3.17}$$

So we would have to impose

$$\begin{aligned}
& \tilde{\eta}\|\hat{A}_{k-1}^{-1}A_{k-1}\|_{A_{k-1}} + \hat{\eta} \leq \tilde{\eta} \\
\Leftrightarrow \quad & \hat{\eta} \leq (1 - \|\hat{A}_{k-1}^{-1}A_{k-1}\|_{A_{k-1}})\tilde{\eta}.
\end{aligned}$$

Now two cases are possible.

1. $\|\hat{A}_{k-1}^{-1}A_{k-1}\|_{A_{k-1}} < 1$. That implies that $1 - \|\hat{A}_{k-1}^{-1}A_{k-1}\|_{A_{k-1}} = \hat{\eta}$, thus we would require $\hat{\eta} \leq \hat{\eta}\tilde{\eta}$, which is true only for $\tilde{\eta} \geq 1$. So we would have no convergence.
2. $\|\hat{A}_{k-1}^{-1}A_{k-1}\|_{A_{k-1}} > 1$. This implies $0 < \hat{\eta} \leq \alpha\tilde{\eta}$, where $\alpha = 1 - \|\hat{A}_{k-1}^{-1}A_{k-1}\|_{A_{k-1}} < 0$, so $\tilde{\eta} < 0$ as well, which is not admissible.

So we conclude that this approach is not feasible to show the desired result: Rewriting the modified method in a way that allows us to split it into one part describing the approximation of the Galerkin operator and another part describing the approximate solution of the modified coarse grid correction using the triangle inequality prohibits to prove uniform convergence. So we have to alter Ruge's and Stüben's theorem in order to allow us to prove uniform convergence in the case that an alternative coarse grid operator is used and the defect equation is solved approximately, only.

For that purpose we show two auxiliary results that will allow us to formulate a convergence theorem that is closely related to Ruge's and Stüben's Theorem 3.1 in [69].

Lemma 3.7 *Let $\hat{T}_k = I - P_k \hat{A}_{k-1}^{-1} R_k A_k$, with $\hat{A}_{k-1} \in \mathbb{C}^{n_{k-1} \times n_{k-1}}$ and $A_k \in \mathbb{C}^{n_k \times n_k}$ symmetric and positive definite, $P_k = R_k^H \in \mathbb{C}^{n_k \times n_{k-1}}$ being a full rank prolongation and $R_k \in \mathbb{C}^{n_{k-1} \times n_k}$ a full rank restriction. Assume that*

$$0 \leq_{A_k} \hat{T}_k \leq_{A_k} I$$

Then for all $\mathbf{e} \in \mathbb{C}^{n_k}$ we have

$$\|P_k \hat{\mathbf{d}}_{k-1}\|_{A_k}^2 \leq \|\mathbf{e}_k\|_{A_k}^2 - \|\hat{T}_k \mathbf{e}_k\|_{A_k}^2,$$

where $\hat{\mathbf{d}}_{k-1}$ is the solution of the linear system $\hat{A}_{k-1}^{-1} \hat{\mathbf{d}}_{k-1} = R_k A_k \mathbf{e}_k$.

Proof. As $\hat{T}_k \leq_{A_k} I$ and as

$$A_k \hat{T}_k = A_k (I - R_k^H A_{k-1}^{-1} R_k A_k) = (I - A_k R_k^H A_{k-1}^{-1} R_k) A_k = \hat{T}_k^H A_k,$$

we have

$$\hat{T}_k^2 - \hat{T}_k \leq 0 \Leftrightarrow A_k \hat{T}_k^2 - A_k \hat{T}_k \leq 0 \Leftrightarrow \hat{T}_k^H A_k \hat{T}_k - A_k \hat{T}_k \leq 0.$$

Now we can write

$$\begin{aligned}
\|R_k^H \hat{\mathbf{d}}_{k-1}\|_{A_k}^2 &= \|\mathbf{e}_k - \hat{T}_k \mathbf{e}_k\|_{A_k}^2 \\
&= \langle A_k(\mathbf{e}_k - \hat{T}_k \mathbf{e}_k), (\mathbf{e}_k - \hat{T}_k \mathbf{e}_k) \rangle \\
&= \langle A_k \mathbf{e}_k, \mathbf{e}_k \rangle - \langle A_k \mathbf{e}_k, \hat{T}_k \mathbf{e}_k \rangle - \langle A_k \hat{T}_k \mathbf{e}_k, \mathbf{e}_k \rangle + \langle A_k \hat{T}_k \mathbf{e}_k, \hat{T}_k \mathbf{e}_k \rangle \\
&= \langle A_k \mathbf{e}_k, \mathbf{e}_k \rangle - \langle A_k \hat{T}_k \mathbf{e}_k, \hat{T}_k \mathbf{e} \rangle + 2 \langle A_k \hat{T}_k \mathbf{e}_k, \hat{T}_k \mathbf{e}_k \rangle \\
&\quad - \langle A_k \mathbf{e}_k, \hat{T} \mathbf{e}_k \rangle - \langle A_k \hat{T}_k \mathbf{e}_k, \mathbf{e}_k \rangle \\
&= \langle A_k \mathbf{e}_k, \mathbf{e}_k \rangle - \langle A_k \hat{T}_k \mathbf{e}, \hat{T}_k \mathbf{e}_k \rangle + 2 \langle A \hat{T}_k \mathbf{e}_k, \hat{T}_k \mathbf{e}_k \rangle \\
&\quad - \langle \hat{T}_k^H A_k \mathbf{e}_k, \mathbf{e}_k \rangle - \langle A_k \hat{T}_k \mathbf{e}_k, \mathbf{e}_k \rangle \\
&= \|\mathbf{e}_k\|_{A_k}^2 - \|\hat{T}_k \mathbf{e}_k\|_{A_k}^2 + 2(\langle \hat{T}_k^H A_k \hat{T}_k \mathbf{e}_k, \mathbf{e}_k \rangle - \langle A_k \hat{T}_k \mathbf{e}_k, \mathbf{e}_k \rangle) \\
&= \|\mathbf{e}_k\|_{A_k}^2 - \|\hat{T}_k \mathbf{e}_k\|_{A_k}^2 + 2 \langle \underbrace{(\hat{T}_k^H A_k \hat{T}_k - A \hat{T}_k)}_{\leq 0} \mathbf{e}_k, \mathbf{e}_k \rangle \\
&\leq \|\mathbf{e}_k\|_{A_k}^2 - \|\hat{T}_k \mathbf{e}_k\|_{A_k}^2.
\end{aligned}$$

□

As before, we assume that we do not solve the coarse grid equation

$$\hat{A}_{k-1} \mathbf{d}_{k-1} = R_k A_k \mathbf{e}_k$$

directly but by an iterative method with iteration matrix $I - \tilde{N}_{k-1} \hat{A}_{k-1}$, yielding another approximate coarse grid correction $\tilde{T}_k$ given by

$$\tilde{T}_k = I - R_k^H \tilde{N}_{k-1} R_k A_k.$$

We assume that the iterative method converges with a convergence rate of at most $\tilde{\eta} < 1$ measured in the $\hat{A}_{k-1}$-norm, i.e. $\|I_{k-1} - \tilde{N}_{k-1}\hat{A}_{k-1}\|_{\hat{A}_{k-1}} \leq \tilde{\eta}$. We define

$$\tilde{\mathbf{d}}_k = \tilde{N}_{k-1} R_k A_k \mathbf{e}_k.$$

The second auxiliary result seems to be a little bit unhandy. We need a feature of the kernels of matrix products in order to show that we can estimate the square of the norm of the modified coarse grid correction times some error plus the prolongation of the difference of the defects using the modified defect equation and its approximation by the sum of the norm of both plus a bit more of the coarse grid correction times the error. Nevertheless we will see later on, that we are able to fulfill this prerequisite at least in the case of circulant matrices.

Lemma 3.8 *Let $\hat{T}_k = I - P_k \hat{A}_{k-1}^{-1} R_k A_k$, $\tilde{T}_k = I - P_k \tilde{N}_k R_k A_k$ with $\hat{A}_{k-1}^{-1} \in \mathbb{C}^{n_{k-1} \times n_{k-1}}$, $\tilde{N}_{k-1} \in \mathbb{C}^{n_{k-1} \times n_{k-1}}$ and $A_k^{n_k \times n_k}$ symmetric and positive definite, $P_k \in \mathbb{C}^{n_{k-1} \times n_k}$ being a full rank prolongation and $R_k \in \mathbb{C}^{n_k \times n_{k-1}}$ a full rank restriction. Assume that*

$$\ker(\hat{T}_k^H A_k \hat{T}_k) \subset \ker((\tilde{T}_k - \hat{T}_k)^H A_k \hat{T}_k + \hat{T}_k^H A_k (\tilde{T}_k - \hat{T}_k)).$$

Then

$$\lambda_k := \min_{\mathbf{e}_k \in \mathbb{C}^{n_k}} \frac{\langle ((\tilde{T}_k - \hat{T}_k)^H A_k \hat{T}_k + \hat{T}_k^H A_k (\tilde{T}_k - \hat{T}_k))\mathbf{e}_k, \mathbf{e}_k \rangle}{\langle \hat{T}_k^H A_k \hat{T}_k \mathbf{e}_k, \mathbf{e}_k \rangle}$$

exists, and for all $\mathbf{e}_k \in \mathbb{C}^{n_k}$ the following holds true:

$$\|\hat{T}_k \mathbf{e}_k + P_k(\hat{\mathbf{d}}_{k-1} - \tilde{\mathbf{d}}_{k-1})\|_{A_k}^2 \leq (1 + \lambda_k)\|\hat{T}_k \mathbf{e}_k\|_{A_k}^2 + \|P_k(\hat{\mathbf{d}}_{k-1} - \tilde{\mathbf{d}}_{k-1})\|_{A_k}^2.$$

Proof. Under the lemma's assumption both $(\tilde{T}_k - \hat{T}_k)^H A_k \hat{T}_k + \hat{T}_k^H A_k (\tilde{T}_k - \hat{T})$ and $\hat{T}_k^H A_k \hat{T}_k$ are symmetric and positive definite linear mappings on the quotient space $\mathbb{C}^{n_k} \setminus \ker((\tilde{T}_k - \hat{T}_k)^H A_k \hat{T}_k + \hat{T}_k^H A_k (\tilde{T}_k - \hat{T}_k))$, so they induce norms on that space that are given by

$$\begin{aligned} \|\cdot\|_{(\tilde{T}_k - \hat{T}_k)^H A_k \hat{T}_k + \hat{T}_k^H A_k (\tilde{T}_k - \hat{T}_k)} &= \langle ((\tilde{T}_k - \hat{T}_k)^H A_k \hat{T}_k + \hat{T}_k^H A_k (\tilde{T}_k - \hat{T}_k))\cdot, \cdot \rangle^{\frac{1}{2}}, \\ \|\cdot\|_{\hat{T}_k^H A_k} &= \langle \hat{T}_k^H A_k \cdot, \cdot \rangle^{\frac{1}{2}}. \end{aligned}$$

Due to the equivalence of norms we can estimate

$$\langle ((\tilde{T}_k - \hat{T}_k)^H A_k \hat{T}_k + \hat{T}_k^H A_k (\tilde{T}_k - \hat{T}_k))\cdot, \cdot \rangle \leq \lambda_k \langle \hat{T}_k^H A_k \cdot, \cdot \rangle,$$

where we chose λ_k to be the minimum λ_k which fulfills this estimate. Now we have:

$$P_k(\hat{\mathbf{d}}_{k-1} - \tilde{\mathbf{d}}_{k-1}) = (P_k \hat{A}_{k-1}^{-1} R_k A_k - P_k \tilde{N}_{k-1} R_k A_k)\mathbf{e}_k = (\tilde{T}_k - \hat{T}_k)\mathbf{e}_k.$$

So we can write:

$$\begin{aligned}
&\|\hat{T}_k\mathbf{e}_k + P_k(\hat{\mathbf{d}}_{k-1} - \tilde{\mathbf{d}}_{k-1})\|_{A_k}^2 \\
&\qquad = \|(\hat{T}_k + (\tilde{T}_k - \hat{T}_k))\mathbf{e}_k\|_{A_k}^2 \\
&\qquad = \langle A_k(\hat{T}_k + (\tilde{T}_k - \hat{T}_k))\mathbf{e}_k, (\hat{T}_k + (\tilde{T}_k - \hat{T}_k))\mathbf{e}_k\rangle \\
&\qquad = \langle A_k\hat{T}_k\mathbf{e}_k, \hat{T}_k\mathbf{e}_k\rangle + \langle A_k(\tilde{T}_k - \hat{T}_k)\mathbf{e}_k, (\tilde{T}_k - \hat{T}_k)\mathbf{e}_k\rangle + \\
&\qquad\quad \langle A_k\hat{T}_k\mathbf{e}, (\tilde{T}_k - \hat{T}_k)\mathbf{e}_k\rangle + \langle A_k(\tilde{T}_k - \hat{T}_k)\mathbf{e}_k, \hat{T}_k\mathbf{e}_k\rangle \\
&\qquad = \|\hat{T}_k\mathbf{e}_k\|_{A_k}^2 + \|(\tilde{T}_k - \hat{T}_k)\mathbf{e}_k\|_{A_k}^2 + \\
&\qquad\quad \langle(\tilde{T}_k - \hat{T}_k)^H A_k\hat{T}_k\mathbf{e}_k, \mathbf{e}_k\rangle + \langle\hat{T}_k^H A_k(\tilde{T}_k - \hat{T}_k)\mathbf{e}_k, \mathbf{e}_k\rangle \\
&\qquad = \|\hat{T}_k\mathbf{e}_k\|_{A_k}^2 + \|P_k(\hat{\mathbf{d}}_{k-1} - \tilde{\mathbf{d}}_{k-1})\|_{A_k}^2 + \\
&\qquad\quad \langle((\tilde{T}_k - \hat{T}_k)^H A_k\hat{T}_k + \hat{T}_k^H A_k(\tilde{T}_k - \hat{T}_k))\mathbf{e}_k, \mathbf{e}_k\rangle \\
&\qquad \leq \|\hat{T}_k\mathbf{e}_k\|_{A_k}^2 + \|P_k(\hat{\mathbf{d}}_{k-1} - \tilde{\mathbf{d}}_{k-1})\|_{A_k}^2 + \lambda_k\langle\hat{T}_k^H A_k\hat{T}_k\mathbf{e}_k, \mathbf{e}_k\rangle \\
&\qquad = \|\hat{T}_k\mathbf{e}_k\|_{A_k}^2 + \|P_k(\hat{\mathbf{d}}_{k-1} - \tilde{\mathbf{d}}_{k-1})\|_{A_k}^2 + \lambda_k\|\hat{T}_k\mathbf{e}_k\|_{A_k}^2 \\
&\qquad = (1+\lambda_k)\|\hat{T}_k\mathbf{e}_k\|_{A_k}^2 + \|P_k(\hat{\mathbf{d}}_{k-1} - \tilde{\mathbf{d}}_{k-1})\|_{A_k}^2.
\end{aligned}$$

□

Now we can show the convergence of the modified multigrid method not using a Galerkin coarse grid operator but rather an approximation to it and solving the coarse grid defect equation using that approximation with the help of an iterative method.

Theorem 3.9 *Let $\hat{T}_k = I - P_k\hat{A}_{k-1}^{-1}R_kA_k$, $\tilde{T}_k = I - P_k\tilde{N}_kR_kA_k$, with $A_k \in \mathbb{C}^{n_k\times n_k}$ and $\hat{A}_{k-1} \in \mathbb{C}^{n_{k-1}\times n_{k-1}}$ both symmetric and positive definite, $P_k = R_k^H \in \mathbb{C}^{n_k\times n_{k-1}}$ being a full rank prolongation and $R_k \in \mathbb{C}^{n_{k-1}\times n_k}$ a full rank restriction. Let $\tilde{N}_{k-1} \in \mathbb{C}^{n_{k-1}\times n_{k-1}}$ be a symmetric and positive definite matrix defined by a linear iterative method given by*

$$\tilde{\phi}_{k-1}(\mathbf{x}_{k-1}, \mathbf{b}_{k-1}) = \tilde{M}_{k-1}\mathbf{x}_{k-1} + \tilde{N}_{k-1}\mathbf{b}_{k-1}$$

converging with a convergence rate of at most $\tilde{\eta}_{k-1}$ given by

$$\tilde{\eta}_{k-1} := \|I - \tilde{N}_{k-1}\hat{A}_{k-1}\|_{\hat{A}_{k-1}} < 1.$$

Further let the linear iterative method $\phi_S^{(k)}$ with iteration matrix S_k used as smoother fulfill the smoothing property (3.12) *and let $\hat{T}_k$ fulfill the approximation property* (3.13)*, i.e.*

$$\|\hat{T}_k\mathbf{e}_k\|_{A_k}^2 \leq \hat{\beta}_k\|\mathbf{e}_k\|_*^2$$

Let

$$\begin{aligned}
0 &\leq_{A_k} \hat{T}_k \leq_{A_k} I, \\
\hat{A}_{k-1} &\geq R_kA_kP_k, \\
\ker(\hat{T}_k^H A_k\hat{T}_k) &\subset \ker((\tilde{T}_k - \hat{T}_k)^H A_k\hat{T}_k + \hat{T}_k^H A_k(\tilde{T}_k - \hat{T}_k))
\end{aligned}$$

and choose λ_k such that

$$\lambda_k := \min_{\mathbf{e}_k \in \mathbb{C}^{n_k}} \frac{\langle ((\tilde{T}_k - \hat{T}_k)^H A_k \hat{T}_k + \hat{T}_k^H A_k (\tilde{T}_k - \hat{T}_k))\mathbf{e}_k, \mathbf{e}_k \rangle}{\langle \hat{T}_k^H A_k \hat{T}_k \mathbf{e}_k, \mathbf{e}_k \rangle}$$

and μ_k such that

$$\mu_k = \min_{\mathbf{e}_k \in \mathbb{C}^{n_k}} \frac{\|\tilde{T}_k \mathbf{e}_k\|_*^2}{\|\hat{T}_k \mathbf{e}_k\|_*^2}$$

Under the assumptions that

$$\sqrt{(1+\lambda_k) - \hat{\alpha}_k/\hat{\beta}_k} < 1,$$

where $\hat{\alpha}_k := \mu_k \alpha_k$, the (post-smoothing) two grid method using the modified coarse grid correction and solving the coarse grid defect correction using the iterative method converges with convergence factor of at most

$$\max\left\{ \tilde{\eta}_{k-1}, \sqrt{(1+\lambda_k) - \alpha_k/\hat{\beta}_k} \right\},$$

i.e.

$$\|S^{\nu_2} \tilde{T}_k \mathbf{e}_k\|_{A_k} \leq \max\left\{ \tilde{\eta}_{k-1}, \sqrt{(1+\lambda_k) - \alpha_k/\hat{\beta}_k} \right\} \|\mathbf{e}_k\|_{A_k} \text{ for all } \mathbf{e}_k \in \mathbb{C}^{n_k}.$$

Proof. Combining the smoothing property (3.12) with (3.13) yields

$$\|S_k^{\nu_2} \mathbf{e}_k\|_{A_k}^2 \leq \|\mathbf{e}_k\|_{A_k}^2 - \frac{\alpha_k}{\hat{\beta}_k} \|\hat{T}_k \mathbf{e}_k\|_A^2 \tag{3.18}$$

for all $\mathbf{e}_k \in \mathbb{C}^{n_k}$. For the error of the approximate defect we can write

$$\begin{aligned}
\|\hat{\mathbf{d}}_{k-1} - \tilde{\mathbf{d}}_{k-1}\|_{A_{k-1}} &= \|\hat{A}_{k-1}^{-1} R_k A_k \mathbf{e}_k - \tilde{N}_{k-1} R_k A_k \mathbf{e}_k\|_{A_{k-1}} \\
&= \|\hat{A}_{k-1}^{-1} R_k A_k \mathbf{e}_k - \tilde{N}_{k-1} \hat{A}_{k-1} \hat{A}_{k-1}^{-1} R_k A_k \mathbf{e}_k\|_{A_{k-1}} \\
&= \|(I - \tilde{N}_{k-1} \hat{A}_{k-1}) \hat{A}_{k-1}^{-1} R_k A_k \mathbf{e}_k\|_{A_{k-1}} \\
&\leq \|I - \tilde{N}_{k-1} \hat{A}_{k-1}\|_{A_{k-1}} \|\hat{A}_{k-1}^{-1} R_k A_k \mathbf{e}_k\|_{A_{k-1}} \\
&\leq \|I - \tilde{N}_{k-1} \hat{A}_{k-1}\|_{\hat{A}_{k-1}} \|\hat{A}_{k-1}^{-1} R_k A_k \mathbf{e}_k\|_{A_{k-1}} \\
&\leq \tilde{\eta}_{k-1} \|\hat{\mathbf{d}}_{k-1}\|_{A_{k-1}}.
\end{aligned}$$

Now we may write for the error after an approximate modified coarse grid correction step:

$$\begin{aligned}
\tilde{T}_k \mathbf{e}_k &= \mathbf{e}_k - R_k^H \tilde{\mathbf{d}}_{k-1} \\
&= \mathbf{e}_k - R_k^H \hat{\mathbf{d}}_{k-1} + R_k^H (\hat{\mathbf{d}}_{k-1} - \tilde{\mathbf{d}}_{k-1}) \\
&= \hat{T}_k \mathbf{e}_k + R_k^H (\hat{\mathbf{d}}_{k-1} - \tilde{\mathbf{d}}_{k-1}).
\end{aligned}$$

As $\|R_k^H \cdot\|_{A_k} = \|\cdot\|_{A_{k-1}}$ we can estimate $\|R_k^H(\hat{\mathbf{d}}_{k-1} - \tilde{\mathbf{d}}_{k-1})\|_{A_k} \leq \tilde{\eta}_{k-1}\|R_k^H\hat{\mathbf{d}}_{k-1}\|_{A_k}$ and combined with Lemma 3.8 we get

$$\begin{aligned}\|\tilde{T}_k\mathbf{e}_k\|_{A_k}^2 &= \|\hat{T}_k\mathbf{e}_k + R_k^H(\hat{\mathbf{d}}_{k-1} - \tilde{\mathbf{d}}_{k-1})\|_{A_k}^2 \\ &\leq (1+\lambda_k)\|\hat{T}_k\mathbf{e}_k\|_{A_k}^2 + \|R_k^H(\hat{\mathbf{d}}_{k-1} - \tilde{\mathbf{d}}_{k-1})\|_{A_k}^2 \\ &\leq (1+\lambda_k)\|\hat{T}_k\mathbf{e}_k\|_{A_k}^2 + \tilde{\eta}_{k-1}^2\|R_k^H\hat{\mathbf{d}}_{k-1}\|_{A_k}^2.\end{aligned}$$

So with Lemma 3.7 we have

$$\|\tilde{T}_k\mathbf{e}_k\|_{A_k}^2 \leq (1+\lambda_k)\|\hat{T}_k\mathbf{e}_k\|_{A_k}^2 + \tilde{\eta}_{k-1}^2(\|\mathbf{e}_k\|_{A_k}^2 - \|\hat{T}_k\mathbf{e}_k\|_{A_k}^2).$$

Overall, with (3.18) we get:

$$\begin{aligned}\|S_k^{\nu_2}\tilde{T}_k\mathbf{e}_k\|_{A_k}^2 &\leq \|\tilde{T}_k\mathbf{e}_k\|_{A_k}^2 - \alpha_k\|\tilde{T}_k\mathbf{e}_k\|_*^2 \\ &\leq \|\tilde{T}_k\mathbf{e}_k\|_{A_k}^2 - \alpha_k\mu_k\|\hat{T}_k\mathbf{e}_k\|_*^2 \\ &\leq \|\tilde{T}_k\mathbf{e}_k\|_{A_k}^2 - \hat{\alpha}_k\|\hat{T}_k\mathbf{e}_k\|_*^2 \\ &\leq \|\tilde{T}_k\mathbf{e}_k\|_{A_k}^2 - \hat{\alpha}_k/\hat{\beta}_k\|\hat{T}_k\mathbf{e}_k\|_{A_k}^2 \\ &\leq ((1+\lambda_k) - \hat{\alpha}_k/\hat{\beta}_k - \tilde{\eta}_{k-1}^2)\|\hat{T}_k\mathbf{e}_k\|_{A_k}^2 + \tilde{\eta}_{k-1}^2\|\mathbf{e}_k\|_{A_k}^2 \\ &\leq \max\{((1+\lambda_k) - \hat{\alpha}_k/\hat{\beta}_k), \tilde{\eta}_{k-1}^2\}\|\mathbf{e}_k\|_A^2.\end{aligned}$$

□

We like to emphasize, that both, λ_k and μ_k depend on $\tilde{T}_k$ and can be very large and small, respectively. So for a detailed analysis of a multigrid method both require further investigation.

By recursive application we immediately obtain the following result.

Theorem 3.10 *Let $\phi_{MGM}^{(k_{max})}$ be a multigrid method where T_k and A_{k-1}, $k = 1, \ldots, k_{max}$ fulfill the requirements of Theorem 3.9. Then the convergence rate of $\phi_{MGM}^{(k_{max})}$ is bounded from above by*

$$\max_{k=1,\ldots,k_{max}} \left\{ \max \left\{ \tilde{\eta}, \sqrt{(1+\lambda_k) - \hat{\alpha}_k/\hat{\beta}_k} \right\} \right\} < 1.$$

It remains to note that the degradation of the performance of the multigrid method using a replacement of the Galerkin operator depends on how much worse the approximation property (3.13) is fulfilled by $\hat{T}_k$ compared to T_k and on the size of λ_k, which should be very small and almost negligible.

We will close this section with a lemma providing an alternative requirement implying $0 \leq_{A_k} \hat{T}_k \leq_{A_k} I$.

Lemma 3.9 *Let $\hat{T}_k = I - P_k \hat{A}_{k-1}^{-1} R_k A_k$, $A_k \in \mathbb{C}^{n_k \times n_k}$ and $\hat{A}_{k-1} \in \mathbb{C}^{n_{k-1} \times n_{k-1}}$ both symmetric and positive definite, $P_k = R_k^H \in \mathbb{C}^{n_k \times n_{k-1}}$ being a full rank prolongation and $R_k \in \mathbb{C}^{n_{k-1} \times n_k}$ a full rank restriction. If*

$$\hat{A}_{k-1} \geq A_{k-1},$$

then we also have

$$0 \leq_{A_k} \hat{T}_k \leq_{A_k} I.$$

Proof. Let $T_{k,2} = I - A_k^{\frac{1}{2}} R_k^H A_{k-1}^{-1} R_k A_k^{\frac{1}{2}}$ and $\hat{T}_{k,2} = I - A_k^{\frac{1}{2}} R_k^H \hat{A}_{k-1}^{-1} R_k A_k^{\frac{1}{2}}$. Then we have

$$\begin{aligned} & 0 \leq_{A_k} \hat{T}_k \leq_{A_k} I \\ \Leftrightarrow\ & 0 \leq I - A_k^{\frac{1}{2}} R_k^H \hat{A}_{k-1}^{-1} R_k A_k^{\frac{1}{2}} \leq I \\ \Leftrightarrow\ & 0 \leq \hat{T}_{k,2} \leq I. \end{aligned}$$

Now we can write

$$\begin{aligned} \hat{T}_{k,2} &= \hat{T}_{k,2} T_{k,2} + \hat{T}_{k,2}(I - T_{k,2}) \\ &= T_{k,2} + \hat{T}_{k,2}(I - T_{k,2}) \\ &= T_{k,2} + (I - A_k^{\frac{1}{2}} R_k^H \hat{A}_{k-1}^{-1} R_k A_k^{\frac{1}{2}})(A_k^{\frac{1}{2}} R_k^H A_{k-1}^{-1} R_k A_k^{\frac{1}{2}}) \\ &= T_{k,2} + (A_k^{\frac{1}{2}} R_k^H A_{k-1}^{-1} R_k A_k^{\frac{1}{2}} - A_k^{\frac{1}{2}} R_k^H \hat{A}_{k-1}^{-1} R_k A_k R_k^H A_{k-1}^{-1} R_k) \\ &= T_{k,2} + A_k^{\frac{1}{2}} R_k^H (A_{k-1}^{-1} - \hat{A}_{k-1}^{-1}) R_k A_k^{\frac{1}{2}}. \end{aligned}$$

As $T_{k,2}$ is the orthogonal projector onto the complement of $A_k^{\frac{1}{2}} R_k^H$ and as the range of $A_k^{\frac{1}{2}} R_k^H (A_{k-1}^{-1} - \hat{A}_{k-1}^{-1}) R_k A_k^{\frac{1}{2}}$ is a subset of the range of $A_k^{\frac{1}{2}} R_k^H$ we obtain that all vectors belonging to the orthogonal complement of $A_k^{\frac{1}{2}} R_k^H$ are mapped to itself, so we only have to show that

$$0 \leq A_k^{\frac{1}{2}} R_k^H (A_{k-1}^{-1} - \hat{A}_{k-1}^{-1}) R_k A_k^{\frac{1}{2}} \leq I. \tag{3.19}$$

From $\hat{A}_{k-1} \geq A_{k-1}$ we immediately obtain the first part of the inequality. Furthermore we obtain that $A_k^{\frac{1}{2}} R_k^H (A_{k-1}^{-1} - \hat{A}_{k-1}^{-1}) R_k A_k^{\frac{1}{2}}$ is positive definite and that

$$A_k^{\frac{1}{2}} R_k^H (A_{k-1}^{-1} - \hat{A}_{k-1}^{-1}) R_k A_k^{\frac{1}{2}} \leq A_k^{\frac{1}{2}} R_k^H A_{k-1}^{-1} R_k A_k^{\frac{1}{2}}.$$

Choosing an arbitrary $\mathbf{x} \in \operatorname{ran}(A_k^{\frac{1}{2}} R_k^H)$ there exists a $\mathbf{y}$ such that $\mathbf{x} = A_k^{\frac{1}{2}} R_k^H \mathbf{y}$ and we get

$$A_k^{\frac{1}{2}} R_k^H A_{k-1}^{-1} R_k A_k^{\frac{1}{2}} \mathbf{x} = A_k^{\frac{1}{2}} R_k^H A_{k-1}^{-1} R_k A_k^{\frac{1}{2}} A_k^{\frac{1}{2}} R_k^H \mathbf{y} = A_k^{\frac{1}{2}} R_k^H \mathbf{y} = \mathbf{x},$$

yielding the second part of inequality 3.19. □

3.3.3 Application to circulant matrices

In the following we will discuss multigrid methods for circulant matrices. As circulant matrices form a matrix algebra, they are relatively easy to analyze. Nevertheless they are an important class of matrices, as they occur in various problems, i.e. when solving discretized partial differential equations with constant coefficients or integral equations on the torus. Further on they are prototypes for the analysis of Toeplitz matrices, as the spectrum of both is asymptotically equal and they serve well for the analysis of non-constant coefficient problems, as well. A review covering both circulant and Toeplitz matrices has been written by Gray [44].

The development of multigrid methods for circulant matrices is based on the theory for Toeplitz matrices. The idea is to apply the algebraic multigrid theory that was presented before to Toeplitz or circulant matrices and to construct prolongation and restriction such that the resulting matrices on the coarser levels still belong to the same class of matrices. This methodology goes back at least to Fiorentino and Serra who published first results for banded symmetric Toeplitz matrices which arise in the discretization of partial differential equation in [32] and in [34] and extended their theory to the indefinite case in [33]. They provided the basis of the theory to be presented later on, namely the choice of the restriction and prolongation operator and the application of the algebraic multigrid theory to structured problems we presented in the previous section. These works were continued by Sun, Chan and Chang in [79]. Chan, Chang and Sun published results on ill-conditioned Toeplitz systems in [17]. Their theory is similar to the theory presented in the works of Fiorentino and Serra, but they use a different interpolation operator. In [80] Sun, Jin and Chang extended the theory to cover ill-conditioned block Toeplitz systems as well. While the theory for Toeplitz matrices uses τ-matrices as a theoretical foundation, in [74, 73] Serra Capizzano and Tablino-Possio presented first results for the application of the theory to circulant matrices. Aricò, Donatelli and Serra-Capizzano provided a proof of the optimality of the V-cycle in the unilevel case in [3], further details and applications of these theoretical results and a general overview can be found in the PhD thesis of Aricò [1] and in the one of Donatelli [22]. In [2] they provided an extension to the multilevel case.

We now start with a brief introduction of circulant matrices and some of their properties.

3.3.4 Circulant matrices

Circulant matrices are a special class of structured matrices, that are given by the following definition.

Definition 3.27 *Let $f : [-\pi, \pi) \to \mathbb{C}$ be a univariate 2π-periodic function and let*

$$(F_n)_{j,k=0}^{n-1} \text{ with } (F_n)_{j,k} = \frac{1}{\sqrt{n}} e^{-2\pi i \frac{jk}{n}}$$

be the Fourier matrix of dimension $n \times n$. The matrix $A \in \mathbb{C}^{n\times n}$ given by

$$A = \mathcal{A}(f) = F_n \operatorname{diag}\left((f(2\pi j/n))_{j=0}^{n-1}\right) F_n^H$$

is called a circulant matrix*, the function f is called the* generating symbol of C.

Circulant matrices are diagonalized by the orthogonal Fourier matrix, the rows of the Fourier matrix are the eigenvectors of circulant matrices. Due to the simultaneous diagonalizability they form a commutative matrix algebra. The multiplication of vectors with circulant matrices and the solution of linear systems with circulant coefficient matrix can be carried out in $\mathcal{O}(n \log n)$ operations using the FFT. The concept of circulant matrices can also be transferred to multiple levels, i.e. multivariate generating symbols and Kronecker products of Fourier matrices.

Definition 3.28 *Let $f : [-\pi, \pi)^d \to \mathbb{C}$ be a d-variate periodic function defined on $[-\pi, \pi)^d$. Let*

$$F_{\mathbf{n}} = \frac{1}{\sqrt{n_1 n_2 \cdots n_d}} \left(e^{-i\mathbf{k}\cdot\mathbf{w}_{\mathbf{j}}^{[\mathbf{n}]}}\right)_{\mathbf{j},\mathbf{k}\in\mathcal{I}_{\mathbf{n}}}.$$

be the d-level Fourier matrix*, where the vector $\mathbf{w}^{[\mathbf{n}]}$ is a sampling of the domain of f, i.e.*

$$\mathbf{w}_{\mathbf{j}}^{[\mathbf{n}]} = \left(\frac{2\pi j_1}{n_1}, \ldots, \frac{2\pi j_d}{n_d}\right),$$

and $\mathcal{I}_{\mathbf{n}} = \{0, \ldots, n_1 - 1\} \times \cdots \times \{0, \ldots, n_d - 1\}$ is the set of multi-indices. Then

$$A = \mathcal{A}(f) := F_{\mathbf{n}} \mathit{Diag}(f(\mathbf{w}^{[\mathbf{n}]}))F_{\mathbf{n}}^H$$

is the d-level circulant matrix with generating symbol f.

All the properties of the unilevel circulant matrix can be transferred to the multilevel case using tensorial arguments. In the following, we will discuss the unilevel case, only where the transfer to the multilevel case gets more involved, we will explicitly switch to that case.

3.3.5 Multigrid methods for circulant matrices

Although there already exist fast $\mathcal{O}(n \log n)$ algorithms for circulant matrices, we are interested in multigrid methods for those matrices, as the multiplication with banded circulant matrices is even cheaper, namely it can be done with $\mathcal{O}(n)$ operations. In the construction of multigrid methods for circulant matrices the zeros of the generating symbols play an important role. As the eigenvalues of the circulant matrices are given by a sampling of the generating symbol, these circulant matrices are at least asymptotically ill-conditioned and may get singular at some point. A singularity can be handled at least theoretically, c.f. [86], by a rank one correction, a technique Aricò and Donatelli [2] refer to as *Strang correction*.

Definition 3.29 (Strang correction) *Let $\mathcal{A}(f)$ be a circulant matrix with generating symbol $f \geq 0$ and let f have a single zero at $x_0 = 2\pi j_0/n$, $j_0 \in \mathbb{N}$. Then the modification of the system by using*

$$f_+(x) = \begin{cases} f(x) & \text{for } x \neq x_0 \\ \delta & \text{for } x = x_0 \end{cases},$$

$\delta > 0$, as generating symbol, resulting in the altered matrix

$$\mathcal{A}(f_+) = \mathcal{A}(f) + \delta(F_n)^H_{j=j_0,k=0,\ldots,n-1}(F_n)_{j=j_0,k=0,\ldots,n-1}$$

is called Strang correction.

This modification still solves the original system, at least if the right hand side does not have components that are collinear to the eigenvector belonging to the originally zero eigenvalue. It does keep the ill-conditioning of the system, so iterative methods like Jacobi or Richardson will fail. Like in the geometric case multigrid methods do not share this downside. The Strang correction approach might be chosen for more than one isolated zero. For methods dealing with generating symbols with zero curves, we refer to the PhD thesis of Fischer [35]. For the definition of multigrid methods for circulant matrices we restrict ourselves to the case $n = 2^{k_{\max}}$, $k_{\max} \in \mathbb{N}$. The extension to other factors than 2 is straightforward. So we define the number of unknowns n_k on level k as $n_k = 2^k$, in the multilevel case we do the same for each direction. For the definition of the restriction operator, we need the cutting matrix K_{n_k}, given by

$$K_{n_k} = \begin{pmatrix} 1 & 0 & & & & & \\ & & 1 & 0 & & & \\ & & & & \ddots & \ddots & \\ & & & & & 1 & 0 \end{pmatrix} \in \mathbb{C}^{n_k \times \frac{n_k}{2}},$$

the multilevel equivalent is given by $K_{\mathbf{n}_k} = K_{(\mathbf{n}_k)_1} \otimes \cdots \otimes K_{(\mathbf{n}_k)_d}$. The restriction operator itself is defined as $K_{n_k}\mathcal{A}(p_k)$, where p_k is a trigonometric polynomial. Assuming that the generating symbol f_k of A_k has a unique zero x_0 the symbol p_k is chosen such that the limit

$$\limsup_{x \to x_0} \left| \frac{p_k(x+\pi)}{f_k(x)} \right|$$

exists. Further for the prolongation to have full rank we demand for all $x \in [-\pi, \pi)$ that $p(x) + p(x+\pi) > 0$. In the multilevel case, i.e. for a unique zero $\mathbf{x}_0$, the symbol p_j is chosen that the limit

$$\limsup_{\mathbf{x} \to \mathbf{x}_0} \left| \frac{p_j(\mathbf{y})}{f_j(\mathbf{x})} \right|$$

exists for all points $\mathbf{y} \in \{\mathbf{z} \mid z_j \in \{x_{0_j}, x_{0_j} + \pi\}\} \backslash \{\mathbf{x}_0\}$ and such that the sum of the value of p over all *mirror points*, i.e. the points $\mathbf{y} \in \{\mathbf{z} \mid z_j \in \{x_{0_j}, x_{0_j} + \pi\}\}$, is larger than zero.

Now for a zero x_0 in accordance to [74] we consider $\hat{x} = x_0 + \pi$ if $x_0 < \pi$ or $\hat{x} = x_0 - \pi$ otherwise, and we set the generating symbol of the restriction to

$$p(x) = (2 - 2\cos(x - \hat{x}))^{\lceil \beta/2 \rceil},$$

with

$$\beta \geq \min\left\{ i \middle| \lim_{x \to x_0} \frac{|x - x_0|^{2i}}{f(x)} < \infty \right\}.$$

Using the transpose of the restriction as prolongation these choices assure, that the Galerkin operator still has only one zero, see [2]. Serra-Capizzano and Tablino-Possio showed in [74] that using these choices the coarse grid correction operator fulfills the approximation property. They have also shown that the Richardson iteration fulfills the smoothing property for the circulant matrices under consideration. In contrast to their work, which is based on the use of the $A_k \operatorname{diag}(A_k)^{-1} A_k$-norm for both, the approximation and the smoothing property, like Ruge and Stüben did in their introduction [69], in [2] Aricò and Donatelli used the A^2 norm for the same purpose, which in our opinion makes the proof a little bit more elegant. Besides this difference, they have also shown the uniform convergence of the multigrid method by analyzing the series of generating symbols of the Galerkin operators, something that is missing in the previous works of Serra and his colleagues. For details of these proofs we refer to their paper [2]. We will use their approach to show that a modified coarse grid correction still possesses the approximation property.

3.3.6 Replacement of the Galerkin operator for circulant matrices

We want to replace the Galerkin operator by some operator that is similar to it. For our purpose we demand from this replacing operator $\hat{A}_{k-1}$ that it is spectrally larger than the Galerkin operator $R_k A_k P_k$, but we want it to be bound by an upper bounded Λ times the Galerkin operator, i.e. we want to have

$$R_k A_k R_k^H \leq \hat{A}_{k-1} \leq \Lambda R_k A_k R_k^H. \tag{3.20}$$

Further on we demand $\hat{A}_{k-1}$'s generating symbol and the generating symbol of $R_k A_k R_k^H$ to have only one zero, that is common. To simplify our theoretical considerations, we require the approximation to satisfy a little more, namely for some $\varepsilon > 0$ we want to have

$$(1 + \varepsilon) R_k A_k P_k \leq \hat{A}_{k-1} \leq \Lambda R_k A_k R_k^H. \tag{3.21}$$

We express both requirements in terms of the generating symbols, the proof is a direct consequence of the properties of the generating symbols.

Lemma 3.10 *Let f_{k-1} be the generating symbol of $A_{k-1} = R_k A_k R_k^H$ and let $\hat{f}_{k-1}$ be the generating symbol of $\hat{A}_{k-1}$ and assume that for some $\varepsilon > 0$ and some $\Lambda > 1$ we have*

$$(1 + \varepsilon) f_{k-1} \leq \hat{f}_{k-1} \leq \Lambda f_{k-1}.$$

Then we have (3.20) *and* (3.21).

Now we have to show four presumptions in order to be able to apply Theorem 3.9, namely

1. $\|\hat{T}_k \mathbf{e}_k\|_{A_k}^2 \leq \hat{\beta} \|\mathbf{e}_k\|_*^2$,
2. $0 \leq_{A_k} \hat{T}_k \leq_{A_k} I$,
3. $\ker(\hat{T}_k^H A_k \hat{T}_k) \subset \ker((\tilde{T}_k - \hat{T}_k)^H A_k \hat{T}_k + \hat{T}_k^H A_k (\tilde{T}_k - \hat{T}_k))$.

We will now show these prerequisites. The second is fulfilled by the requirements stated above and Lemma 3.9. Now we have to show the remaining two items. We start with the first one, the proof is similar to the proof of the approximation property of the coarse grid correction involving the Galerkin operator by Aricò and Donatelli in [2].

Theorem 3.11 *For a fixed level k let f_k be the generating symbol of the matrix A_k, f_{k-1} be the generating symbol of $A_{k-1} = R_k A_k R_k^H$ and let $\hat{f}_{k-1}$ be the generating symbol of the matrix $\hat{A}_{k-1}$. Assume that $f_{k-1} \leq \hat{f}_{k-1} \leq \Lambda f_{k-1}$ and that the generating symbol p_k defining the restriction fulfills the conditions*

$$\limsup_{\mathbf{x} \to \mathbf{x}_0} \left| \frac{p_k(\mathbf{y})}{f_k(\mathbf{x})} \right| < \infty \text{ for all } \mathbf{y} \in \Omega(\mathbf{x}_0) \backslash \{\mathbf{x}_0\} \text{ and} \tag{3.22}$$

$$\sum_{\mathbf{y} \in \Omega(\mathbf{x})} p_k^2(\mathbf{y}) > 0 \text{ for all } \mathbf{x} \in [-\pi, \pi)^d, \tag{3.23}$$

where

$$\Omega(\mathbf{x}) := \{\mathbf{z} \mid z_l \in \{x_l, x_l + \pi\}\}.$$

Then there exists a constant $\hat{\beta}$, depending only on p, f and Λ, such that

$$||\hat{T}_k \mathbf{e}_k||_{A_k}^2 \leq \hat{\beta} ||\mathbf{e}_j||_{A_k^2}^2. \tag{3.24}$$

Proof. Equation (3.24) can equivalently be formulated as

$$\hat{T}_k^H A_k \hat{T}_k \leq \hat{\beta} A_k^2.$$

Now,

$$\begin{aligned}
\hat{T}_k^H A_k \hat{T}_k &= (I - R_k^H \hat{A}_{k-1}^{-1} R_k A_k)^H A_k (I - R_k^H \hat{A}_{k-1}^{-1} R_k A_k) \\
&= (I - A_k R_k^H \hat{A}_{k-1}^{-1} R_k)(A_k - A_k R_k^H \hat{A}_{k-1}^{-1} R_k A_k) \\
&= A_k - A_k R_k^H \hat{A}_{k-1}^{-1} R_k A_k - A_k R_k^H \hat{A}_{k-1}^{-1} R_k A_k + A_k R_k^H \hat{A}_{k-1}^{-1} \underbrace{R_k A_k R_k^H}_{\leq \hat{A}_{k-1}} \hat{A}_{k-1}^{-1} R_k A_k \\
&\leq A_k - A_k R_k^H \hat{A}_{k-1}^{-1} R_k A_k - A_k R_k^H \hat{A}_{k-1}^{-1} R_k A_k + A_k R_k^H \hat{A}_{k-1}^{-1} \hat{A}_{k-1} \hat{A}_{k-1}^{-1} R_k A_k \\
&= A_k - A_k R_k^H \hat{A}_{k-1}^{-1} R_k A_k \\
&= A_k \hat{T}_k.
\end{aligned}$$

To prove (3.24) it is thus sufficient to show

$$A_j \hat{T}_k \leq \hat{\beta} A_k^2. \tag{3.25}$$

This will now be done in a manner similar to the convergence proof for multigrid for multilevel matrix algebras in [2]. Define $\hat{R}_k = R_k A_k^{1/2} = K_{\mathbf{n_j}} \mathcal{A}_k(\hat{p}_k)$ with $\hat{p}_k = p_k f_k^{1/2}$. Then (3.25) is implied by

$$I - \hat{R}_k^H \hat{A}_{k-1}^{-1} \hat{R}_k \leq \hat{\beta} A_k, \tag{3.26}$$

which is what we will show now. Let us for the moment assume that A_k is 1-level circulant, i.e. $d = 1$. Multiplying the Fourier matrix from the left with the cut matrix then yields the decomposition

$$K_{n_k} F_{n_k} = \frac{1}{\sqrt{2}} \left(F_{n_{k-1}} \left| F_{n_{k-1}} \right. \right),$$

as is shown in [74], e.g. So

$$\begin{aligned} \mathcal{A}_{n_k}(f_{k-1}) = R_k A_k R_k^H &= K_{n_k} \mathcal{A}_{n_k}(p_k) A_k \mathcal{A}_{n_k}(p_k)^H K_{n_k}^H \\ &= K_{n_k} F_{n_k} F_{n_k}^H \mathcal{A}_{n_k}(p_k) \mathcal{A}_{n_k}(f) \mathcal{A}_{n_k}(p_k)^H F_{n_k} F_{n_k}^H K_{n_k}^H \\ &= \frac{1}{2} \left(F_{n_{k-1}} \left| F_{n_{k-1}} \right. \right) F_{n_k}^H \mathcal{A}_{n_k}(p_k f_k p_k) F_{n_k} \left(F_{n_{k-1}}^H \left| F_{n_{k-1}}^H \right. \right)^H, \end{aligned}$$

which gives

$$F_{n_{k-1}}^H \mathcal{A}_{n_k}(f_{k-1}) F_{n_{k-1}} = \frac{1}{2} \left(I \mid I \right) F_{n_k}^H \mathcal{A}_{n_k}(p_k f_k p_k)^H F_{n_k} \left(I \mid I \right)^T. \tag{3.27}$$

This decomposition can be generalized to $d > 1$ using tensorial arguments.

According to [74] the matrix $F_{\mathbf{n}_k}^H \hat{T}_k F_{\mathbf{n}_k}$ can be symmetrically permuted to a block diagonal matrix with $2^d \times 2^d$-blocks. Using the "square bracket notation" $f[\mathbf{x}]$ to denote the vector of length 2^d with

$$f[\mathbf{x}] = \frac{1}{2^d} \cdot (f(\mathbf{y}_1), \ldots, f(\mathbf{y}_{2^d}))^T,$$

where the $\mathbf{y}_j$ are a systematic enumeration of all the 2^d elements of the set $\Omega(\mathbf{x})$, these blocks are given as

$$I - \frac{1}{\hat{f}_{k-1}(2\,\mathbf{w}_{\mathbf{k}}^{[\mathbf{n}]})} \hat{p}_k[\mathbf{w}_{\mathbf{k}}^{[\mathbf{n}]}] \left(\hat{p}_k[\mathbf{w}_{\mathbf{k}}^{[\mathbf{n}]}] \right)^H.$$

With the d-dimensional analogue to (3.27) we obtain

$$f_{k-1}(2\,\mathbf{w}_{\mathbf{k}}^{[\mathbf{n}]}) = ||(p_k f_k^{1/2})[\mathbf{w}_{\mathbf{k}}^{[\mathbf{n}]}]||_2^2 = ||\hat{p}_k[\mathbf{w}_{\mathbf{k}}^{[\mathbf{n}]}]||_2^2.$$

Using $\hat{f}_{k-1} \leq \Lambda f_{k-1}$ and the definition of the Galerkin coarse grid operator we obtain

$$\begin{aligned} & I - \frac{1}{\hat{f}_{k-1}(2\,\mathbf{w}_{\mathbf{k}}^{[\mathbf{n}]})} \hat{p}_k[\mathbf{w}_{\mathbf{k}}^{[\mathbf{n}]}] \left(\hat{p}_k[\mathbf{w}_{\mathbf{k}}^{[\mathbf{n}]}]\right)^H \\ & \leq I - \frac{1}{\Lambda f_{k-1}(2\,\mathbf{w}_{\mathbf{k}}^{[\mathbf{n}]})} \hat{p}_k[\mathbf{w}_{\mathbf{k}}^{[\mathbf{n}]}] \left(\hat{p}_k[\mathbf{w}_{\mathbf{k}}^{[\mathbf{n}]}]\right)^H \\ & = I - \frac{1}{\Lambda ||\hat{p}_k[\mathbf{w}_{\mathbf{k}}^{[\mathbf{n}]}]||_2^2} \hat{p}_k[\mathbf{w}_{\mathbf{k}}^{[\mathbf{n}]}] \left(\hat{p}_k[\mathbf{w}_{\mathbf{k}}^{[\mathbf{n}]}]\right)^H. \end{aligned}$$

Consequently, to show (3.26), it is sufficient to prove

$$I - \frac{1}{\Lambda ||\hat{p}_k[\mathbf{w}_{\mathbf{k}}^{[\mathbf{n}]}]||_2^2} \hat{p}_k[\mathbf{w}_{\mathbf{k}}^{[\mathbf{n}]}] \left(\hat{p}_k[\mathbf{w}_{\mathbf{k}}^{[\mathbf{n}]}]\right)^H < \hat{\beta}\, \mathrm{diag}(f_k[\mathbf{w}_{\mathbf{k}}^{\mathbf{n}}]).$$

Actually, we will show slightly more, namely that for all $\mathbf{x}$ we have

$$Z(\mathbf{x}) = (\mathrm{diag}(f_k[\mathbf{x}]))^{-1/2} \left(I - \frac{1}{\Lambda ||\hat{p}_k[\mathbf{x}]||_2^2} \hat{p}_k[\mathbf{x}]\, (\hat{p}_k[\mathbf{x}])^H \right) (\mathrm{diag}(f_k[\mathbf{x}]))^{-1/2} \leq \hat{\beta} I.$$

First we deal with an entry $Z(\mathbf{x})_{q,r}$, where $q \neq r$:

$$\begin{aligned} Z(\mathbf{x})_{q,r} &= -\frac{\hat{p}_k(\mathbf{y}_q)\hat{p}_k(\mathbf{y}_q)}{\sqrt{f_k(\mathbf{y}_q) f_k(\mathbf{y}_r)}} \cdot \frac{1}{\Lambda ||\hat{p}_k[\mathbf{x}]||_2^2} \\ &= -\frac{p_k(\mathbf{y}_q) p_k(\mathbf{y}_r)}{\Lambda \sum\limits_{\mathbf{y} \in \Omega(\mathbf{x})} p_k^2(\mathbf{y}) f_k(\mathbf{y})}. \end{aligned}$$

This is bounded due to the hypothesis on p_k from (3.22). For $Z(\mathbf{x})_{q,q}$ we can write

$$\begin{aligned} Z(\mathbf{x})_{q,q} &= \sum_{\mathbf{y} \in \Omega(\mathbf{x}) \setminus \{\mathbf{x}\}} \frac{\hat{p}_k(\mathbf{y}_q)^2}{f_k(\mathbf{y}_q)} \cdot \frac{1}{\Lambda ||\hat{p}_k[\mathbf{x}]||_2^2} \\ &= \frac{1}{\Lambda} \left(\frac{1}{f_k(\mathbf{y}_q)} - \frac{p_k^2(\mathbf{y}_q)}{\sum\limits_{\mathbf{y} \in \Omega(\mathbf{x})} p_k^2(\mathbf{y}) f_k(\mathbf{y})} \right). \end{aligned}$$

If $q > 1$, then $f_k(\mathbf{y}_q) \neq 0$ and by (3.22) again we have that $Z(\mathbf{x})_{q,q}$ is bounded. For $q = 1$ we have $\mathbf{y}_q = \mathbf{x}$, so we get

$$Z(\mathbf{x})_{1,1} = \frac{\sum\limits_{\mathbf{y} \in \Omega(\mathbf{x}) \setminus \{\mathbf{x}\}} p_k(\mathbf{y})^2 f_k(\mathbf{y})}{f_k(\mathbf{x})^2} \cdot \frac{1}{\Lambda \sum\limits_{\mathbf{y} \in \Omega(\mathbf{x})} p_k(\mathbf{y})^2 \frac{f_k(\mathbf{y})}{f_k(\mathbf{x})}},$$

which is also bounded, as the first part of the product is bounded due to the same argument as before and the second part is bounded since the sum in the denominator is bounded away from 0 due to (3.23). So we can choose $\hat{\beta}$ as

$$\hat{\beta} := \max_{q,r=1,\dots,d} \left\{ \max_{\mathbf{x}\in[-\pi,\pi)^d} (Z_{q,r}(\mathbf{x})) \right\} \ (< \infty).$$

□

Comparing the proof of this Theorem with the proof of the approximation property by Aricò and Donatelli yields that $\hat{\beta}$ differs from β by a factor of $1/\Lambda$. Now we proceed showing the last required property. Using the altered requirement (3.21) we can show the following.

Lemma 3.11 *Let $\hat{T}_k = I - R_k^H \hat{A}_{k-1}^{-1} R_k A_k$, with $R_k^H \in \mathbb{C}^{n_k \times n_{k-1}}$ being a full rank prolongation operator. Let both $A_{k-1} = R_k A_k R_k^H$ and $\hat{A}_k$ be non-singular. Assume that for some $\varepsilon > 0$ we have*

$$(1+\varepsilon) A_{k-1} \leq \hat{A}_k \leq \Lambda A_{k-1}.$$

Then $\hat{T}_k$ is non-singular.

Proof. As $T_k = I - P_k (R_k A_k P_k)^{-1} R_k A_k$ is the A-orthogonal projector onto the complement of $\mathrm{ran}(R_k)$, we have $\dim(\ker(T_k)) = \dim(\mathrm{ran}(P_k))$, this is the maximum possible dimension of the kernel of a coarse grid correction. As we have $A_{k-1} < \hat{A}_{k-1}$, we immediately obtain that $\hat{T}_k$ has full rank. □

Obviously with this lemma the last requirement is fulfilled, as $\ker(\hat{T}_k^H A_k \hat{T}_k) = \emptyset$.

3.3.7 Replacement strategies for the Galerkin operator for circulant matrices with compact stencils

Our original goal was to provide an alternative to the usage of the Galerkin operator for circulant matrices with compact stencils, like the ones presented as motivation at the beginning of Section 3.3.2. We will now give examples of replacement strategies, that guarantee multigrid performance, as the prerequisites of the theory presented in the former sections are fulfilled. We do this by analyzing the generating symbols. We start by a general result on d-variate periodic functions.

Lemma 3.12 *Let $f, \hat{f} \in C^2 : [-\pi,\pi)^d \to \mathbb{R}_0^+$ be two nonnegative non-vanishing periodic functions on $[0,2\pi)^d$ having only common zeros and that for some $\varepsilon > 0$ we have $\hat{f} \geq (1+\varepsilon) f$. Furthermore, assume that there are only finitely many such zeros $\mathbf{x}^*$ and that they all satisfy*

$$\nabla^2 f(\mathbf{x}^*) \text{ is positive definite and } \nabla^2 \hat{f}(\mathbf{x}^*) \text{ is positive definite.}$$

Then there exists a constant $\Lambda > 1$ *such that*

$$\hat{f}(\mathbf{x}) \leq \Lambda f(\mathbf{x}) \text{ for all } \mathbf{x} \in [-\pi, \pi)^d.$$

Proof. Let $\mathbf{x}^*$ be a zero of f and $\hat{f}$. Since $\nabla^2 f(\mathbf{x}^*)$ as well as $\nabla^2 \hat{f}(\mathbf{x}^*)$ are positive definite for all $\mathbf{v} \in \mathbb{R}^d, \mathbf{v} \neq 0$ we have

$$0 < \frac{\lambda_{\min}(\nabla^2 \hat{f}(\mathbf{x}^*))}{\lambda_{\max}(\nabla^2 f(\mathbf{x}^*))} \leq \frac{\mathbf{v}^T \nabla^2 \hat{f}(\mathbf{x}^*)\mathbf{v}}{\mathbf{v}^T \nabla^2 f(\mathbf{x}^*)\mathbf{v}} \leq \frac{\lambda_{\max}(\nabla^2 \hat{f}(\mathbf{x}^*))}{\lambda_{\min}(\nabla^2 f(\mathbf{x}^*))} < \infty.$$

By continuity, and since we only have finitely many (common) zeros of f and $\hat{f}$ in $[0, 2\pi)^d$, there exists $\tilde{\varepsilon} > 0$ and $\tilde{\Lambda}$ such that whenever $\|\mathbf{x} - \mathbf{x}^*\| < \tilde{\varepsilon}$ and $\|\mathbf{y} - \mathbf{x}^*\| < \tilde{\varepsilon}$ we have

$$\frac{\mathbf{v}^T \nabla^2 \hat{f}(\mathbf{x})\mathbf{v}}{\mathbf{v}^T \nabla^2 f(\mathbf{y})\mathbf{v}} \leq \tilde{\Lambda}.$$

Using the Taylor expansion

$$\begin{aligned} f(\mathbf{x}) &= f(\mathbf{x}^*) + \nabla f(\mathbf{x}^*)^T(\mathbf{x} - \mathbf{x}^*) + \frac{1}{2}(\mathbf{x} - \mathbf{x}^*)^T \nabla^2 f(\mathbf{x}^* + \theta(\mathbf{x} - \mathbf{x}^*))(\mathbf{x} - \mathbf{x}^*) \\ &= \frac{1}{2}(\mathbf{x} - \mathbf{x}^*)^T \nabla^2 f(\mathbf{x}^* + \theta(\mathbf{x} - \mathbf{x}^*))(\mathbf{x} - \mathbf{x}^*), \ \theta \in [0, 1], \end{aligned}$$

and similarly for $\hat{f}$, we see that whenever $\|\mathbf{x} - \mathbf{x}^*\| < \tilde{\varepsilon}$ for some zero $\mathbf{x}^*$ we have

$$\hat{f}(\mathbf{x}) \leq \tilde{\Lambda} f(\mathbf{x}).$$

The complement C in $[0, 2\pi]^d$ of these finitely many balls is compact, and the function $\hat{f}/f$ is continuous and positive on C. Putting

$$\Lambda = \max \left\{ \tilde{\Lambda}, \max_{\mathbf{x} \in C} \left(\frac{\hat{f}(\mathbf{x})}{f(\mathbf{x})} \right) \right\} \ (< \infty),$$

we finally obtain

$$\hat{f}(\mathbf{x}) \leq \Lambda f(\mathbf{x}) \text{ for all } \mathbf{x} \in [-\pi, \pi)^d.$$

□

This lemma provides all necessary conditions to formulate concrete schemes for the replacement of the Galerkin operator. First we consider the replacement of a compact 9-point stencil of a 2-level circulant matrix.

Definition 3.30 (Replacement 5-point stencil in 2D) *Let $a, b, c \in \mathbb{R}_0^-$ and let*

$$\begin{bmatrix} c & b & c \\ a & -2(a+b)-4c & a \\ c & b & c \end{bmatrix} \tag{3.28}$$

be a 9-point stencil in 2D. We define the replacement 5-point stencil as

$$(1+\varepsilon)\begin{bmatrix} & b+2c & \\ a+2c & -2(a+b)-8c & a+2c \\ & b+2c & \end{bmatrix}. \tag{3.29}$$

If the Galerkin operator is a member of the matrix sequence defined by a 9-point stencil of the form (3.28), the sparser 5-point stencil defined by (3.29) can be used instead. The generating symbol $\hat{f}$ of the circulant matrix sequence defined by the 9-point stencil (3.28) is given by

$$f(x,y) = -2(a+b) - 4c + 2a\cos(x) + 2b\cos(y) + 4c\cos(x)\cos(y).$$

It is non-negative and has a unique zero at the origin with vanishing gradient. The same holds for the generating symbol $\hat{g}$ of the 5-point stencil (3.29),

$$\hat{f}(x,y) = (1+\varepsilon)(-2(a+b) - 8c + 2(a+2c)\cos(x) + 2(b+2c)\cos(y)).$$

Moreover,

$$\nabla^2 f(0,0) = \begin{pmatrix} -2a & 0 \\ 0 & -2b \end{pmatrix} \text{ and}$$
$$\nabla^2 \hat{f}(0,0) = \begin{pmatrix} -2(a+2c) & 0 \\ 0 & -2(b+2c) \end{pmatrix}$$

are both positive definite and for some $\varepsilon > 0$ we have

$$(1+\varepsilon) f(x,y) \leq \hat{f}(x,z).$$

So all requirements are fulfilled and a method using this modified coarse grid operator still converges. Analogously, a replacement stencil can be defined for 3-level circulant matrices

Definition 3.31 (Replacement 7-point stencil in 3D) *Let $a, b, c, d, e, f, g \in \mathbb{R}_0^-$ and let*

$$\begin{bmatrix} g & f & g \\ e & c & e \\ g & f & g \end{bmatrix}$$
$$\begin{bmatrix} d & & b & & d \\ a & -2(a+b+c)-4(d+e+f)-8g & & & a \\ d & & b & & d \end{bmatrix}$$
$$\begin{bmatrix} g & f & g \\ e & c & e \\ g & f & g \end{bmatrix}$$

be a 27-point stencil in 3D. We define the associated 7-point stencil as

$$(1+\varepsilon)\begin{bmatrix} c+2(e+f)+4g \end{bmatrix}$$

$$(1+\varepsilon)\begin{bmatrix} & b+2(d+f)+4g & \\ a+2(d+e)+4g & -2(a+b+c)-8(d+e+f)-16g & a+2(d+e)+4g \\ & b+2(d+f)+4g & \end{bmatrix}$$

$$(1+\varepsilon)\begin{bmatrix} c+2(e+f)+4g \end{bmatrix}.$$

In a similar manner as before – we refrain from reproducing all the details – the corresponding generating functions

$$\begin{aligned} f(x,y,z) &= -2(a+b+c)-4(d+e+f)-8g+2a\cos(x)+2b\cos(y)+2c\cos(z) \\ &\quad +4d\cos(x)\cos(y)+4e\cos(x)\cos(z)+4f\cos(y)\cos(z) \\ &\quad +8g\cos(x)\cos(y)\cos(z) \\ \hat{f}(x,y,z) &= (1+\varepsilon)(-2(a+b+c)-8(d+e+f)-16g+(a+2(d+e)+4g)\cos(x) \\ &\quad +(b+2(d+f)+4g)\cos(y)+(c+2(e+f)+4g)\cos(z)) \end{aligned}$$

can be shown to again have a unique common zero at 0, thus fulfilling all postulated conditions.

The application to stencils of other shapes or involving generating symbols with zeros at other positions can be done in the same way.

3.3.8 Numerical Examples

We tested our replacement strategy in different settings. In contrast to the theory we always chose $\varepsilon = 0$, as this did not harm convergence. This is an indicator that this requirement can probably be skipped. We start with some experiments for 2-level circulant matrices where the replacement has almost no influence on the convergence rate. Both, the standard model problem with linear interpolation and full-weighting and a non-standard problem, involving a zero of the generating symbol which is not at the origin, are presented. After the examples for the 2-level circulant matrices we present an example for 3-level circulant matrices, where the generating symbol has a zero at the origin, again.

5-point Laplacian in 2D

First we consider the standard model problem of Poisson's equation in 2D with periodic boundary conditions yielding a circulant coefficient matrix of the linear system arising

from a discretization using the well-known 5-point stencil

$$\begin{bmatrix} & -1 & \\ -1 & 4 & -1 \\ & -1 & \end{bmatrix}.$$

The symbol

$$p(x,z) = \frac{1}{8}(2 - 2\cos(x-\pi))(2 - 2\cos(y-\pi))$$

was used for interpolation, thus the stencil describing $\mathcal{A}(p)$ is given by

$$\begin{bmatrix} \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \end{bmatrix},$$

resulting in the Galerkin coarse grid operator given by the stencil

$$\begin{bmatrix} -\frac{1}{64} & -\frac{1}{32} & -\frac{1}{64} \\ -\frac{1}{32} & \frac{6}{32} & -\frac{1}{32} \\ -\frac{1}{64} & -\frac{1}{32} & -\frac{1}{64} \end{bmatrix}.$$

The Galerkin operator has been replaced by the operator described by the following stencil, which was chosen in the way defined in Definition 3.30.

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

This coincides with the original stencil multiplied by $1/16$. Due to the factor $1/h^2 = 1/4$ from the doubling of the grid-spacing and another factor of $1/4$ from the inter-grid transfer operators defined with the help of p, the proposed method is equivalent to standard geometric multigrid method in this case. A plot of the associated generating symbols can be found in Fig. 3.4. Fig. 3.5 reports the convergence behavior of the method going down to the level that contains one variable only. As expected, the convergence of the method is only marginally affected by the use of the replacement coarse grid operators.

Example with a zero which is not the origin

Our next example is the stencil

$$\begin{bmatrix} & 1 & \\ -1 & 4 & -1 \\ & 1 & \end{bmatrix},$$

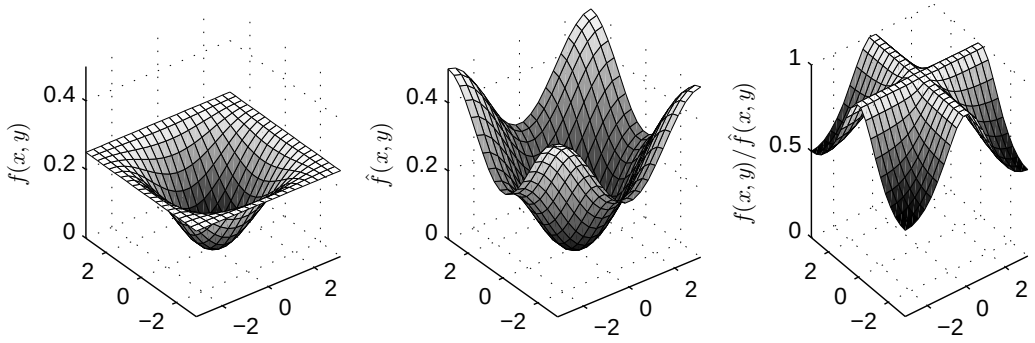

Figure 3.4: Generating symbols f of the Galerkin coarse grid operator for the 5-point discretization of Poisson's equation, $\hat{f}$ of the replacement operator and of the ratio $f/\hat{f}$.

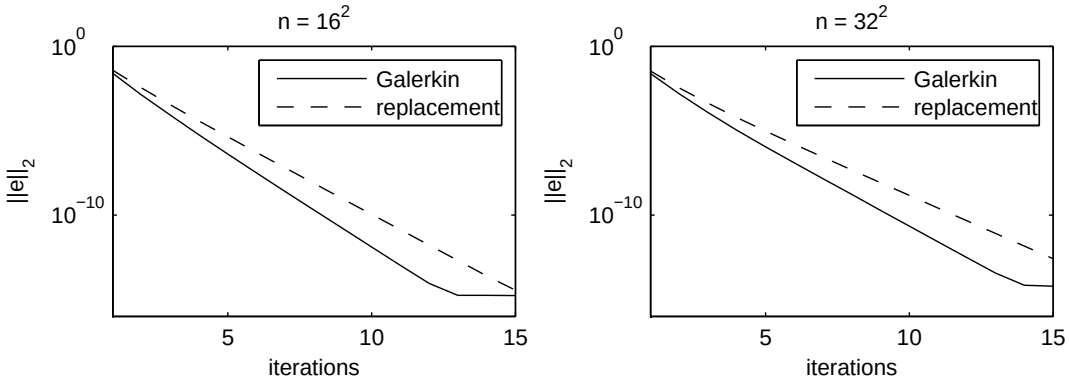

Figure 3.5: Convergence of the multigrid method for the 5-point Laplacian using the Galerkin operator and the replacement operator for $n = 16^2$ and $n = 32^2$.

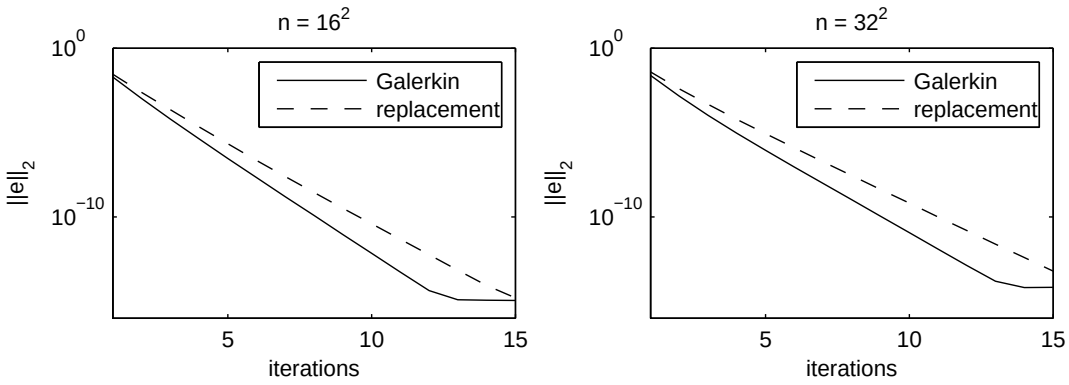

Figure 3.6: Convergence of the multigrid method for the example with zero at $(0, \pi)$ rather than at the origin using the Galerkin operator and the replacement operator for $n = 16^2$ and $n = 32^2$.

as it can be found in [78], e.g. Such a stencil cannot be handled by standard geometric multigrid methods. We chose the symbol for the interpolation as

$$p(x, z) = \frac{1}{8}(2 - 2\cos(x - \pi))(2 - 2\cos(y)),$$

as suggested by Serra Capizzano and Tablino-Possio in [74], so that $\mathcal{A}(p)$ is described by the stencil

$$\begin{bmatrix} -\frac{1}{8} & -\frac{1}{4} & -\frac{1}{8} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ -\frac{1}{8} & -\frac{1}{4} & -\frac{1}{8} \end{bmatrix}.$$

The Galerkin operator is then given by the (scaled) stencil of the standard Poisson problem

$$\begin{bmatrix} -\frac{1}{16} & -\frac{1}{8} & -\frac{1}{16} \\ -\frac{1}{8} & \frac{3}{4} & -\frac{1}{8} \\ -\frac{1}{16} & -\frac{1}{8} & -\frac{1}{16} \end{bmatrix}.$$

The convergence of the stencil collapsing multigrid method going down to the maximum possible level is depicted in Fig. 3.6. The results are very similar to the results for the standard model problem. In particular, the convergence rate degrades only marginally as compared to the multigrid using the Galerkin operators.

7-point Laplacian in 3D

The 3D test is again the model problem, i.e. the 7-point stencil for the 3D-Laplacian. It is given by

$$\begin{bmatrix} & & \\ & -1 & \\ & & \end{bmatrix} \begin{bmatrix} & -1 & \\ -1 & 6 & -1 \\ & -1 & \end{bmatrix} \begin{bmatrix} & & \\ & -1 & \\ & & \end{bmatrix}.$$

n	# iterations	final rel. residual	time per iteration	total time
16^3	6	$2.6446 \cdot 10^{-7}$	$0.0047\,s$	$0.0308\,s$
32^3	6	$3.4160 \cdot 10^{-7}$	$0.0352\,s$	$0.2215\,s$
64^3	6	$3.4430 \cdot 10^{-7}$	$0.2833\,s$	$1.7576\,s$
128^3	6	$3.4429 \cdot 10^{-7}$	$2.2263\,s$	$13.7980\,s$

Table 3.1: Convergence of the multigrid method for the 7-point Laplacian in 3D using the Galerkin coarse grid operator.

n	# iterations	final rel. residual	time per iteration	total time
16^3	7	$1.4726 \cdot 10^{-7}$	$0.0024\,s$	$0.0182\,s$
32^3	7	$1.5726 \cdot 10^{-7}$	$0.0165\,s$	$0.1255\,s$
64^3	7	$1.5813 \cdot 10^{-7}$	$0.1333\,s$	$0.9830\,s$
128^3	7	$1.5853 \cdot 10^{-7}$	$1.0347\,s$	$7.5916\,s$

Table 3.2: Convergence of the multigrid method for the 7-point Laplacian in 3D using the replacement grid operator.

The interpolation is defined by the symbol

$$p(x, y, z) = \frac{1}{8}(2 - 2\cos(x - \pi))(2 - 2\cos(y - \pi))(2 - 2\cos(z - \pi)).$$

The resulting Galerkin coarse grid operator has 19 entries, and the Galerkin operators on all subsequent levels have 27 entries. The stencil collapsing multigrid method was incorporated into a multigrid code for 3-level circulant matrices, thus keeping the size of the stencils corresponding to the coarse grid operators constantly at 7.

In order to measure timings for 3D problems, a multigrid method for circulant matrices with generating symbols having zeros at the origin was implemented in C and compiled using the gcc compiler with O3-optimization. The Galerkin coarse grid operator was formed automatically on each level and the replacement given in Definition 3.31 was computed automatically as well. The measurements were taken on a Linux machine with 3.2 GHz Pentium 4 CPU. The times needed by the method to reduce the relative residual to 10^{-7} using the Galerkin coarse grid operator can be found in Table 3.1, the ones for the replacement operator are given in Table 3.2. It can be seen that one additional iteration is needed when using the Galerkin coarse grid operator, but the execution using the replacement operator is much faster.

3.4 Parallelization

Parallelization of algorithms of numerical linear algebra is an important part of the development of scientific applications, as many applications from different fields of research

spend a lot of time in these routines. For that purpose various books with a special focus on parallelization have been published, for example the book by Golub and Ortega [42] or the book by Frommer [37]. Albeit multigrid algorithms are very fast and efficient methods for the solution of linear systems and although our extension to the theory allows additional savings in terms of CPU cycles and wall clock time, the parallelization of multigrid still can be necessary for two reasons:

1. The lack of memory on one node when the system that should be solved is too large.
2. Parallelization is necessary because of the computational requirements of the underlying problem, that requires the solution of the linear system.

While the first is relatively easy to understand, we like to emphasize the second part a little bit more. If the underlying problem that requires to solve the system, is computationally complex, for example because forming the right hand side of our linear system costs a lot of time, it might be necessary to parallelize the problem. It would be unsatisfactory not to parallelize the multigrid part, because due to Amdahl's law, the speedup will be bound by the time spent in the solution of the linear system.

The parallelization of multigrid methods is well analyzed. For an overview see the work of Chow, Falgout, Hu, Tuminaro and Yang [18], a more detailed introduction and analysis of the parallelization of geometric multigrid methods can be found in the PhD thesis of Tuminaro [85]. Our parallel implementation, which was used to produce the results in Section 3.4.2, is kept as simple as possible, i.e. a data distribution scheme is chosen that is equivalent to a domain decomposition approach, and processors become idle when there are no variables left that belong to them. It shares this concept with the code of Ashby and Falgout introduced in [4] that is a predecessor to the structured multigrid code that is contained in the hypre package [28, 29]. Other parallelization approaches, especially some that utilize idle processors on coarse levels, are possible, but they are not covered, here.

3.4.1 Data distribution for banded matrices

What we want to do is solving a linear system on a parallel computer. In the cases we are interested in, here, we deal with banded circulant matrices, although the chosen approach can be transferred to band matrices with similar structures, as well. For our algorithms we need matrix vector multiplication with a matrix $A := \mathcal{A}(f)$ and transfer of the vectors, only. We start with 1-level circulants with a fixed bandwidth m, that is independent of the system size n. That means that in order to calculate the i-th entry of the matrix-vector product we only need the information of the entries that have indices from $\mathcal{I}_{i,m,n}$, where

$$\mathcal{I}_{i,m,n} = \{(i-m) \bmod n, \ldots, (i-1) \bmod n, i, (i+1) \bmod n, \ldots, (i+m) \bmod n\}.$$

Concretely we have

$$(A\mathbf{x})_i = \sum_{j \in \mathcal{I}_{i,m,n}} a_{i,j} x_j.$$

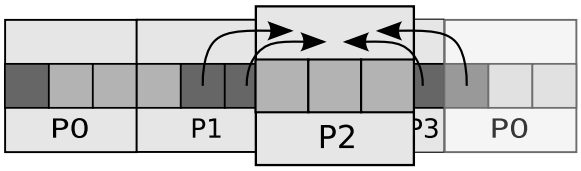

Figure 3.7: Communication pattern for a vector with 10 components, distributed to 4 processors. Highlighted is processor P2 and its communication, when $m = 2$ neighbors are needed for the matrix vector multiplication.

In order to evaluate this product on a parallel computer it is favorable to have as much as possible of this information stored locally. Therefore we choose to distribute the vector over the processors block-wise, i.e. when we have p processors the i-th processor gets the components ranging from $(i-1)\lceil n/p \rceil$ to $\min\{i\lceil n/p \rceil, n\}$. Using this distribution the processors are logically arranged in a 1-D torus and they only have to exchange components with $\lceil m/\min\{\lceil n/p \rceil, n-(p-1)\lceil n/p \rceil\}\rceil$ neighbors in a one-dimensional torus. An outline of the communication needed can be found in Figure 3.7. As multilevel circulant matrices are formed by the use of tensor products, this concept can be carried over to that case, as well. As long as the bandwidth of the according circulant matrices is fixed and independent of n_i, the same communication pattern can be used in d different directions in the d-level case. So the optimal communication topology for circulant matrices is a d-dimensional torus. Obviously for the non-periodic case, that leads to a Toeplitz matrix, a d-dimensional mesh is sufficient. Going down to the coarser levels, the locality of a variable on the fine level determines on which processor the coarse level variable will be located. The variables on the coarse level are located on the same processor as their fine grid counterpart. This leads to a structured communication scheme on the lower levels. Starting with a d-dimensional torus we have communication with the next neighbors holding the m needed components, as long as all processors still have variables to treat. At some point, namely when the number of processors in one direction is bigger than the number of unknowns, we will have idle processors, which do not hold any variable on that level anymore. These processors then have to be ignored, when the communication takes place. Technically we tackle this issue by storing the neighborhood information on each level. In the initialization step processors ask the neighbors of the previous level which neighbor they should use on this level. The asked processor answers this question with its own id, if it still has to do work, or with its own neighbor. Of course this scheme requires that only every second processor may become idle per level, but that is guaranteed if the unknowns are equally distributed on the finest level at the beginning. Otherwise it could be fixed by providing a function that computes the corresponding variable on the finest level, eventually combined with a distributed directory of the variable location, like proposed in the work of Baker, Falgout and Yang [5].

3.4.2 Example results on Blue Gene/L and Blue Gene/P

The algorithm was implemented in the C programming language, using MPI for the distributed memory parallelization. As the torus is well-suited to implement the communication pattern of the algorithm, the implementation makes use of cartesian communicators and the associated functions. The implementation was tested on both Blue Gene systems of the Jülich Supercomputing Centre, the 8-rack Blue Gene/L system JUBL [71] and the 16-rack Blue Gene/P system JUGENE [72]. Both Blue Gene generations, the Blue Gene/L and the Blue Gene/P, consist of several racks, where each rack consists of two midplanes with 512 nodes each. The nodes are designed as systems on chip, i.e. one chip contains all necessary components as the processor itself, network adaptors, memory controllers etc., where each system has two cores in the Blue Gene/L and four cores in the Blue Gene/P. The chips are clocked at 700 MHz in the Blue Gene/L, in the Blue Gene/P the clock rate has been raised to 850 MHz. Besides Gigabit networking for communication with the outside world, a very fast interrupt network, and a network for system management purpose, the Blue Gene architecture has two networks that are used for the communication of the parallel programs. These are a torus network that is used for point to point communication and a tree network for collective communication. For an overview of the Blue Gene/L architecture see the article of Gara et al. [39]. Further details can be found on the web pages on JUBL [71] and JUGENE [72] and the references therein.

The implementations of the solver for circulant matrices using the Galerkin operator and of the one using the replacement were tested in different configurations. First we like to emphasize, that the use of a V-cycle instead of a W-cycle is mandatory. Not only is a W-cycle in general slower than a V-cycle, but in the W-cycle the amount of time the multigrid method spends in the coarser levels is much larger than in the V-cycle and many more communication steps are necessary. To illustrate that, we refer to Figure 3.8, where the weak scaling behavior of the V-cycle and the W-cycle using the Galerkin operator for a system with $64 \times 128 \times 128$ unknowns per processor are depicted. The tests were carried out on JUBL and the system was arising from a 7-point discretization of the Laplacian with periodic boundary conditions. It is clear, that the W-cycle's performance decreases in the parallel case, thus the effort spend in order to proof V-cycle convergence in Section 3.3.2 is necessary in the parallel case. Otherwise the time that is saved by the replacement of the Galerkin operator gets lost in the parallel case or even more time is spent.

To illustrate the good scaling results of the V-cycle using the Galerkin operator as well as the replacement we ran a number of tests. Strong scaling results on up to one rack of Blue Gene/L were obtained for both for a 7-point discretization of the Laplace operator with periodic boundary conditions resulting in a system with 128^3 unknowns. The timings for the solution of the system up to an absolute error of 10^{-7} of this case are found in Table 3.3, a plot of the speedup and the efficiency can be found in Figure 3.9. Obviously the replacement of the operator does not harm the scaling behavior of the method, although the time needed to solve the system is smaller and thus the ratio of communication and

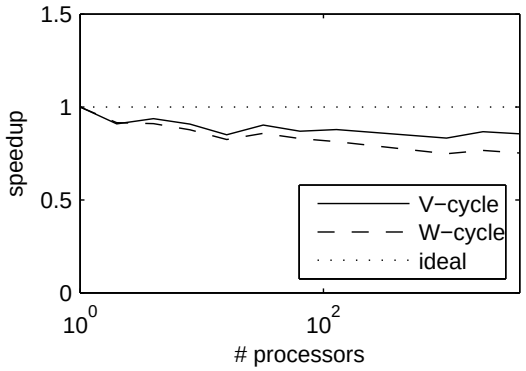

Figure 3.8: Speedup for the V-cycle and the W-cycle compared.

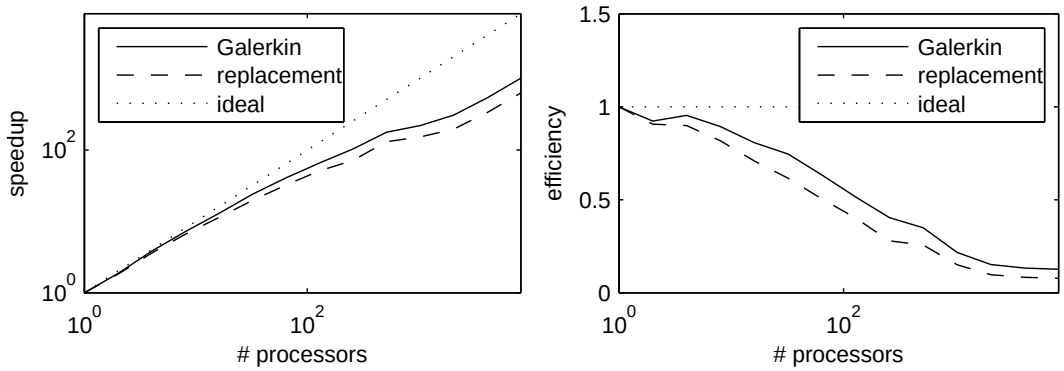

Figure 3.9: Speedup and efficiency on Blue Gene/L for the solution of a system with 128^3 unknowns arising from the discretization of the Laplacian using a 7-point stencil.

computation is even worse than in the case, where the Galerkin operator was used. We like to emphasize that although the scaling curves do not look very impressive, the results are nevertheless pretty good, considering that the system consists of 128^3 unknowns, only. This is the largest problem, that can be solved on a single Blue Gene/L node and could thus be easily solved on a desktop PC, as the node of a Blue Gene is much slower than todays' PCs. Nevertheless we increase the number of nodes to 1024, as a result each node is responsible for handling 2048 unknowns on the finest level, only. Additionally we ran some tests on the newly installed Blue Gene/P system for the V-cycle using the Galerkin operator. The behavior of the method using the replacement operator should be similar. In the test, a system with 1024^3 unknowns has been solved, see Table 3.4 and Figure 3.4 for the results. Again, the system was arising from a 7-point discretization of the Laplacian with periodic boundary conditions. We can see that the scaling looks much better for this case, although we have to mention that the amount of data is 64 times as big. Regarding weak scaling the results are very good. The results of a run where each processor has $64 \times 128 \times 128$ unknowns are as expected, see the measurements in Table 3.5 and the plot of this data in Figure 3.11.

	Galerkin operator		replacement operator	
#processors	time/iteration	total time	time/iteration	total time
1	$3.218399 \cdot 10^{0}$	$3.604389 \cdot 10^{1}$	$1.896993 \cdot 10^{0}$	$2.138000 \cdot 10^{1}$
2	$1.741969 \cdot 10^{0}$	$1.951582 \cdot 10^{1}$	$1.045804 \cdot 10^{0}$	$1.178795 \cdot 10^{1}$
4	$8.436338 \cdot 10^{-1}$	$9.454753 \cdot 10^{0}$	$5.272539 \cdot 10^{-1}$	$5.947049 \cdot 10^{0}$
8	$4.503158 \cdot 10^{-1}$	$5.045441 \cdot 10^{0}$	$2.902817 \cdot 10^{-1}$	$3.270748 \cdot 10^{0}$
16	$2.493376 \cdot 10^{-1}$	$2.790456 \cdot 10^{0}$	$1.677954 \cdot 10^{-1}$	$1.887067 \cdot 10^{0}$
32	$1.351773 \cdot 10^{-1}$	$1.510425 \cdot 10^{0}$	$9.678527 \cdot 10^{-2}$	$1.085267 \cdot 10^{0}$
64	$7.951982 \cdot 10^{-2}$	$8.889820 \cdot 10^{-1}$	$5.906118 \cdot 10^{-2}$	$6.629070 \cdot 10^{-1}$
128	$4.887073 \cdot 10^{-2}$	$5.466090 \cdot 10^{-1}$	$3.710509 \cdot 10^{-2}$	$4.169230 \cdot 10^{-1}$
256	$3.117418 \cdot 10^{-2}$	$3.487930 \cdot 10^{-1}$	$2.662664 \cdot 10^{-2}$	$2.989930 \cdot 10^{-1}$
512	$1.794464 \cdot 10^{-2}$	$2.019890 \cdot 10^{-1}$	$1.440055 \cdot 10^{-2}$	$1.634570 \cdot 10^{-1}$
1024	$1.443436 \cdot 10^{-2}$	$1.627610 \cdot 10^{-1}$	$1.227636 \cdot 10^{-2}$	$1.393520 \cdot 10^{-1}$
2048	$1.029345 \cdot 10^{-2}$	$1.164280 \cdot 10^{-1}$	$9.510182 \cdot 10^{-3}$	$1.085170 \cdot 10^{-1}$
4096	$5.794727 \cdot 10^{-3}$	$6.665800 \cdot 10^{-2}$	$5.452455 \cdot 10^{-3}$	$6.333800 \cdot 10^{-2}$
8192	$2.941091 \cdot 10^{-3}$	$3.515200 \cdot 10^{-2}$	$2.787636 \cdot 10^{-3}$	$3.370500 \cdot 10^{-2}$

Table 3.3: Timings on Blue Gene/L for the solution of a system with 128^3 unknowns arising from the discretization of the Laplacian using a 7-point stencil.

# processors	average time per iteration
4096	$5.216130 \cdot 10^{-1}$
8192	$2.789460 \cdot 10^{-1}$
16384	$1.938290 \cdot 10^{-1}$
32768	$7.484900 \cdot 10^{-2}$
65536	$4.131500 \cdot 10^{-2}$

Table 3.4: Timings on Blue Gene/P for the solution of a system with 1024^3 unknowns arising from the discretization of the Laplacian using a 7-point stencil.

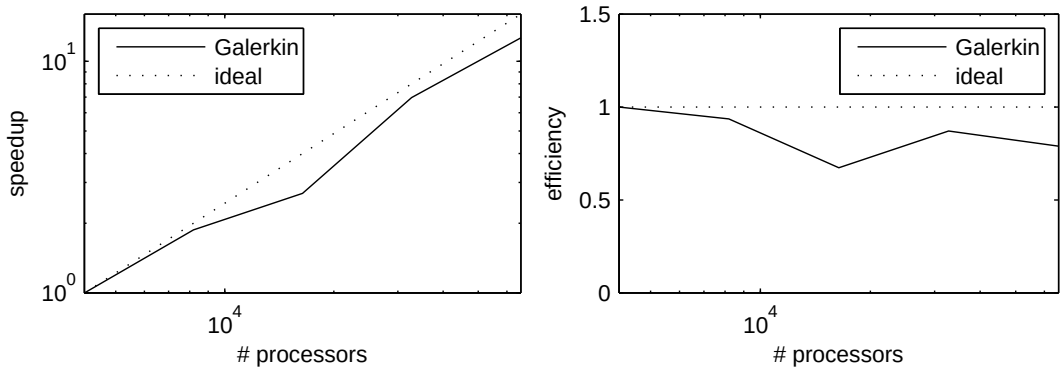

Figure 3.10: Speedup and efficiency relative to one rack with 4096 processors on Blue Gene/P for the solution of a system with 1024^3 unknowns arising from the discretization of the Poisson equation using a 7-point stencil.

	Galerkin operator		replacement operator	
#processors	time/iteration	total time	time/iteration	total time
1	$1.586269 \cdot 10^0$	$1.776711 \cdot 10^1$	$9.536420 \cdot 10^{-1}$	$1.074958 \cdot 10^1$
2	$1.741970 \cdot 10^0$	$1.951583 \cdot 10^1$	$1.045803 \cdot 10^0$	$1.178818 \cdot 10^1$
4	$1.686735 \cdot 10^0$	$1.890513 \cdot 10^1$	$1.013224 \cdot 10^0$	$1.143372 \cdot 10^1$
8	$1.742680 \cdot 10^0$	$1.952863 \cdot 10^1$	$1.016574 \cdot 10^0$	$1.146883 \cdot 10^1$
16	$1.857210 \cdot 10^0$	$2.079570 \cdot 10^1$	$1.084866 \cdot 10^0$	$1.221705 \cdot 10^1$
32	$1.758144 \cdot 10^0$	$1.969952 \cdot 10^1$	$1.041178 \cdot 10^0$	$1.174254 \cdot 10^1$
64	$1.824098 \cdot 10^0$	$2.043441 \cdot 10^1$	$1.059706 \cdot 10^0$	$1.195079 \cdot 10^1$
128	$1.885549 \cdot 10^0$	$2.111226 \cdot 10^1$	$1.087700 \cdot 10^0$	$1.225203 \cdot 10^1$
256	$1.856749 \cdot 10^0$	$2.080243 \cdot 10^1$	$1.059373 \cdot 10^0$	$1.194493 \cdot 10^1$
512	$1.843628 \cdot 10^0$	$2.065635 \cdot 10^1$	$1.018313 \cdot 10^0$	$1.148949 \cdot 10^1$
1024	$1.919460 \cdot 10^0$	$2.149173 \cdot 10^1$	$1.085729 \cdot 10^0$	$1.222963 \cdot 10^1$
2048	$1.976223 \cdot 10^0$	$2.213161 \cdot 10^1$	$1.191898 \cdot 10^0$	$1.341163 \cdot 10^1$
4096	$1.970838 \cdot 10^0$	$2.207169 \cdot 10^1$	$1.187782 \cdot 10^0$	$1.336699 \cdot 10^1$
8192	$1.923521 \cdot 10^0$	$2.153583 \cdot 10^1$	$1.090093 \cdot 10^0$	$1.227793 \cdot 10^1$

Table 3.5: Weak scaling results on Blue Gene/L for different numbers of unknowns for the discretization of the Laplacian with periodic boundary conditions using a 7-point stencil. Each Processor has $64 \times 128 \times 128$ unknowns on the finest level.

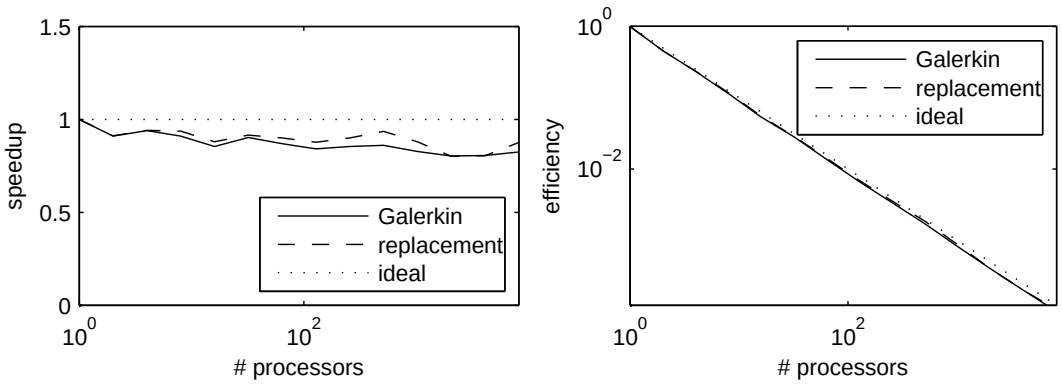

Figure 3.11: Speedup and efficiency for the weak scaling test on Blue Gene/L for different numbers of unknowns for the discretization of the Laplacian with periodic boundary conditions using a 7-point stencil. Each Processor has $64 \times 128 \times 128$ unknowns on the finest level.

3.4.3 Further parallelization issues

What we have not covered here is the parallelization of the FAC method introduced in Section 3.2.4. Although the communication pattern will be more involved, the problem still possesses a lot of structure that can be exploited for the solution on a parallel system.

Besides massively parallel systems that are similar to the Blue Gene architecture, recently multicore architectures became more and more important. One famous member of this family is the hybrid multicore architecture Cell Broadband Engine Architecture or CBEA for short. We investigated the usefulness of the CBEA for multigrid methods for structured matrices and published some ideas and preliminary results in [9]. A general analysis of the CBEA for scientific applications can be found in [88].

Chapter 4

Particle Simulation

4.1 Introduction

Particle simulation plays an important role in computational science. For many fields of applications the simulation of atomistic particles using simple integration of Newton's equations of motion is sufficient. Considering e.g. astrophysics computer experiments are the only choice to verify new models, as the studied phenomenon can not be influenced by the researcher and the time-scales in question are far too large. Another example is the field of biophysics, which became more and more important in recent years. Here, computer experiments help to save a lot of money, as the experiments that have to be conducted are very expensive and time-consuming. So computer experiments are used to have a guideline, which experiments one wants to carry out in reality. At the Jülich Supercomputing Centre there exists the complex atomistic modeling and simulation group, where scientists with different backgrounds and applications work on the development of particle simulation methods. Most of these methods are highly scalable, as a huge amount of supercomputer time is spend in particle simulation codes. The method that will be described in the following was developed as part of the work in this group that led to this thesis.

Given that computers became available in the middle of the last century, the field of particle simulation is relatively old. As a consequence a huge number of algorithms using different techniques and approximations exist. In the following, we will present a short introduction into the problem. A more detailed overview on classical molecular dynamics is given by Sutmann [81], and an overview over long-range interactions by Gibbon and Sutmann [40], introductions with larger details can be found in the books of Hockney and Eastwood [56] and in the book of Griebel, Knapek, Zumbusch and Caglar [46]. After the introduction we give a brief overview over the available methods for particle simulation, and finally present the approach that allows us to use multigrid methods in the context of particle simulation.

4.2 Mathematical formulation

Given is an initial state $\mathcal{S}_0 = [\mathbf{x}_1, \ldots, \mathbf{v}_1, \ldots]$ of a, not necessarily finite, set $\mathcal{P}$ of particles. In classical mechanics the system is described completely by this set, i.e. the coordinates and the velocities of the particles. The time evolution of the system is described by Newton's equations of motion, i.e.

$$\mathbf{v}_i = \frac{d}{dt}\mathbf{x}_i,$$
$$\mathbf{F}_i = \frac{d}{dt}m_i\mathbf{v}_i$$

for a particle with index i. The force acting on particle i is given by the sum of the forces due to all other particles in the system, i.e.

$$\mathbf{F}_i = \sum_{i \in \mathcal{P}\setminus\{i\}} \mathbf{F}_{i,j}. \tag{4.1}$$

In some cases an external force may be present as well. The forces are given by the gradient of the potentials, yielding

$$\mathbf{F}_i = -\nabla\Phi_i, \text{ respectively } \mathbf{F}_{i,j} = -\nabla\Phi_{i,j}, \tag{4.2}$$

where

$$\Phi_i = \sum_{i \in \mathcal{P}\setminus\{i\}} \Phi_{i,j}. \tag{4.3}$$

So the evolution of the system is a consequence of the effective potential. Depending on the type of application different potentials are used, e.g.

1. Coulomb potential
$$\Phi_{i,j} = \frac{1}{4\pi\varepsilon_0}\frac{q_j}{\|\mathbf{x}_i - \mathbf{x}_j\|_2}, \tag{4.4}$$
2. Gravitational potential
$$\Phi_{i,j} = -G\frac{m_j}{\|\mathbf{x}_i - \mathbf{x}_j\|_2},$$
3. Van der Waals potential
$$\Phi_{i,j} = -a\left(\frac{1}{\|\mathbf{x}_i - \mathbf{x}_j\|_2}\right)^6,$$
4. Lenard-Jones potential
$$\Phi_{i,j} = \alpha\varepsilon\left[\left(\frac{\sigma}{\|\mathbf{x}_i - \mathbf{x}_j\|_2}\right)^n - \left(\frac{\sigma}{\|\mathbf{x}_i - \mathbf{x}_j\|_2}\right)^m\right], m < n,$$
$$\alpha := \frac{1}{n-m}\left(\frac{n^n}{m^m}\right)^{\frac{1}{n-m}}.$$

We differentiate potentials by their range, i.e. a potential that decays faster than $1/r^d$, where d is the space dimension, is called a short-ranged potential, whereas potentials decaying at least as slowly as that are called long-ranged potentials. Short ranged potentials, like the Van der Waals potential or the Lenard-Jones potential, can be easily evaluated using list-techniques, like the the linked list array (see [56]).

While we need methods for short-ranged potentials later on to correct artificially introduced errors in our potential, we cover the Coulomb potential, here. Differentiating the potential energy, i.e. the potential of the particle times it's charge, leads to the force that is acting on a particle. The Coulomb potential is one of the most important potentials, as it arises in various applications like biophysics and plasma physics. For the forces due to this potential we obtain

$$\mathbf{F}_i = \frac{1}{4\pi\varepsilon_0} \sum_{j \in \mathcal{P}\setminus\{i\}} q_i q_j \frac{\mathbf{x}_i - \mathbf{x}_j}{\|\mathbf{x}_i - \mathbf{x}_j\|_2}. \tag{4.5}$$

Another important quantity of Coulomb systems is the electrostatic energy that can be calculated with the help of the potential, as

$$E = \frac{1}{2} \sum_{i \in \mathcal{P}} q_i \Phi_i = \frac{1}{4\pi\varepsilon_0} \sum_{i \in \mathcal{P}} q_i \sum_{j \in \mathcal{P}\setminus\{i\}} \frac{q_j}{\|\mathbf{x}_i - \mathbf{x}_j\|_2^3}. \tag{4.6}$$

We like to note that the gravitational potential is of the same form as the Coulomb potential, so we are able to cover applications from astrophysics, as well.

In order to simulate a particle system, a time integration scheme is required. For that purpose we use a simple integrator like the Euler integration scheme or a leapfrog scheme. These integration schemes need at least the input of the forces and velocities at one time step and they provide the new positions and updated velocities as output. Integration schemes are not covered by this work, we refer to the books of Hockney and Eastwood [56] or the book of Griebel, Knapek, Zumbusch and Caglar [46], which both cover particle simulation methods in general. The applications in the first book are focussed on plasma physics and astrophysics and the authors of the second book concentrate on biophysical applications. We will focus on methods that calculate the potential of particles and thus provide a way to calculate the forces needed as input to the integrators.

We will now provide a rough overview over the different ways the problem may be posed. Particle systems differ in the domain they cover. In this work we will cover the most important options, namely open and periodic systems.

4.2.1 Open systems

In open systems the set of particles $\mathcal{P}$ is finite and the particles can move in the open space freely. As the number of involved particles is finite, the problem can be directly solved

by evaluating (4.1) or (4.3), utilizing (4.2). As an example consider the total energy of a system of N particles. Substituting $\{1, 2, \ldots, N\}$ for $\mathcal{P}$ in (4.6) yields

$$E = \frac{1}{2} \sum_{i=1}^{N} q_i \sum_{\substack{j=1 \\ j \neq i}}^{N} \frac{1}{4\pi\varepsilon_0} \frac{q_j}{\|\mathbf{x}_i - \mathbf{x}_j\|_2}.$$

Using symmetry we can write

$$E = \sum_{i=1}^{N} q_i \sum_{j=i+1}^{N} \frac{1}{4\pi\varepsilon_0} \frac{q_j}{\|\mathbf{x}_i - \mathbf{x}_j\|_2}.$$

We note that the complexity for evaluating E is quadratic. Therefore methods have been developed that reduce the complexity to $\mathcal{O}(N \log N)$ or even $\mathcal{O}(N)$. The price to pay these methods is accuracy, as they only compute an approximation to the real solution. As all the computations are carried out in floating point arithmetic on a finite computer, this is not necessarily a downside, as the direct calculation is inexact there, as well.

4.2.2 Periodic systems

In periodic systems the set of particles $\mathcal{P}$ is infinite, but the particle distribution itself is periodic and the number of particles in a box representing the whole system is finite. The particles in the box are interacting with each other and with all periodic images of all particles in the box, including the periodic images of the particle itself. As an example we consider the total electrostatic energy of the system, again, which is given by

$$E = \frac{1}{2} \sum_{i=1}^{N} q_i \sum_{\substack{j=1 \\ j \neq i}}^{N} \frac{1}{4\pi\varepsilon_0} \frac{q_j}{\|\mathbf{x}_i - \mathbf{x}_j\|_2} + \sum_{\mathbf{n} \in \mathbb{Z}^3 \setminus \{\mathbf{0}\}} \frac{1}{2} \sum_{i=1}^{N} q_i \sum_{j=1}^{N} \frac{1}{4\pi\varepsilon_0} \frac{q_j}{\|\mathbf{x}_i - \mathbf{x}_j + \mathbf{n}\|_2}. \quad (4.7)$$

Here, without loss of generality, we assume the system to be represented by a cube with side length 1. This system cannot be solved using direct summation anymore, as the sum over $\mathbf{n}$ is infinite. Furthermore this sum is divergent, so other summation techniques have to be used, which take information about the underlying physics into account. An example is the Ewald summation [27], which splits this sum into two parts replacing the point charge by a charge distribution described by a gaussian and correcting this afterwards. The sum can now be split into two parts, where the point charge minus the charge distribution decays very fast and the other part converges very fast after transformation to Fourier space. This approach will be used later on when we discuss the numerical solution scheme used.

4.2.3 Relation to the Poisson equation

There is an obvious connection between the electrostatic potential and the solution of the Poisson equation discussed as model problem in Chapter 2. In Theorem 2.7 we have shown that the Green function of the Poisson equation in $\mathbb{R}^3$ is given by (2.10), i.e.

$$U(\mathbf{x}) = \frac{1}{4\pi\|\mathbf{x}\|_2}.$$

This reminds us of the definition of the Coulomb potential in (4.4). In fact we have that

$$\Phi_i = \sum_{\substack{j=1 \\ j\neq i}}^{N} \frac{1}{4\pi\varepsilon_0} \frac{q_j}{\|\mathbf{x}_i - \mathbf{x}_j\|_2}$$

is a solution of the Poisson equation

$$\Delta\Phi_i(\mathbf{x}) = \rho_i := \frac{1}{\varepsilon_0} \sum_{\substack{j=1 \\ j\neq i}}^{N} q_j \delta(\|\mathbf{x} - \mathbf{x}_j\|_2). \tag{4.8}$$

Therefore we call the solution of this Poisson equation the *potential induced* by all particles except for the i-th particle. Now the potential of particle i is given as $\Phi_i(\mathbf{x}_i)$ and the force acting on it by $-\nabla\Phi_i(\mathbf{x})$. The connection to the Poisson equation provides us with a way to define numerical schemes to calculate the electrostatic quantities of the system that are based on the solution of the Poisson equation on a mesh.

4.3 Numerical solution

Before we come to our numerical method we like to subsume the available methods for the calculation of forces and energies. On the one hand there are mesh-free methods, that directly tackle the sums in (4.5) or (4.6). Other methods exploit the fact that the potential can be evaluated on a mesh, effectively solving the Poisson equation. The forces are obtained by numerical differentiation afterwards.

4.3.1 Mesh-free methods

A $1/r$-term, where r denotes the distance between two particles, is not neglectable, i.e. even particles far away have a noticeable impact on the force. Nevertheless, changes in the position of the other particles that are small compared to the distance, will not induce noticeable changes in neither the potential or the forces. That observation led to the development of tree codes. In the Barnes-Hut tree code [7] the whole simulation domain is put into

a box. By recursively subdividing the box into sub-boxes that are represented by pseudo-particles, the calculation of particle-box interactions is possible. For each particle-box interaction a criterion controls, whether this pair is chosen or whether a further subdivision is chosen. As a consequence, the Barnes-Hut tree code has a complexity of $\mathcal{O}(N \log N)$, where N is the number of particles.

The idea can be extended to not only exploit the idea in one direction, i.e. computing particle-box interactions, but to computing box-box interactions for boxes that are far enough away, as well. This is the basis of the Fast Multipole Method (FMM), that was presented by Greengard and Rokhlin in [45]. They combined this idea with not only taking monopole interactions into account but rather computing multipole interactions, as well. The multipole interactions in principle are a Taylor-Expansion of the potential.

Both methods originally have been developed for open systems. Mesh-free methods for periodic systems include the Ewald summation [27], although there are efforts to extend tree codes to the periodic case, as well. As an example consider the method of Kudin and Scuseria recently presented in [60].

4.3.2 Mesh-based methods

The developed numerical method is mesh-based and thus similar to the P3M, the SPME and the method presented in the diploma thesis of Füllenbach [38]. As mentioned above, these methods exploit the connection between the Poisson equation and the electrostatic potential. All of these methods have in common that they are based on the development of the Ewald summation and thus have been developed for periodic systems. To derive the methods, we start with (4.7), i.e.

$$E = \frac{1}{2}\sum_{i=1}^{N} q_i \sum_{\substack{j=1 \\ j\neq i}}^{N} \frac{1}{4\pi\varepsilon_0}\frac{q_j}{\|\mathbf{x}_i - \mathbf{x}_j\|_2} + \sum_{\mathbf{n}\in\mathbb{Z}^3\setminus\{\mathbf{0}\}} \frac{1}{2}\sum_{i=1}^{N} q_i \sum_{j=1}^{N} \frac{1}{4\pi\varepsilon_0}\frac{q_j}{\|\mathbf{x}_i - \mathbf{x}_j + \mathbf{n}\|_2}.$$

This sum is not absolutely convergent. In order to define the sum's value, we split it using the identity

$$\frac{1}{\|\mathbf{x}_i - \mathbf{x}_j\|_2} = \frac{f(\|\mathbf{x}_i - \mathbf{x}_j\|_2)}{\|\mathbf{x}_i - \mathbf{x}_j\|_2} + \frac{1 - f(\|\mathbf{x}_i - \mathbf{x}_j\|_2)}{\|\mathbf{x}_i - \mathbf{x}_j\|_2},$$

where we choose f, such that $f(\|\mathbf{x}_i - \mathbf{x}_j\|_2)/\|\mathbf{x}_i - \mathbf{x}_j\|_2$ decays very fast and thus can be neglected beyond some cutoff and such that $(1 - f(\|\mathbf{x}_i - \mathbf{x}_j\|_2))/\|\mathbf{x}_i - \mathbf{x}_j\|_2$ is slowly varying, i.e. the Fourier coefficients belonging to large indices become small. As a consequence the first sum can be evaluated like a short-ranged potential and the second sum is calculated by calculating the Fourier sum only up to a certain index. In order to properly Fourier transform the second part of the sum, the so-called "self-energy" E_s has to be

introduced, yielding

$$E = \underbrace{\frac{1}{2}\sum_{i=1}^{N}\sum_{\substack{j=1\\ j\neq i}}^{N}\frac{q_i q_j}{4\pi\varepsilon_0}\frac{f(\|\mathbf{x}_i-\mathbf{x}_j+\mathbf{n}\|_2)}{\|\mathbf{x}_i-\mathbf{x}_j\|_2} + \sum_{\mathbf{n}\in\mathbb{Z}^3\setminus\{\mathbf{0}\}}\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{q_i q_j}{4\pi\varepsilon_0}\frac{f(\|\mathbf{x}_i-\mathbf{x}_j+\mathbf{n}\|_2)}{\|\mathbf{x}_i-\mathbf{x}_j+\mathbf{n}\|_2}}_{=:E_r}$$
$$+ \underbrace{\sum_{\mathbf{n}\in\mathbb{Z}^3}\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{q_i q_j}{4\pi\varepsilon_0}\frac{1-f(\|\mathbf{x}_i-\mathbf{x}_j+\mathbf{n}\|_2)}{\|\mathbf{x}_i-\mathbf{x}_j+\mathbf{n}\|_2}}_{=:E_k} - \underbrace{\frac{1}{2}\sum_{i=1}^{N}\frac{q_i^2}{4\pi\varepsilon_0}f'(0)}_{=:E_s}.$$

The traditional choice for f is the complementary error function

$$\operatorname{erfc}(r) := \frac{2}{\sqrt{\pi}}\int_r^\infty e^{-x^2}dx.$$

This also has a physical interpretation, namely that the point charges are "hidden" by a "charge cloud" of the same charge. The "charge cloud" simply is a distribution of measure one, in the Ewald case it is point-symmetric and described by the error function.

The Ewald approach can easily be transferred to a grid based approach. In that case the particles charges are mapped to grid points in an appropriate way. Several ways exist to do that, the simplest one being the nearest neighbor scheme. More sophisticated approaches split a particle into several pseudo-particles, which reside on the grid point. The charges of these pseudo-particles are then calculated using interpolation schemes or using B-splines. Once the charges are mapped to the mesh, the mesh can be transferred to Fourier space using the FFT and the reciprocal sum E_k can be evaluated there using convolution with the Fourier transformed version of $f(1-\|\mathbf{x}_i-\mathbf{x}_j\|_2)$ or another so-called "influence function". The summation E_r in real (physical) space can be carried out approximately using a cut-off radius. At that point a data structure comes in handy that stores the particles that are contained in a certain grid cell. For that purpose the *linked list* algorithm has been developed (c.f. [56]). It creates a three-dimensional array HOC that contains the index of the first particle inside the corresponding box. Another one-dimensional array LL contains the next particle in that box of each particle. If there is no particle in a box or if a particle has no successor, the entries are set to zero. The algorithm that creates the data structures is to be found in Algorithm 4.1.

For the evaluation of the reciprocal sum E_k, there exist two different approaches. The first one is the Particle Particle Particle Mesh Method (P3M) developed by Hockney and Eastwood. They have published several papers concerning this method, e.g. [23, 24, 25, 55, 56, 57]. They do not approximate the sum by using the discrete Fourier transform of a periodic version of the error function, but they optimize that to minimize the discretization error that has been introduced by meshing-up the charges.

Algorithm 4.1 Creation of the linked list arrays HOC and LL for a grid $\mathcal{G}$.

for $\mathbf{i} \in \mathcal{G}$ **do**
  $\text{HOC}(\mathbf{i}) \leftarrow 0$
end for
for $i = 1$ to $|\mathcal{P}|$ **do**
  $\mathbf{j} \leftarrow \text{round}(\mathbf{x}/h)$
  $\text{LL}(i) \leftarrow \text{HOC}(\mathbf{j})$
  $\text{HOC}(\mathbf{j}) \leftarrow i$
end for

Another approach was chosen by Essmann, Perera, Berkowitz, Darden, Lee and Pedersen, who introduced the Smooth Particle Mesh Ewald (SPME) method in [26], which is an improvement of the Particle Mesh Ewald (PME) method that was introduced by Darden, York and Pedersen in [19]. They use the unmodified Fourier transform when calculating the reciprocal sum, effectively solving the Poisson equation after smoothing with the error function. In order to compensate the discretization error, in the SPME the point charges are gridded using splines, resulting in an approximation to cardinal Euler B-splines.

All these methods use the FFT, thus the complexity is $\mathcal{O}(N + n \log n)$, where N is the number of particles and n is the number of grid points. A comparison of these mesh-based methods for the evaluation of the Ewald sum can be found in [20], further information on the P3M is contained in [21]. Although the analysis and experiments in [20] have shown, that the SPME method is not as accurate as the P3M, it has the big advantage of being able to use other solvers for the Poisson equation. So Sagui and Darden were able to use a multigrid method in a modification of the SPME presented in [70]. They also suggested to use a diffusion approach to prevent smearing in the reciprocal space. This results in an computationally optimal algorithm. Another method that uses multigrid has been published by Sutmann and Steffen in [82]. In contrast to the approach by Sagui and Darden and to the approach presented here, they use an discrete approximation to the fundamental solution to carry out the self-energy correction.

4.4 Meshed continuum method

Unlike the P3M and the SPME method, we chose a continuum approach that is not assigning the point charges to a grid. Instead we replace the point charges by charge distributions that are sampled on the mesh. As a result, unlike P3M or SPME, we do not introduce additional discretization errors. We like to note, that our approach is very similar to the one presented by Füllenbach in [38].

4.4.1 Derivation of the method

To replace the point charges, we need to choose another point symmetric density.

Definition 4.1 *Let $g : \mathbb{R}^+ \to \mathbb{R}^+$ be a function with supp$(g) = [0, r_{cut}]$, $r_{cut} > 0$ the* cut-off radius *and let $\rho_g : \mathbb{R}^3 \to \mathbb{R}^+$ be a function defined by*

$$\rho_g(\mathbf{x}) := g(|\mathbf{x}|).$$

If

$$\int_{\mathbb{R}^3} \rho_g(\mathbf{x}) d\mathbf{x} = 1,$$

then ρ_g is called a point symmetric density.

If such a point symmetric density is used as the right hand side of the Poisson equation, beyond the cut-off radius we have that the solution is equal to the solution of the Poisson equation with the δ-distribution as right-hand side.

Lemma 4.1 *Let ρ_g be a point symmetric density with cut-off radius r_{cut}. Let u and v be the solutions of the respective Poisson equations*

$$\Delta u(\mathbf{x}) = \delta(\mathbf{x}),$$
$$\Delta v(\mathbf{x}) = \rho_g(\mathbf{x}),$$

for all $\mathbf{x} \in \mathbb{R}^3$. Then for all $\mathbf{x}$ with $\|\mathbf{x}\|_2 \geq r_{cut}$ we have

$$u(\mathbf{x}) = v(\mathbf{x}).$$

Proof. We solve

$$\Delta u(\mathbf{x}) = \delta(\mathbf{x}), \text{ for all } \mathbf{x} \in \mathbb{R}$$

by convolution with the Green function, yielding

$$u(\mathbf{x}) = \int_{\mathbb{R}^3} \frac{1}{4\pi} \frac{1}{\|\mathbf{x} - \mathbf{y}\|_2} \rho_g(\mathbf{y}) d\mathbf{y}.$$

Without loss of generality we set $\mathbf{x} = (0, 0, z)^T$. Transformation to spherical coordinates yields

$$\begin{aligned}
u(\mathbf{x}) &= \frac{1}{4\pi} \int_0^{\infty} \int_0^{\pi} \int_0^{2\pi} \frac{g(r) r^2 \sin(\theta)}{\sqrt{(r\sin(\theta)\sin(\phi))^2 + (r\sin(\theta)\cos(\phi))^2 + (r\cos(\theta) - z)^2}} d\phi d\theta dr \\
&= \frac{1}{2} \int_0^{\infty} \int_0^{\pi} \frac{g(r) r^2 \sin(\theta)}{\sqrt{r^2\sin^2(\theta) + r^2\cos^2(\theta) + z^2 - 2rz\cos(\theta)}} d\theta dr \\
&= \frac{1}{2} \int_0^{\infty} \int_0^{\pi} \frac{g(r) r^2 \sin(\theta)}{\sqrt{r^2 + z^2 - 2rz\cos(\theta)}} d\theta dr \\
&= \frac{1}{2} \int_0^{\infty} \frac{g(r) r (r + z - |r - z|)}{z} dr
\end{aligned}$$

So for $z > r_{\text{cut}}$ we obtain finally

$$u(\mathbf{x}) = \frac{1}{2} \int_0^{r_{\text{cut}}} \frac{g(r) r (r + z - z + r)}{z} dr = \frac{1}{z} \int_0^{r_{\text{cut}}} g(r) r^2 dr = \frac{1}{4\pi z} \int_{\mathbb{R}^3} \phi_g(\mathbf{y}) d\mathbf{y} = \frac{1}{4\pi z}$$

□

To further simplify the representation we assume that the point symmetric density has cut-off radius $r_{\text{cut}} = 1/2$. Other radii can be obtained according to the following Lemma.

Lemma 4.2 *Let ϕ_g be a point symmetric density with cut-off radius r_{cut}. A point symmetric density with cut-off radius $\frac{1}{a} r_{cut}$ is given by*

$$\rho_{g_a}(\mathbf{x}) := a g(a \, \|\mathbf{x}\|_2).$$

The solution of the Poisson equation with this function instead of the non-scaled version is obtained in terms of the solution of the non-scaled version as

$$\Phi_{g_a}(\mathbf{x}) = a \Phi_g(a \, \mathbf{x}). \tag{4.9}$$

Proof. Obviously ρ_{g_a} is point symmetric and its cut-off radius is $\frac{1}{a} r_{\text{cut}}$. The volume is

$$\int_{\mathbb{R}^d} \rho_{g_a}(\mathbf{x}) d\mathbf{x} = 4\pi \int_0^{\frac{1}{a} r_{\text{cut}}} a^3 g(ar) r^2 dr = 4\pi \int_0^{r_{\text{cut}}} g(r) r^2 dr = 1.$$

The remaining equation (4.9) directly follows when the solution is constructed by convolution with the Green's function. □

Now, in analogy to (4.8), we define

$$\Delta\Phi_{g_a,i}(\mathbf{x}) = \rho_{g_a,i} := \frac{1}{\varepsilon_0}\sum_{\substack{j=1\\ j\neq i}}^{N} q_j \rho_{g_a}(\|\mathbf{x}-\mathbf{x}_j\|_2). \tag{4.10}$$

If ρ_g is sufficiently smooth, we can solve (4.10) numerically. Furthermore, we have

$$\Phi_i\mathbf{x} = \Phi_{g_a,i} - (\Phi_{g_a,i} - \Phi_i),$$

so if we are given $\Phi_{g_a,i}$ we can calculate Φ_i by subtracting the solution of the equation

$$\Delta(\Phi_{g_a,i} - \Phi_i) = \frac{1}{\varepsilon_0}\sum_{\substack{j=1\\ j\neq i}}^{N} q_j(\rho_{g_a} - \delta)(\|\mathbf{x}-\mathbf{x}_j\|_2).$$

As a consequence of Lemma 4.1 this can be evaluated by direct particle-particle interactions with the help of a near-field correction, as the potential induced by this right hand side only has to be evaluated in a ball of radius $\frac{1}{2a}$ around $\mathbf{x}$. The use of smooth point symmetric densities instead of the δ-distribution allows another reduction of complexity. Instead of computing $\Phi_{g_a,i}$ for each particle, we can compute

$$\Delta\Phi_{g_a,\mathcal{P}}(\mathbf{x}) = \rho_{g_a} := \frac{1}{\varepsilon_0}\sum_{j=1}^{N} q_j \rho_{g_a}(\|\mathbf{x}-\mathbf{x}_j\|_2).$$

From this we can obtain the needed $\Phi_{g_a,i}(\mathbf{x}_i)$ as

$$\Phi_{g_a,i}(\mathbf{x}_i) = \Phi_{g_a,\mathcal{P}}(\mathbf{x}_i) - q_i\Phi_{g_a}.$$

This step corresponds to the self-energy correction in P3M or SPME and allows the definition of an optimal method, as the Poisson equation only has to be solved once. So a necessary condition for defining an optimal method this way is having an optimal Poisson solver, e.g. a multigrid method. In Algorithm 4.2 we subsume the method for the calculation of the system's electrostatic energy. The calculation of the forces is carried out by numerical differentiation of the potential surface. The resulting method will be optimal in case when the number of particles in the near field, i.e. the particles that have to be treated using a particle-particle method, can be kept constant when the number of particles is growing. We can ensure this while keeping the same accuracy if only the number of particles grows, but not their mean distance, i.e. if only the system grows. As an example, we consider a system of randomly distributed charges inside of a unit cell to be

Algorithm 4.2 Calculation of the energies using the meshed continuum method. The linked list arrays HOC and LL for the grid $\mathcal{G}$ are used to speed up sampling of the point symmetric densities.

for $\mathbf{i} \in \mathcal{G}$ **do**
  for $\mathbf{j} \in \{\mathbf{j} | \|\mathbf{i} - \mathbf{j}\|_\infty \leq a/h\}$ **do**
    $k = HOC(\mathbf{j})$
    while $k \neq 0$ **do**
      $f(\mathbf{x_i}) = q_k \rho_{g_a}(\mathbf{x_i} - \mathbf{x}_k)$
      $k = LL(k)$
    end while
  end for
end for
Solve $\Delta \Phi_{g_a,\mathcal{P}} = f$ numerically using Poisson solver
$E = 0$
for $k = 1, \ldots, N$ **do**
  Approximate $\Phi_{g_a,\mathcal{P}}(\mathbf{x}_k)$ by interpolating the potential surface
  $E = E + q_k(\Phi_{g_a,\mathcal{P}}(\mathbf{x}_k) - \Phi_{g_a}(\mathbf{0}))$
end for

simulated. In order to keep the number of particles in the near-field constant, the number of grid points has to grow as the number of particles grows, while the radius of the point symmetric densities replacing the δ-distribution has to shrink reciprocally in order to keep the number of particles in the near-field constant. E.g. if the number of particles grows by a factor of b^3, the number of grid points in each dimension grows by b and the radius of the replacing charge distribution shrinks by a factor of $1/b$. As only the extent of the system is enlarged, the charge of the particles inside of the unit cell is multiplied with $1/b$ like the radius, yielding a potential as large in magnitude as the potential of the smaller system. So, for the potential of a single unit charge in the center

$$\Delta_h u(\mathbf{x}) = -4\pi \rho_{g_a}(\mathbf{x}) \Rightarrow u(\mathbf{x}) = \phi_{g_a}(\mathbf{x}) + e(\mathbf{x}),$$

we get

$$\Delta_{\frac{h}{b}} u(\mathbf{x}) = -b^3 4\pi \rho_{g_a}(b\mathbf{x}) \Rightarrow u(\mathbf{x}) = b\, \phi_{g_a}(b\mathbf{x}) + b\, e(b\mathbf{x}). \tag{4.11}$$

If the charge is multiplied with $1/b$, we see that neither the magnitude of the potential nor the magnitude of the error change, thus the method has the same accuracy and scales linearly.

4.4.2 Point symmetric densities described by B-splines

We chose point symmetric densities that can be described by B-splines. A B-spline is given by the following definition:

Definition 4.2 *A* B-spline $B_i, i = 0, 1, \ldots$ *of unit width is given by*

$$B_0(x) = \begin{cases} 1 & \textit{for } -\frac{1}{2} \leq x \leq \frac{1}{2} \\ 0 & \textit{otherwise} \end{cases},$$
$$B_{i+1}(x) = 2\, B_{\lfloor i/2 \rfloor}(2\, x) * 2\, B_{\lceil i/2 \rceil}(2\, x), \textit{ for } i = 1, 2, \ldots.$$

For example, the resulting quadratic B-spline density is given by:

$$\rho_{B_2}(r) = \begin{cases} \frac{-27r^2+36}{16} & : \ 0 \leq r < \frac{1}{6} \\ \frac{27r^2-108r+108}{32} & : \ \frac{1}{6} \leq r \leq \frac{1}{2} \\ 0 & : \ \text{otherwise} \end{cases} \tag{4.12}$$

and it induces the potential:

$$\phi_{B_2}(r) = \begin{cases} \frac{3(1296r^4-360r^2+65)}{40} & : \ 0 \leq r \leq \frac{1}{6} \\ \frac{-8505r^5+12960r^4-6480r^3+810r-2}{160r} & : \ \frac{1}{6} < r \leq \frac{1}{2} \\ \frac{1}{r} & : \ \frac{1}{2} < r \end{cases} \tag{4.13}$$

4.4.3 Numerical experiments

In the following we will present some tests of the method. First we compare the influence of the width of the replacing charge distribution while using either a standard 7-point stencil or the compact fourth-order scheme presented in Section 2.3.1. The potential surface due to a single unit charge distribution in the center of the simulation box was computed using a multigrid method an either the standard 7-point discretization of the Laplacian or the compact fourth-order discretization given by (2.21). The absolute error e between the analytical and the numerical solution was measured. In Tables 4.1 and 4.2 the results for various widths of the charge distribution are printed. Furthermore in Tables 4.3 and 4.4 the dependence of the error on the number of neighboring cells, i.e. the radius of the charge distribution divided by the grid spacing, can be found for the second-order and the fourth-order solver, respectively. In Figures 4.1 and 4.2 this dependence is shown graphically. We can see that keeping the number of neighbors constant while halving the grid spacing and doubling the grid size, the error is doubled as predicted by (4.11). Comparison of the results of the second-order solver and the fourth-order solver strongly suggests the use of high order solvers.

Next we consider a test of randomly distributed charges inside of a cube. In accordance to the considerations at the end of Section 4.4.1, the charges were scaled with the help of (4.9) such that the expected potential energy per particle for the system was constant. The results for the different steps of Algorithm 4.2 can be found in Table 4.5 and Figure 4.3. Here “sampling” denotes the process of sampling the right hand side on the grid, in the

width	$h = 1/32$		$h = 1/64$		$h = 1/128$	
	$\|\|e\|\|_\infty$	# cells	$\|\|e\|\|_\infty$	# cells	$\|\|e\|\|_\infty$	# cells
$2/32$	$6.740 \cdot 10^{0}$	$(2 \cdot 1)^3$	$7.658 \cdot 10^{-1}$	$(2 \cdot 2)^3$	$1.699 \cdot 10^{-1}$	$(2 \cdot 4)^3$
$4/32$	$3.823 \cdot 10^{-1}$	$(2 \cdot 2)^3$	$8.486 \cdot 10^{-2}$	$(2 \cdot 4)^3$	$2.049 \cdot 10^{-2}$	$(2 \cdot 8)^3$
$6/32$	$9.874 \cdot 10^{-2}$	$(2 \cdot 3)^3$	$2.378 \cdot 10^{-2}$	$(2 \cdot 6)^3$	$5.973 \cdot 10^{-3}$	$(2 \cdot 12)^3$
$8/32$	$4.232 \cdot 10^{-2}$	$(2 \cdot 4)^3$	$1.023 \cdot 10^{-2}$	$(2 \cdot 8)^3$	$2.527 \cdot 10^{-3}$	$(2 \cdot 16)^3$
$10/32$	$2.159 \cdot 10^{-2}$	$(2 \cdot 5)^3$	$5.212 \cdot 10^{-3}$	$(2 \cdot 10)^3$	$1.291 \cdot 10^{-3}$	$(2 \cdot 20)^3$
$12/32$	$1.188 \cdot 10^{-2}$	$(2 \cdot 6)^3$	$2.980 \cdot 10^{-3}$	$(2 \cdot 12)^3$	$7.444 \cdot 10^{-4}$	$(2 \cdot 24)^3$
$14/32$	$7.666 \cdot 10^{-3}$	$(2 \cdot 7)^3$	$1.885 \cdot 10^{-3}$	$(2 \cdot 14)^3$	$4.686 \cdot 10^{-4}$	$(2 \cdot 28)^3$
$16/32$	$5.090 \cdot 10^{-3}$	$(2 \cdot 8)^3$	$1.258 \cdot 10^{-3}$	$(2 \cdot 16)^3$	$3.133 \cdot 10^{-4}$	$(2 \cdot 32)^3$
$18/32$	$3.515 \cdot 10^{-3}$	$(2 \cdot 9)^3$	$8.795 \cdot 10^{-4}$	$(2 \cdot 18)^3$	$2.195 \cdot 10^{-4}$	$(2 \cdot 36)^3$
$20/32$	$2.584 \cdot 10^{-3}$	$(2 \cdot 10)^3$	$6.409 \cdot 10^{-4}$	$(2 \cdot 20)^3$	$1.598 \cdot 10^{-4}$	$(2 \cdot 40)^3$
$22/32$	$1.929 \cdot 10^{-3}$	$(2 \cdot 11)^3$	$4.804 \cdot 10^{-4}$	$(2 \cdot 22)^3$	$1.198 \cdot 10^{-4}$	$(2 \cdot 44)^3$
$24/32$	$1.474 \cdot 10^{-3}$	$(2 \cdot 12)^3$	$3.684 \cdot 10^{-4}$	$(2 \cdot 24)^3$	$9.206 \cdot 10^{-5}$	$(2 \cdot 48)^3$
$26/32$	$1.163 \cdot 10^{-3}$	$(2 \cdot 13)^3$	$2.897 \cdot 10^{-4}$	$(2 \cdot 26)^3$	$7.230 \cdot 10^{-5}$	$(2 \cdot 52)^3$
$28/32$	$9.305 \cdot 10^{-4}$	$(2 \cdot 14)^3$	$2.315 \cdot 10^{-4}$	$(2 \cdot 28)^3$	$5.780 \cdot 10^{-5}$	$(2 \cdot 56)^3$
$30/32$	$7.509 \cdot 10^{-4}$	$(2 \cdot 15)^3$	$1.879 \cdot 10^{-4}$	$(2 \cdot 30)^3$	$4.695 \cdot 10^{-5}$	$(2 \cdot 60)^3$

Table 4.1: Error of the potential of a single charge distribution for different widths and grid spacings calculated using the 7-point discretization of the Laplacian.

width	$h = 1/32$		$h = 1/64$		$h = 1/128$	
	$\|\|e\|\|_\infty$	# cells	$\|\|e\|\|_\infty$	# cells	$\|\|e\|\|_\infty$	# cells
$2/32$	$8.633 \cdot 10^{0}$	$(2 \cdot 1)^3$	$2.345 \cdot 10^{-1}$	$(2 \cdot 2)^3$	$1.204 \cdot 10^{-2}$	$(2 \cdot 4)^3$
$4/32$	$1.172 \cdot 10^{-1}$	$(2 \cdot 2)^3$	$6.012 \cdot 10^{-3}$	$(2 \cdot 4)^3$	$3.018 \cdot 10^{-4}$	$(2 \cdot 8)^3$
$6/32$	$1.318 \cdot 10^{-2}$	$(2 \cdot 3)^3$	$6.187 \cdot 10^{-4}$	$(2 \cdot 6)^3$	$3.421 \cdot 10^{-5}$	$(2 \cdot 12)^3$
$8/32$	$3.001 \cdot 10^{-3}$	$(2 \cdot 4)^3$	$1.504 \cdot 10^{-4}$	$(2 \cdot 8)^3$	$7.687 \cdot 10^{-6}$	$(2 \cdot 16)^3$
$10/32$	$8.581 \cdot 10^{-4}$	$(2 \cdot 5)^3$	$4.424 \cdot 10^{-5}$	$(2 \cdot 10)^3$	$2.436 \cdot 10^{-6}$	$(2 \cdot 20)^3$
$12/32$	$3.106 \cdot 10^{-4}$	$(2 \cdot 6)^3$	$1.711 \cdot 10^{-5}$	$(2 \cdot 12)^3$	$9.839 \cdot 10^{-7}$	$(2 \cdot 24)^3$
$14/32$	$1.456 \cdot 10^{-4}$	$(2 \cdot 7)^3$	$7.834 \cdot 10^{-6}$	$(2 \cdot 14)^3$	$4.466 \cdot 10^{-7}$	$(2 \cdot 28)^3$
$16/32$	$7.451 \cdot 10^{-5}$	$(2 \cdot 8)^3$	$3.845 \cdot 10^{-6}$	$(2 \cdot 16)^3$	$2.287 \cdot 10^{-7}$	$(2 \cdot 32)^3$
$18/32$	$3.725 \cdot 10^{-5}$	$(2 \cdot 9)^3$	$2.161 \cdot 10^{-6}$	$(2 \cdot 18)^3$	$1.229 \cdot 10^{-7}$	$(2 \cdot 36)^3$
$20/32$	$2.195 \cdot 10^{-5}$	$(2 \cdot 10)^3$	$1.212 \cdot 10^{-6}$	$(2 \cdot 20)^3$	$7.196 \cdot 10^{-8}$	$(2 \cdot 40)^3$
$22/32$	$1.356 \cdot 10^{-5}$	$(2 \cdot 11)^3$	$7.586 \cdot 10^{-7}$	$(2 \cdot 22)^3$	$4.417 \cdot 10^{-8}$	$(2 \cdot 44)^3$
$24/32$	$8.415 \cdot 10^{-6}$	$(2 \cdot 12)^3$	$4.823 \cdot 10^{-7}$	$(2 \cdot 24)^3$	$2.854 \cdot 10^{-8}$	$(2 \cdot 48)^3$
$26/32$	$5.395 \cdot 10^{-6}$	$(2 \cdot 13)^3$	$3.130 \cdot 10^{-7}$	$(2 \cdot 26)^3$	$1.883 \cdot 10^{-8}$	$(2 \cdot 52)^3$
$28/32$	$3.757 \cdot 10^{-6}$	$(2 \cdot 14)^3$	$2.137 \cdot 10^{-7}$	$(2 \cdot 28)^3$	$1.276 \cdot 10^{-8}$	$(2 \cdot 56)^3$
$30/32$	$2.581 \cdot 10^{-6}$	$(2 \cdot 15)^3$	$1.460 \cdot 10^{-7}$	$(2 \cdot 30)^3$	$8.749 \cdot 10^{-9}$	$(2 \cdot 60)^3$

Table 4.2: Error of the potential of a single charge distribution for different widths and grid spacings calculated using the compact fourth-order discretization of the Laplacian.

neighbors	$h = 1/32$	$h = 1/64$	$h = 1/128$
1	$6.740 \cdot 10^{0}$	$1.357 \cdot 10^{1}$	$2.722 \cdot 10^{1}$
2	$3.823 \cdot 10^{-1}$	$7.658 \cdot 10^{-1}$	$1.533 \cdot 10^{0}$
3	$9.874 \cdot 10^{-2}$	$1.977 \cdot 10^{-1}$	$3.957 \cdot 10^{-1}$
4	$4.232 \cdot 10^{-2}$	$8.486 \cdot 10^{-2}$	$1.699 \cdot 10^{-1}$
5	$2.159 \cdot 10^{-2}$	$4.332 \cdot 10^{-2}$	$8.676 \cdot 10^{-2}$
6	$1.188 \cdot 10^{-2}$	$2.378 \cdot 10^{-2}$	$4.756 \cdot 10^{-2}$
7	$7.666 \cdot 10^{-3}$	$1.541 \cdot 10^{-2}$	$3.087 \cdot 10^{-2}$
8	$5.090 \cdot 10^{-3}$	$1.023 \cdot 10^{-2}$	$2.049 \cdot 10^{-2}$
9	$3.515 \cdot 10^{-3}$	$7.071 \cdot 10^{-3}$	$1.415 \cdot 10^{-2}$
10	$2.584 \cdot 10^{-3}$	$5.212 \cdot 10^{-3}$	$1.044 \cdot 10^{-2}$
11	$1.929 \cdot 10^{-3}$	$3.894 \cdot 10^{-3}$	$7.802 \cdot 10^{-3}$
12	$1.474 \cdot 10^{-3}$	$2.980 \cdot 10^{-3}$	$5.973 \cdot 10^{-3}$
13	$1.163 \cdot 10^{-3}$	$2.354 \cdot 10^{-3}$	$4.720 \cdot 10^{-3}$
14	$9.305 \cdot 10^{-4}$	$1.885 \cdot 10^{-3}$	$3.784 \cdot 10^{-3}$
15	$7.509 \cdot 10^{-4}$	$1.520 \cdot 10^{-3}$	$3.051 \cdot 10^{-3}$

Table 4.3: Influence of the width of the charge distribution measured in neighboring cells in each direction for various grid-spacings for the 7-point discretization of the Laplacian.

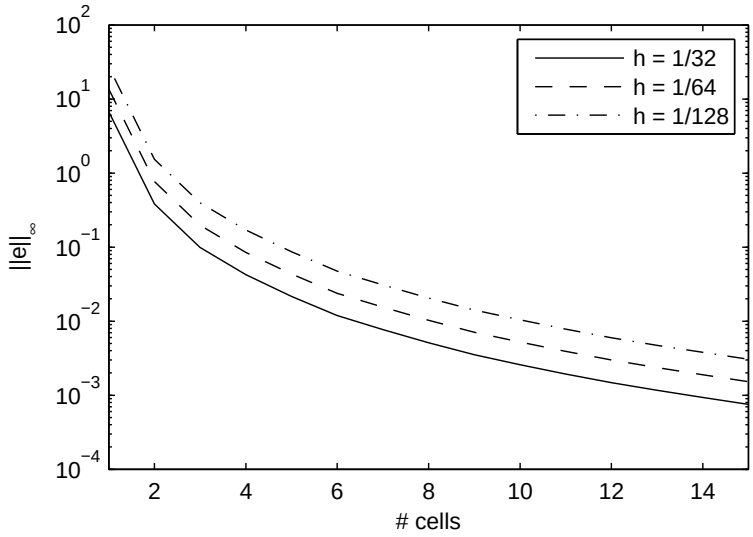

Figure 4.1: Influence of the width of the charge distribution measured in neighboring cells in each direction for various grid-spacings for the 7-point discretization of the Laplacian.

neighbors	$h = 1/32$	$h = 1/64$	$h = 1/128$
1	$8.633 \cdot 10^{0}$	$1.748 \cdot 10^{1}$	$3.518 \cdot 10^{1}$
2	$1.172 \cdot 10^{-1}$	$2.345 \cdot 10^{-1}$	$4.690 \cdot 10^{-1}$
3	$1.318 \cdot 10^{-2}$	$2.661 \cdot 10^{-2}$	$5.348 \cdot 10^{-2}$
4	$3.001 \cdot 10^{-3}$	$6.012 \cdot 10^{-3}$	$1.204 \cdot 10^{-2}$
5	$8.581 \cdot 10^{-4}$	$1.732 \cdot 10^{-3}$	$3.480 \cdot 10^{-3}$
6	$3.106 \cdot 10^{-4}$	$6.187 \cdot 10^{-4}$	$1.235 \cdot 10^{-3}$
7	$1.456 \cdot 10^{-4}$	$2.934 \cdot 10^{-4}$	$5.886 \cdot 10^{-4}$
8	$7.451 \cdot 10^{-5}$	$1.504 \cdot 10^{-4}$	$3.018 \cdot 10^{-4}$
9	$3.725 \cdot 10^{-5}$	$7.477 \cdot 10^{-5}$	$1.495 \cdot 10^{-4}$
10	$2.195 \cdot 10^{-5}$	$4.424 \cdot 10^{-5}$	$8.852 \cdot 10^{-5}$
11	$1.356 \cdot 10^{-5}$	$2.762 \cdot 10^{-5}$	$5.545 \cdot 10^{-5}$
12	$8.415 \cdot 10^{-6}$	$1.711 \cdot 10^{-5}$	$3.421 \cdot 10^{-5}$
13	$5.395 \cdot 10^{-6}$	$1.109 \cdot 10^{-5}$	$2.218 \cdot 10^{-5}$
14	$3.757 \cdot 10^{-6}$	$7.834 \cdot 10^{-6}$	$1.570 \cdot 10^{-5}$
15	$2.581 \cdot 10^{-6}$	$5.527 \cdot 10^{-6}$	$1.113 \cdot 10^{-5}$

Table 4.4: Influence of the width of the charge distribution measured in neighboring cells in each direction for various grid-spacings for the compact fourth-order discretization of the Laplacian.

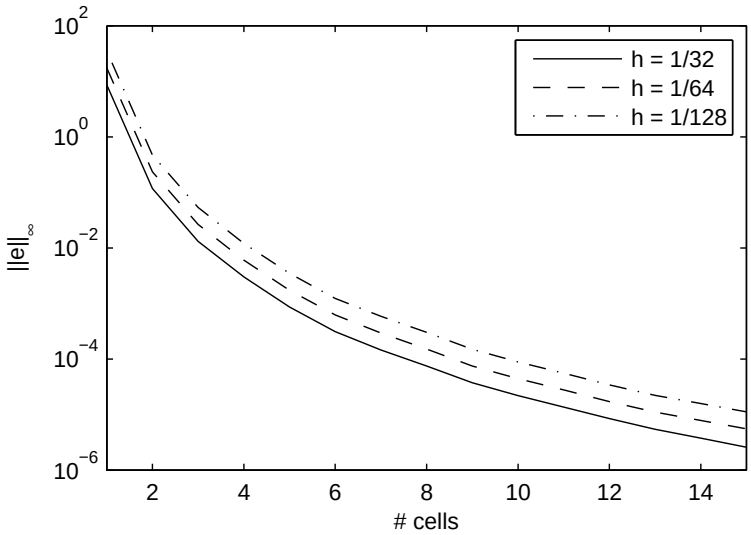

Figure 4.2: Influence of the width of the charge distribution measured in neighboring cells in each direction for various grid-spacings for the compact fourth-order discretization of the Laplacian.

# particles	grid size	$\left\lvert\frac{E_{\text{pot}}-E^*_{\text{pot}}}{E^*_{\text{pot}}}\right\rvert$	time/s sampling	solution of PDE	back interp.
1000	33^3	$1.579 \cdot 10^{-2}$	0.25	0.14	0.16
8000	65^3	$1.989 \cdot 10^{-3}$	2.01	3.46	1.41
64000	129^3	$1.033 \cdot 10^{-2}$	16.34	35.18	12.29
512000	257^3	$2.481 \cdot 10^{-3}$	132.30	340.05	108.95

Table 4.5: Scaling behavior and accuracy of Algorithm 4.2 for randomly distributed particles using the fourth-order discretization of the Laplacian and a B-spline width of 10 grid spacings.

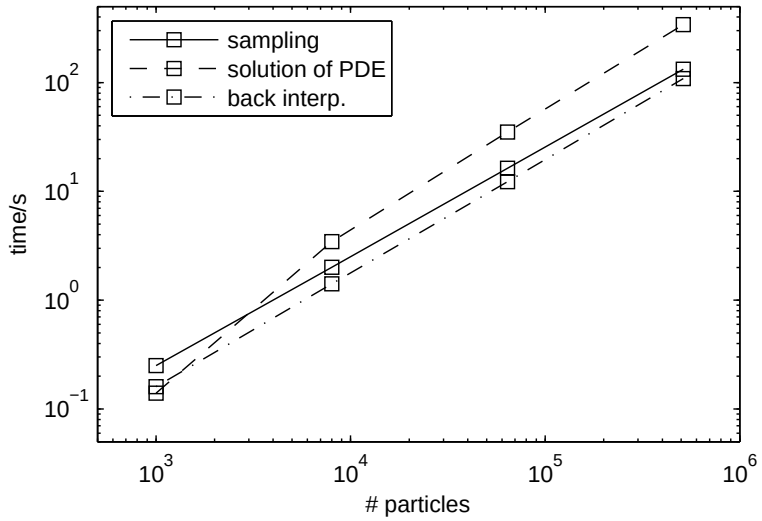

Figure 4.3: Scaling behavior of Algorithm 4.2.

column titled “solution of PDE” the times for the multigrid solver can be found and finally the times measured for the back interpolation of the potential to the particle positions is shown in the outermost right column. We see that the proposed method scales linearly with the number of particles while keeping about the same accuracy. Fluctuations in the accuracy are due to the random distributions of the charges inside the simulation box.
The last test we want to present is the calculation of the total electrostatic energy of a DNA fragment including counter ions consisting of 1316 atoms. This test was performed in order to show that the method provides a way to accurately calculate the electrostatic energy of a real molecule. The relative error of the total electrostatic energy is found in Table 4.6.

neighbors	$h = 1/32$	$h = 1/64$	$h = 1/128$
1	$1.656 \cdot 10^{-1}$	$1.007 \cdot 10^{0}$	$1.701 \cdot 10^{0}$
2	$7.370 \cdot 10^{-3}$	$7.935 \cdot 10^{-2}$	$1.595 \cdot 10^{-1}$
3	$7.658 \cdot 10^{-4}$	$4.963 \cdot 10^{-3}$	$2.218 \cdot 10^{-2}$
4	$1.104 \cdot 10^{-4}$	$4.584 \cdot 10^{-4}$	$3.879 \cdot 10^{-3}$
5	$2.985 \cdot 10^{-5}$	$1.436 \cdot 10^{-4}$	$6.941 \cdot 10^{-4}$
6	$9.356 \cdot 10^{-6}$	$5.147 \cdot 10^{-5}$	$1.356 \cdot 10^{-4}$
7	$3.309 \cdot 10^{-6}$	$1.578 \cdot 10^{-5}$	$5.151 \cdot 10^{-5}$
8	$1.078 \cdot 10^{-6}$	$4.660 \cdot 10^{-6}$	$2.568 \cdot 10^{-5}$
9	$1.733 \cdot 10^{-7}$	$2.996 \cdot 10^{-6}$	$1.718 \cdot 10^{-5}$
10	$1.356 \cdot 10^{-7}$	$1.185 \cdot 10^{-6}$	$7.477 \cdot 10^{-6}$
11	$6.242 \cdot 10^{-7}$	$5.755 \cdot 10^{-7}$	$5.208 \cdot 10^{-6}$
12	$2.506 \cdot 10^{-8}$	$1.637 \cdot 10^{-7}$	$2.472 \cdot 10^{-6}$
13	$2.075 \cdot 10^{-8}$	$5.355 \cdot 10^{-8}$	$1.542 \cdot 10^{-6}$
14	$1.132 \cdot 10^{-8}$	$2.995 \cdot 10^{-8}$	$9.037 \cdot 10^{-7}$
15	$2.503 \cdot 10^{-10}$	$0.152 \cdot 10^{-8}$	$5.209 \cdot 10^{-7}$

Table 4.6: Relative error of the electrostatic energy of a DNA fragment calculated for various grid spacings using the compact fourth-order discretization of the Laplacian.

Chapter 5

Conclusion

In this work we presented a framework for the application of multigrid methods as a solver for the Poisson equation that arises in particle simulation methods. As the Poisson equation's Green's function is equal to the Coulomb potential and gravitational potential up to a constant factor, the use of multigrid methods is possible for a wide range of applications, i.e. in molecular dynamics simulations and in the simulation of astrophysical phenomena. We reformulated the problem in a consistent way, such that the problem is equivalent to the solution of a partial differential equation with a special right hand side. Additionally, a near field correction has to be applied. Given that the continuous partial differential equation is solved analytically, no errors are introduced by this reformulation. When solved numerically, the only errors introduced are the discretization error of the numerical scheme used to solve the PDE and the error of the back-interpolation scheme.

For the solution of PDEs in open systems we introduced the hierarchical grid refinement technique by Washio and Oosterlee [87] and a new modification of this technique which is guaranteed to yield a result of the desired accuracy. We were able to show that the modified method still scales optimally in terms of unknowns, although new grid points are introduced. For the solution of the resulting method a geometric multigrid method using the FAC method is appropriate.

In the periodic case the problem of solving the Poisson equation with constant coefficients on an equispaced grid yields a linear system with circulant coefficient matrix. We reviewed the algebraic multigrid theory for hermitian positive matrices in general and its use in the circulant case. Motivated by the possible computational savings, we analyzed the theory and developed sufficient conditions for a replacement coarse grid operator instead of the Galerkin operator. The derived conditions were verified for schemes that are applicable to certain circulant matrices.

Although multigrid methods are fast methods, it can still be desirable to parallelize even fast methods. Therefore we presented a parallel implementation of the solver for circulant matrices, which included the Galerkin operator as well as its replacement. The results

were obtained on up to 65536 processors on Jülich Supercomputing Centre's Blue Gene/P system JUGENE and on the Blue Gene/L system JUBL. The method shows very good scaling results, allowing very large systems to be solved in fractions of a second.

With this work a new method using multigrid for the solution of the long-ranged Coulomb potential or gravitational potential becomes available for the simulation of systems consisting of atomic particles.

The obtained theoretical results for multigrid methods pay off in this application. In the future we will extend the theory to cover other classes of matrices to be able to replace the Galerkin operator there, as well.

Acknowledgments

This work would not have been possible without the support of a lot of people. First of all I like to thank my wife Beate, who always understood me and encouraged me to continue to work in a field that really fascinates me and makes me curious. I like to thank our daughter Emma for being cute and sympathetic when I had to work.

Next I like to thank my advisor Prof. Andreas Frommer a lot. It was a pleasure to work with him and I always had a lot of fun – and gained additional insight – travelling to Wuppertal.

Dr. Godehard Sutmann was a great help to me. I enjoyed our fruitful discussions and I am grateful for his introduction to particle simulation in general and his patient answers to all of my questions.

Besides Godehard Sutmann I like to thank many other former and current colleagues at the Jülich Supercomputing Centre: Prof. Dr. Dr. Thomas Lippert, who gave me the possibility to work on this thesis at his institute and who acted as a second examiner, Dr. Rüdiger Esser, who was a great department manager, and the other Ph.D. students at the FZJ, i.e. Daniel Becker, Ivo Kabadshow, Stefan Krieg, Tom Schröder, Robert Speck, Dr. Tatjana Streit, and Binh Trieu. I also like to mention the group members of the complex atomistic modelling and simulation group at the Jülich Supercomputing Centre and the great technical and administrative staff at the institute, there are just too many to mention all.

My thanks go to James Brannick who acted as another second examiner.

I also like to thank Rob Falgout and the whole Scalable Linear Solvers team at Lawrence Livermore National Laboratory for the nice atmosphere during my stay at Livermore and for a many lessons learned in parallel and algebraic multigrid.

Bibliography

[1] A. ARICÒ, *Fast algorithms for some structured linear algebra problems*, PhD thesis, Uiversità degli studi di pavia, Pavia, 2005.

[2] A. ARICÒ AND M. DONATELLI, *A V-cycle multigrid for multilevel matrix algebras: proof of optimality*, Numer. Math., 105 (2007), pp. 511–547.

[3] A. ARICÒ, M. DONATELLI, AND S. SERRA-CAPIZZANO, *V-cycle optimal convergence for certain (multilevel) structured linear systems*, SIAM J. Matrix Anal. Appl., 26 (2004), pp. 186–214.

[4] S. F. ASHBY AND R. D. FALGOUT, *A parallel multigrid preconditioned conjugate gradient algorithm for groundwater flow simulations*, Nucl. Sci. Eng., 124 (1996), pp. 145–159.

[5] A. H. BAKER, R. D. FALGOUT, AND U. M. YANG, *An assumed partition algorithm for determining processor inter-communication*, Parallel Computing, 32 (2006), pp. 394–414.

[6] N. S. BAKHVALOV, *On the convergence of a relaxation method with natural constraints on the elliptic operator*, USSR Comp. Math. Math. Phys., 6 (1966), pp. 101–135.

[7] J. E. BARNES AND P. HUT, *A hierarchical O(N log N) force calculation algorithm*, Nature, 324 (1986), pp. 446–449.

[8] M. BOLTEN, *Hierarchical grid coarsening for the solution of the Poisson equation in free space*, Electron. Trans. Numer. Anal., 29 (2008), pp. 70–80.

[9] M. BOLTEN, A. DOLFEN, N. EICKER, I. GUTHEIL, W. HOMBERG, E. KOCH, A. SCHILLER, G. SUTMANN, AND L. YANG, *Juice - Jülich Intitiative Cell Cluster - Report 2007*, Tech. Rep. FZJ-JSC-IB-2007-13, Jülich Supercomputing Centre, Research Centre Jülich, Jülich, December 2007.

[10] D. BRAESS AND W. HACKBUSCH, *A new convergence proof for the multigrid method including the V-cycle*, SIAM J. Numer. Anal., 20 (1983), pp. 967–975.

[11] A. BRANDT, *Multi-level adaptive technique (MLAT) for fast numerical solution to boundary value problems*, in Proceedings of the Third International Conference on

Numerical Methods in Fluid Mechanics, H. Cabammes and R. Temam, eds., no. 18 in Lecture Notes in Physics, Berlin, Heidelberg, New York, July 1973, Springer-Verlag, pp. 82–89.

[12] ——, *Multi-level adaptive solutions to boundary-value problems*, Math. Comp., 31 (1977), pp. 333–390.

[13] ——, *Multi-level adaptive technique (MLAT) for partial differential equations: Ideas and software*, in Proceedings of a Symposium Conducted by the Mathematics Research Center, J. R. Rice, ed., vol. 3 of Mathematical Software, New York, San Francisco, London, March 1977, The University of Wisconsin–Madison, Academic Press, pp. 277–318.

[14] ——, *Algebraic multigrid theory: The symmetric case*, Appl. Math. Comput., 19 (1986), pp. 23–56.

[15] W. L. BRIGGS, V. E. HENSON, AND S. F. MCCORMICK, *A Multigrid Tutorial*, SIAM, Philadelphia, 2000.

[16] R. H. BURKHART, *Asymptotic expansion of the free-space green's function for the discrete 3-D Poisson equation*, SIAM J. Sci. Comput., 18 (1997), pp. 1142–1162.

[17] R. H. CHAN, Q.-S. CHANG, AND H.-W. SUN, *Multigrid method for ill-conditioned symmetric Toeplitz systems*, SIAM J. Sci. Comput., 19 (1998), pp. 516–529.

[18] E. CHOW, R. D. FALGOUT, J. J. HU, R. S. TUMINARO, AND U. M. YANG, *A survey of parallelization techniques for multigrid solvers*, in Parallel Processing for Scientific Computing, M. A. Heroux, P. Raghavan, and H. D. Simon, eds., SIAM Series on Software, Environments, and Tools, SIAM, Philadelphia, 2006, ch. 10.

[19] T. DARDEN, D. YORK, AND L. PEDERSEN, *Particle mesh ewald: An $N\log(N)$ method for ewald sums in large systems*, J. Chem. Phys., 98 (1993), pp. 10089–10092.

[20] M. DESERNO AND C. HOLM, *How to mesh up Ewald sums (I): A theoretical and numerical comparison of various particle mesh routines*, J. Chem. Phys., 109 (1998), pp. 7678–7693.

[21] ——, *How to mesh up Ewald sums (II): An accurate error estimate for the P3M algorithm*, J. Chem. Phys., 109 (1998), pp. 7694–7701.

[22] M. DONATELLI, *Image deconvolution and multigrid methods*, PhD thesis, Uiversità degli studi di milano, Milano, 2005.

[23] J. EASTWOOD, R. HOCKNEY, AND D. LAWRENCE, *PM3DP – the three dimensional periodic particle-particle/particle-mesh program*, Comp. Phys. Comm., 19 (1977), pp. 215–261.

[24] J. W. EASTWOOD, *Optimal particle mesh algorithms*, J. Comput. Phys., 18 (1975), pp. 1–20.

[25] ——, *Optimal P^3M algorithms for molecular dynamics simulations*, in Computational Methods in Classical and Quantum Physics, M. B. Hooper, ed., London, 1976, Advance Publications Ltd, pp. 206–228.

[26] U. ESSMANN, L. PERERA, M. L. BERKOWITZ, T. DARDEN, H. LEE, AND L. G. PEDERSEN, *A smooth particle mesh ewald method*, J. Chem. Phys., 103 (1995), pp. 8577–8593.

[27] P. EWALD, *Die Berechnung optischer und elektrostatischer Gitterpotentiale*, Ann. Phys., 64 (1921), p. 253.

[28] R. D. FALGOUT, J. E. JONES, AND U. M. YANG, *The design and implementation of hypre, a library of parallel high performance preconditioners*, in Numerical Solution of Partial Differential Equations on Parallel Computers, A. M. Bruaset and A. Tveito, eds., vol. 51 of Lecture Notes in Computational Science and Engineering, Springer-Verlag, Berlin, Heidelberg, 2006, ch. 8, pp. 267–294.

[29] R. D. FALGOUT AND U. M. YANG, *hypre: a library of high performance preconditioners*, in Computational Science - ICCS 2002 Part III, P. M. A. Sloot, C. J. K. Tan., and J. J. Dongarra, eds., vol. 2331 of Lecture Notes in Computer Science, Berlin, Heidelberg, 2002, Springer-Verlag, pp. 632–641.

[30] R. P. FEDORENKO, *A relaxation method for solving elliptic difference equations*, USSR Comp. Math. Math. Phys., 1 (1962), pp. 1092–1096.

[31] ——, *The speed of convergence of one iterative process*, USSR Comp. Math. Math. Phys., 4 (1964), pp. 227–235.

[32] G. FIORENTINO AND S. SERRA, *Multigrid methods for Toeplitz matrices*, Calcolo, 28 (1991), pp. 238–305.

[33] ——, *Multigrid methods for indefinite Toeplitz matrices*, Calcolo, 33 (1996), pp. 223–236.

[34] G. FIORENTINO AND S. SERRA, *Multigrid methods for symmetric positive definite block Toeplitz matrices with nonnegative generating functions*, SIAM Journal on Scientific Computing, 17 (1996), pp. 1068–1081.

[35] R. FISCHER, *Multigrid methods for anisotropic and indefinite structured linear systems of equations*, PhD thesis, Technische Universität München, 2006.

[36] A. FRIEDMAN, *Partial differential equations*, Holt, Rinehart and Winston, New York, Chicago, San Francisco, Atlanta, Dallas, Montreal, Toronto, London, Sydney, 1969.

[37] A. FROMMER, *Lösung linearer Gleichungssysteme auf Parallelrechnern*, Vieweg, Braunschweig, 1990.

[38] T. FÜLLENBACH, *Mehrgitterverfahren für die zwei- und dreidimensionale Poissongleichung mit periodischen Randbedingungen und eine Anwendung in der Molekulardynamik*, Tech. Rep. 11/2000, GMD, St. Augustin, 2000.

[39] A. GARA, M. A. BLUMRICH, D. CHEN, G. L.-T. CHIU, P. COTEUS, M. E. GIAMPAPA, R. A. HARING, P. HEIDELBERGER, D. HOENICKE, G. V. KOPCSAY, T. A. LIEBSCH, M. OHMACHT, B. D. STEINMACHER-BUROW, T. TAKKEN, AND P. VRANAS, *Overview of the Blue Gene/L system architecture*, IBM J. Res. & Dev., 49 (2005), pp. 195–212.

[40] P. GIBBON AND G. SUTMANN, *Long range interactions in many-particle simulation*, in Quantum simulations of many-body systems: from theory to algorithms, J. Grotendorst, D. Marx, and A. Muramatsu, eds., vol. 10 of NIC series, Jülich, 2001, John von Neumann Institute for Computing, pp. 467–506.

[41] D. GILBARG AND N. S. TRUDINGER, *Elliptic partial differential equations of second order*, vol. 224 of A series of comprehensive studies in mathematics, Springer-Verlag, Berlin, Heidelberg, New York, 1970.

[42] G. GOLUB AND J. ORTEGA, *Scientific Computing an introduction with parallel computing*, Academic Press, San Diego, 1993.

[43] E. A. GONZÁLEZ-VELASCO, *Fourier analysis and boundary value problems*, Academic Press, San Diego, 1995.

[44] R. GRAY, *Toeplitz and circulant matrices: A review*, Tech. Rep. 6504-1, Stanford University, Stanford, CA, 1977.

[45] L. GREENGARD AND V. ROKHLIN, *A fast algorithm for particle simulations*, J. Comp. Phys., 73 (1987), pp. 325–348.

[46] M. GRIEBEL, S. KNAPEK, G. ZUMBUSCH, AND A. CAGLAR, *Numerische Simulation in der Moleküldynamik*, Springer-Verlag, Berlin, Heidelberg, New York, 2004.

[47] K. E. GUSTAFSON, *Introduction to partial differential equations and Hilbert space methods*, Dover, Mineola, 1999.

[48] W. HACKBUSCH, *Ein iteratives Verfahren zur schnellen Auflösung elliptischer Randwertprobleme*, Rep. 76-12, Institute for Applied Mathematics, University of Cologne, West Germany, Cologne, 1976.

[49] ——, *On the convergence of a multi-grid iteration applied to finite element equations*, Rep. 77-8, Institute for Applied Mathematics, University of Cologne, West Germany, Cologne, 1977.

[50] ——, *Convergence of multi-grid iterations applied to difference equations*, Math. Comp., 34 (1980), pp. 425–440.

[51] ——, *On the convergence of multi-grid iterations*, Beiträge Numer. Math., 9 (1981), pp. 213–239.

[52] ——, *Multi-grid convergence theory*, in Multigrid methods, W. Hackbusch and U. Trottenberg, eds., vol. 960 of Lecture Notes in Mathematics, Berlin, 1982, Springer-Verlag, pp. 177–219.

[53] ——, *Multi-Grid Methods and Applications*, Springer-Verlag, Berlin, 1985.

[54] ——, *Iterative solution of large sparse systems of equations*, no. 95 in Applied Mathematical Sciences, Springer-Verlag, New York, 1994.

[55] R. W. HOCKNEY, *The potential calculation and some applications*, in Methods in Computational Physics: Plasma Physics, B. Alder, S. Fernbach, and M. Rotenberg, eds., vol. 9 of Methods in Computational Physics, Academic Press, New York, 1970, pp. 136–211.

[56] R. W. HOCKNEY AND J. W. EASTWOOD, *Computer simulation using particles*, Institute of Physics, Bristol, 1988.

[57] R. W. HOCKNEY, S. P. GOEL, AND J. W. EASTWOOD, *A 10000 particle molecular dynamics model with long rang forces*, Chem. Phys. Lett., 21 (1973), pp. 589–591.

[58] J. JOST, *Partial Differential Equations*, Springer-Verlag, Berlin, Heidelberg, New York, 2002.

[59] T. W. KÖRNER, *Fourier analysis*, Cambridge University Press, Cambridge, 1988.

[60] K. N. KUDIN AND G. E. SCUSERIA, *Revisiting infinite lattice sums with the periodic fast multipole method*, J. Chem. Phys., 121 (2004), pp. 2886–2890.

[61] S. LARSSON AND V. THOMÉE, *Partial Differential Equations with Numerical Methods*, Springer-Verlag, Berlin, Heidelberg, New York, 2005.

[62] P. D. LAX AND A. N. MILGRAM, *Parabolic equations*, Ann. of Math., 33 (1954).

[63] J. MANDEL, *Algebraic study of multigrid methods for symmetric, definite problems*, Appl. Math. Comput., 25 (1988), pp. 39–56.

[64] S. F. MCCORMICK, *Multigrid methods for variational problems: General theory for the v-cycle*, SIAM J. Numer. Anal., 22 (1985), pp. 634–643.

[65] ——, *Multilevel adaptive methods for partial differential equations*, vol. 6 of Frontiers Appl. Math., SIAM, Philadelphia, 1989.

[66] S. F. MCCORMICK AND J. THOMAS, *The fast adaptive composite grid (FAC) method for elliptic equations*, Math. Comp., 46 (1986), pp. 439–456.

[67] A. MEISTER, *Numerik linearer Gleichungssysteme*, Vieweg, Braunschweig/Wiesbaden, 1999.

[68] U. RÜDE, *Mathematical and computational techniques for multilevel adaptive techniques*, vol. 13 of Frontiers Appl. Math., SIAM, Philadelphia, 1993.

[69] J. W. RUGE AND K. STÜBEN, *Algebraic multigrid*, in Multigrid methods, S. F. McCormick, ed., vol. 3 of Frontiers Appl. Math., SIAM, Philadelphia, 1987, pp. 73–130.

[70] C. SAGUI AND T. DARDEN, *Multigrid methods for classical molecular dynamics simulations of biomolecules*, J. Chem. Phys., 114 (2001), pp. 6578–6591.

[71] U. SCHMIDT, *FZJ-ZAM IBM Blue Gene/L - JUBL home page*. Website, November 2007. Available online at `http://www.fz-juelich.de/jsc/ibm-bgl` visited on March 3rd 2008.

[72] ——, *FZJ-JSC IBM Blue Gene/P - JUGENE home page*. Website, February 2008. Available online at `http://www.fz-juelich.de/jsc/jugene` visited on March 3rd 2008.

[73] S. SERRA-CAPIZZANO AND C. TABLINO-POSSIO, *Preliminary remarks on multigrid methods for circulant matrices*, in Numerical Analysis and its Applications, Second International Conference, NAA 2000, Rousse, Bulgaria, June 11–15, 2000, Revised, L. V. amd J. Waśnievski and P. Y. Yalamov, eds., vol. 1988 of Lecture Notes in Computer Science, Berlin, 1988, Springer-Verlag, pp. 152–159.

[74] ——, *Multigrid methods for multilevel circulant matrices*, SIAM J. Sci. Comput., 26 (2004), pp. 55–85.

[75] W. SPOTZ AND G. CAREY, *A high-order compact formulation for the 3D Poisson equation*, Numer. Methods Partial Differential Equations, 12 (1996), pp. 235–243.

[76] K. STÜBEN, *Algebraic multigrid: An introduction with applications*, GMD Report 70, GMD - Forschungszentrum Informationstechnik GmbH, St. Augustin, November 1999.

[77] ——, *A review of algebraic multigrid*, GMD Report 69, GMD - Forschungszentrum Informationstechnik GmbH, St. Augustin, November 1999.

[78] ——, *An introduction to algebraic multigrid*, in Multigrid, U. Trottenberg, C. Oosterlee, and A. Schüller, eds., Academic Press, 2001, ch. Appendix A, pp. 413–532.

[79] H.-W. SUN, R. H. CHAN, AND Q.-S. CHANG, *A note on the convergence of the two-grid method for Toeplitz systems*, Computers Math. Applic., 34 (1997), pp. 11–18.

[80] H.-W. SUN, X.-Q. JIN, AND Q.-S. CHANG, *Convergence of the multigrid method for ill-conditioned block Toeplitz systems*, BIT, 41 (2001), pp. 179–190.

[81] G. SUTMANN, *Classical molecular dynamics*, in Quantum simulations of many-body systems: from theory to algorithms, J. Grotendorst, D. Marx, and A. Muramatsu, eds., vol. 10 of NIC series, Jülich, 2001, John von Neumann Institute for Computing, pp. 211–254.

[82] G. SUTMANN AND B. STEFFEN, *A particle-particle particle-multigrid algorithm for long range interactions in molecular systems*, Comp. Phys. Comm., 169 (2005), pp. 343–346.

[83] ——, *High-order compact solvers for the three dimensional Poisson equation*, J. Comp. Appl. Math., 187 (2006), pp. 142–170.

[84] U. TROTTENBERG, C. OOSTERLEE, AND A. SCHÜLLER, *Multigrid*, Academic Press, San Diego, 2001.

[85] R. S. TUMINARO, *Multigrid algorithms on parallel processing systems*, PhD thesis, Stanford University, Stanford, December 1989.

[86] E. E. TYRTYSHNIKOV, *Circulant preconditioners with unbounded inverses*, Linear Algebra Appl., 216 (1995), pp. 1–23.

[87] T. WASHIO AND C. W. OOSTERLEE, *Error analysis for a potential problem on locally refined grids*, Numer. Math., 86 (2000), pp. 539–563.

[88] S. WILLIAMS, J. SHALF, L. OLIKER, P. HUSBANDS, S. KAMIL, AND K. YELICK, *The potential of the Cell processor for scientific computing*, Tech. Rep. LBNL-59071, Lawrence Berkeley National Laboratory, Berkeley, October 2005.

[89] K. YOSIDA, *Functional analysis*, Springer-Verlag, Berlin, Heidelberg, New York, 1971.

Already published:

Modern Methods and Algorithms of Quantum Chemistry - Proceedings
Johannes Grotendorst (Editor)
Winter School, 21 - 25 February 2000, Forschungszentrum Jülich
NIC Series Volume 1
ISBN 3-00-005618-1, February 2000, 562 pages
out of print

Modern Methods and Algorithms of Quantum Chemistry - Poster Presentations
Johannes Grotendorst (Editor)
Winter School, 21 - 25 February 2000, Forschungszentrum Jülich
NIC Series Volume 2
ISBN 3-00-005746-3, February 2000, 77 pages
out of print

Modern Methods and Algorithms of Quantum Chemistry - Proceedings, Second Edition
Johannes Grotendorst (Editor)
Winter School, 21 - 25 February 2000, Forschungszentrum Jülich
NIC Series Volume 3
ISBN 3-00-005834-6, December 2000, 638 pages
out of print

Nichtlineare Analyse raum-zeitlicher Aspekte der hirnelektrischen Aktivität von Epilepsiepatienten
Jochen Arnold
NIC Series Volume 4
ISBN 3-00-006221-1, September 2000, 120 pages

Elektron-Elektron-Wechselwirkung in Halbleitern: Von hochkorrelierten kohärenten Anfangszuständen zu inkohärentem Transport
Reinhold Lövenich
NIC Series Volume 5
ISBN 3-00-006329-3, August 2000, 146 pages

Erkennung von Nichtlinearitäten und wechselseitigen Abhängigkeiten in Zeitreihen
Andreas Schmitz
NIC Series Volume 6
ISBN 3-00-007871-1, May 2001, 142 pages

Multiparadigm Programming with Object-Oriented Languages - Proceedings
Kei Davis, Yannis Smaragdakis, Jörg Striegnitz (Editors)
Workshop MPOOL, 18 May 2001, Budapest
NIC Series Volume 7
ISBN 3-00-007968-8, June 2001, 160 pages

Europhysics Conference on Computational Physics - Book of Abstracts
Friedel Hossfeld, Kurt Binder (Editors)
Conference, 5 - 8 September 2001, Aachen
NIC Series Volume 8
ISBN 3-00-008236-0, September 2001, 500 pages

NIC Symposium 2001 - Proceedings
Horst Rollnik, Dietrich Wolf (Editors)
Symposium, 5 - 6 December 2001, Forschungszentrum Jülich
NIC Series Volume 9
ISBN 3-00-009055-X, May 2002, 514 pages

Quantum Simulations of Complex Many-Body Systems: From Theory to Algorithms - Lecture Notes
Johannes Grotendorst, Dominik Marx, Alejandro Muramatsu (Editors)
Winter School, 25 February - 1 March 2002, Rolduc Conference Centre, Kerkrade, The Netherlands
NIC Series Volume 10
ISBN 3-00-009057-6, February 2002, 548 pages

Quantum Simulations of Complex Many-Body Systems: From Theory to Algorithms- Poster Presentations
Johannes Grotendorst, Dominik Marx, Alejandro Muramatsu (Editors)
Winter School, 25 February - 1 March 2002, Rolduc Conference Centre, Kerkrade, The Netherlands
NIC Series Volume 11
ISBN 3-00-009058-4, February 2002, 194 pages

Strongly Disordered Quantum Spin Systems in Low Dimensions: Numerical Study of Spin Chains, Spin Ladders and Two-Dimensional Systems
Yu-cheng Lin
NIC Series Volume 12
ISBN 3-00-009056-8, May 2002, 146 pages

Multiparadigm Programming with Object-Oriented Languages - Proceedings
Jörg Striegnitz, Kei Davis, Yannis Smaragdakis (Editors)
Workshop MPOOL 2002, 11 June 2002, Malaga
NIC Series Volume 13
ISBN 3-00-009099-1, June 2002, 132 pages

Quantum Simulations of Complex Many-Body Systems: From Theory to Algorithms - Audio-Visual Lecture Notes
Johannes Grotendorst, Dominik Marx, Alejandro Muramatsu (Editors)
Winter School, 25 February - 1 March 2002, Rolduc Conference Centre, Kerkrade, The Netherlands
NIC Series Volume 14
ISBN 3-00-010000-8, November 2002, DVD

Numerical Methods for Limit and Shakedown Analysis
Manfred Staat, Michael Heitzer (Eds.)
NIC Series Volume 15
ISBN 3-00-010001-6, February 2003, 306 pages

Design and Evaluation of a Bandwidth Broker that Provides Network Quality of Service for Grid Applications
Volker Sander
NIC Series Volume 16
ISBN 3-00-010002-4, February 2003, 208 pages

Automatic Performance Analysis on Parallel Computers with SMP Nodes
Felix Wolf
NIC Series Volume 17
ISBN 3-00-010003-2, February 2003, 168 pages

Haptisches Rendern zum Einpassen von hochaufgelösten Molekülstrukturdaten in niedrigaufgelöste Elektronenmikroskopie-Dichteverteilungen
Stefan Birmanns
NIC Series Volume 18
ISBN 3-00-010004-0, September 2003, 178 pages

Auswirkungen der Virtualisierung auf den IT-Betrieb
Wolfgang Gürich (Editor)
GI Conference, 4 - 5 November 2003, Forschungszentrum Jülich
NIC Series Volume 19
ISBN 3-00-009100-9, October 2003, 126 pages

NIC Symposium 2004
Dietrich Wolf, Gernot Münster, Manfred Kremer (Editors)
Symposium, 17 - 18 February 2004, Forschungszentrum Jülich
NIC Series Volume 20
ISBN 3-00-012372-5, February 2004, 482 pages

Measuring Synchronization in Model Systems and Electroencephalographic Time Series from Epilepsy Patients
Thomas Kreutz
NIC Series Volume 21
ISBN 3-00-012373-3, February 2004, 138 pages

Computational Soft Matter: From Synthetic Polymers to Proteins - Poster Abstracts
Norbert Attig, Kurt Binder, Helmut Grubmüller, Kurt Kremer (Editors)
Winter School, 29 February - 6 March 2004, Gustav-Stresemann-Institut Bonn
NIC Series Volume 22
ISBN 3-00-012374-1, February 2004, 120 pages

Computational Soft Matter: From Synthetic Polymers to Proteins - Lecture Notes
Norbert Attig, Kurt Binder, Helmut Grubmüller, Kurt Kremer (Editors)
Winter School, 29 February - 6 March 2004, Gustav-Stresemann-Institut Bonn
NIC Series Volume 23
ISBN 3-00-012641-4, February 2004, 440 pages

Synchronization and Interdependence Measures and their Applications to the Electroencephalogram of Epilepsy Patients and Clustering of Data
Alexander Kraskov
NIC Series Volume 24
ISBN 3-00-013619-3, May 2004, 106 pages

High Performance Computing in Chemistry
Johannes Grotendorst (Editor)
Report of the Joint Research Project:
High Performance Computing in Chemistry - HPC-Chem
NIC Series Volume 25
ISBN 3-00-013618-5, December 2004, 160 pages

Zerlegung von Signalen in unabhängige Komponenten: Ein informationstheoretischer Zugang
Harald Stögbauer
NIC Series Volume 26
ISBN 3-00-013620-7, April 2005, 110 pages

Multiparadigm Programming 2003
Joint Proceedings of the
3rd International Workshop on Multiparadigm Programming with Object-Oriented Languages (MPOOL'03)
and the
1st International Workshop on Declarative Programming in the Context of Object-Oriented Languages (PD-COOL'03)
Jörg Striegnitz, Kei Davis (Editors)
NIC Series Volume 27
ISBN 3-00-016005-1, July 2005, 300 pages

Integration von Programmiersprachen durch strukturelle Typanalyse und partielle Auswertung
Jörg Striegnitz
NIC Series Volume 28
ISBN 3-00-016006-X, May 2005, 306 pages

OpenMolGRID - Open Computing Grid for Molecular Science and Engineering
Final Report
Mathilde Romberg (Editor)
NIC Series Volume 29
ISBN 3-00-016007-8, July 2005, 86 pages

GALA Grünenthal Applied Life Science Analysis
Achim Kless and Johannes Grotendorst (Editors)
NIC Series Volume 30
ISBN 3-00-017349-8, November 2006, 204 pages

Computational Nanoscience: Do It Yourself!
Lecture Notes
Johannes Grotendorst, Stefan Blügel, Dominik Marx (Editors)
Winter School, 14. - 22 February 2006, Forschungszentrum Jülich
NIC Series Volume 31
ISBN 3-00-017350-1, February 2006, 528 pages

NIC Symposium 2006 - Proceedings
G. Münster, D. Wolf, M. Kremer (Editors)
Symposium, 1 - 2 March 2006, Forschungszentrum Jülich
NIC Series Volume 32
ISBN 3-00-017351-X, February 2006, 384 pages

Parallel Computing: Current & Future Issues of High-End Computing
Proceedings of the International Conference ParCo 2005
G.R. Joubert, W.E. Nagel, F.J. Peters,
O. Plata, P. Tirado, E. Zapata (Editors)
NIC Series Volume 33
ISBN 3-00-017352-8, October 2006, 930 pages

From Computational Biophysics to Systems Biology 2006 Proceedings
U.H.E. Hansmann, J. Meinke, S. Mohanty, O. Zimmermann (Editors)
NIC Series Volume 34
ISBN-10 3-9810843-0-6, ISBN-13 978-3-9810843-0-6,
September 2006, 224 pages

Dreistufig parallele Software zur Parameteroptimierung von Support-Vektor-Maschinen mit kostensensitiven Gütemaßen
Tatjana Eitrich
NIC Series Volume 35
ISBN 978-3-9810843-1-3, March 2007, 262 pages

From Computational Biophysics to Systems Biology (CBSB07)
Proceedings
U.H.E. Hansmann, J. Meinke, S. Mohanty, O. Zimmermann (Editors)
NIC Series Volume 36
ISBN 978-3-9810843-2-0, August 2007, 330 pages

Parallel Computing: Architectures, Algorithms and Applications -
Book of Abstracts
Book of Abstracts, ParCo 2007 Conference, 4. - 7. September 2007
G.R. Joubert, C. Bischof, F. Peters, T. Lippert, M. Bücker, P. Gibbon, B. Mohr (Eds.)
NIC Series Volume 37
ISBN 978-3-9810843-3-7, August 2007, 216 pages

Parallel Computing: Architectures, Algorithms and Applications -
Proceedings
Proceedings, ParCo 2007 Conference, 4. - 7. September 2007
C. Bischof, M. Bücker, P. Gibbon, G.R. Joubert, T. Lippert, B. Mohr, F. Peters (Eds.)
NIC Series Volume 38
ISBN 978-3-9810843-4-4, December 2007, 830 pages

NIC Symposium 2008 - Proceedings
G. Münster, D. Wolf, M. Kremer (Editors)
Symposium, 20 - 21 February 2008, Forschungszentrum Jülich
NIC Series Volume 39
ISBN 978-3-9810843-5-1, February 2008, 380 pages

From Computational Biophysics to Systems Biology (CBSB08)
Proceedings
Ulrich H.E. Hansmann, Jan H. Meinke, Sandipan Mohanty, Walter Nadler,
Olav Zimmermann (Eds.) (Editors)
NIC Series Volume 40
ISBN 978-3-9810843-6-8, July 2008, 452 pages

All volumes are available online at

http://www.fz-juelich.de/nic-series/.