JYVÄSKYLÄ STUDIES IN COMPUTING
128

Anssi Pennanen

A Graph-based Multigrid with Applications

JYVÄSKYLÄN YLIOPISTO

JYVÄSKYLÄ STUDIES IN COMPUTING 128

Anssi Pennanen

A Graph-based Multigrid with Applications

Esitetään Jyväskylän yliopiston informaatioteknologian tiedekunnan suostumuksella
julkisesti tarkastettavaksi yliopiston Agora-rakennuksen auditoriossa 2
joulukuun 22. päivänä 2010 kello 12.

Academic dissertation to be publicly discussed, by permission of
the Faculty of Information Technology of the University of Jyväskylä,
in the Agora building, auditorium 2, on December 22, 2010 at 12 o'clock noon.

UNIVERSITY OF JYVÄSKYLÄ

JYVÄSKYLÄ 2010

A Graph-based Multigrid with Applications

JYVÄSKYLÄ STUDIES IN COMPUTING 128

Anssi Pennanen

A Graph-based Multigrid with Applications

UNIVERSITY OF JYVÄSKYLÄ

JYVÄSKYLÄ 2010

Editor
Timo Männikkö
Department of Mathematical Information Technology, University of Jyväskylä
Pekka Olsbo, Sini Rainivaara
Publishing Unit, University Library of Jyväskylä

URN:ISBN:978-951-39-4158-1
ISBN 978-951-39-4158-1 (PDF)

ISBN 978-951-39-4155-0 (nid.)
ISSN 1456-5390

Jyväskylä University Printing House, Jyväskylä 2010

ABSTRACT

Pennanen, Anssi
A graph-based multigrid with applications
Jyväskylä: University of Jyväskylä, 2010, 52 p.(+included articles)
(Jyväskylä Studies in Computing
ISSN 1456-5390; 128)
ISBN 978-951-39-4155-0 (nid.), 978-951-39-4158-1 (PDF)
Finnish summary
Diss.

This thesis studies a graph-based multigrid method designed for solving a large, linear and sparse system of equations arising from different discretizations of partial differential equations. The method studied here is an improvement of the method originally introduced by F. Kickinger. The improvements include processing the Dirichlet boundary condition after creating coarse levels, adding flexibility by giving a possibility to use other graphs than the one obtained from the system matrix and fast calculation of coarse level system matrices.

Our method is used as a preconditioner for time-harmonic acoustic and elastic wave equations modelled respectively by the Helmholtz and Navier equations. The equations are discretized by higher-order finite element method and spectral element method. Two approaches are considered. In the first approach damped Helmholtz and Navier operators are used as preconditioners. In the second approach, exact control technique is used to represent time-harmonic problems as time-dependent ones. The original problem is formulated as a least-squares optimization problem that is solved by a preconditioned conjugate gradient method. In both approaches our multigrid method is used to solve a linear system associated with the preconditioning step.

Furthermore, we have used our method as a solver in computational fluid dynamics. Stokes and Navier-Stokes equations are considered. The equations are discretized by a stabilized finite element method and different ILU-type smoothers are used in multigrid cycles. The results of the numerical experiments show that our method is efficient both as a preconditioner and as a stand-alone solver.

Keywords: Algebraic multigrid, preconditioning, exact controllability, Helmholtz equation, Navier equation, Stokes equations, Navier-Stokes equations

Author

Anssi Pennanen
Department of Mathematical Information Technology
University of Jyväskylä
Finland

Supervisors

Professor Tuomo Rossi
Department of Mathematical Information Technology
University of Jyväskylä
Finland

Professor Pekka Neittaanmäki
Department of Mathematical Information Technology
University of Jyväskylä
Finland

Reviewers

Professor Gabriel Wittum
Goethe Center for Scientific Computing (G-CSC)
University of Frankfurt
Germany

Professor Jan Valdman
School of Engineering and Natural Sciences
University of Iceland
Iceland

Opponent

Dr. Yuri Vassilevski
Leading scientist
Institute of Numerical Mathematics
Russian Academy of Science
Russia

ACKNOWLEDGEMENTS

A challenging project like a preparation of a doctoral thesis would never happen without favourable people and communities around you. First of all, I want to thank my supervisors, Professor Tuomo Rossi and Professor Pekka Neittaanmäki for inspiring, patient and encouraging quidance on my way in this field of science. Without them this work would never have begun or finished.

I am very grateful to many different funding sources that have supported my work: Department of Mathematical Information Technology for the funding and giving the facilities for my work; Finnish Cultural Foundation who kindly supported me in the beginning of my research as well as did Jyväskylä Graduate School in Computing and Mathematical Sciences (COMAS). The latest two and a half years of this research was funded by Academy of Finland, project #121271. All the funding given is kindly acknowledged.

I would like to thank all my colleagues in the Department of Mathematical Information Technology, especially Dr. Jari Toivanen, Professor Raino Mäkinen, Ph.Lic. Sanna Mönkölä and Dr. Tuomas Airaksinen for the fruitful co-operation. Special thanks goes to Dr. Janne Martikainen (Numerola Ltd.) for providing the finite element library and familiarizing me with the multigrid methods. Sincere thanks to Dr. Matti Nurmia (Department of Physics) for doing a huge job in checking the linguistic form of this thesis.

On this occasion I also want to express my gratitude to all of my friends and family. Specially I want to thank my fiancée Anna for her loving presence in my life. Last, but truly not least, I would like to express my gratefulness to my late mother Maj-Lis who always encouraged me in my studies and life. Lovely memories of you will always live in my heart.

Kesälahti, December 2010

Anssi Pennanen

LIST OF FIGURES

LIST OF TABLES

CONTENTS

LIST OF INCLUDED ARTICLES

PI Erkki Heikkola, Sanna Mönkölä, Anssi Pennanen, Tuomo Rossi. Controllability method for the Helmholtz equation with higher-order discretizations. *Journal of Computational Physics 225 (2007) 1553–1576*.

PII Tuomas Airaksinen, Erkki Heikkola, Anssi Pennanen, Jari Toivanen. An algebraic multigrid based shifted-Laplacian preconditioner for the Helmholtz equation. *Journal of Computational Physics 226 (2007) 1196–1210*.

PIII Sanna Mönkölä, Erkki Heikkola, Anssi Pennanen, Tuomo Rossi. Time-harmonic elasticity with controllability and higher-order discretization methods. *Journal of Computational Physics 227 (2008) 5513–5534*.

PIV Tuomas Airaksinen, Anssi Pennanen, Jari Toivanen. A damping preconditioner for time-harmonic wave equations in fluid and elastic material. *Journal of Computational Physics 228 (2009) 1466–1479*.

The contribution of the author of this thesis in the articles [PI]–[PIV] has been especially in writing sections concerning graph-based or algebraic multigrid as well as in developing algorithms, software and numerical tests. The multigrid method introduced in this thesis is an improvement of the method by Kickinger [47]. The first improvements were introduced in [53] by Martikainen.

For the numerical tests in the article [PI] the author provided AMG solver code written in C++ and a Fortran interface to the solver. The author also wrote the part concerning the AMG algorithm in the Section 6.1 in the article [PI]. For the article [PII] author provided the AMG solver code and wrote the Section 4. In the articles [PIII] and [PIV] the author wrote the parts describing the use of the multigrid, that is Section 5.3 in the article [PIII] and Section 5 in the article [PIV]. In addition to articles [PI]–[PIV], the thesis consists of results concerning computational fluid dynamics. The results are included in Chapter 3 that is an upgraded version of technical report [54].

1 INTRODUCTION

At the core of countless numerical simulations is a solution of a large, linear and sparse system of equations

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

that may have even billions of unknowns. Equations of this kind arise from different discretizations of partial differential equations, for example those of finite differences, finite element or finite volume method. The most robust ways to solve these equations are direct methods but they are usually computationally too expensive, especially in complex 3D simulations. Even though the computational power of modern computers is exhaustive, the need for efficient iterative solution methods is still reasonable, as simulations carried out with these computers are more and more complex and large.

Iterative solvers try to find a solution of (1) step by step starting from an initial guess $\mathbf{x}^0$ and then updating this approximation sequentially $\mathbf{x}^{i+1} = \mathbf{x}^i + \mathbf{M}^{-1}(\mathbf{b} - \mathbf{A}\mathbf{x}^i), i = 0, 1, \ldots$ until a reasonable approximation is achieved.

First multilevel-type iterative methods to solve equation (1) were introduced in the 1960's by Fedorenko [33, 34], and in the 1970's their real efficiency was discovered by Brandt [10]. At the same time, Hackbusch independently published his multigrid method in [41]. In these *geometric multigrid methods*, a hierarchy of coarser grids and interpolation (and restriction) operators between the grids is fixed, and a smoother is chosen according to the problem at hand. A good introduction to geometric multigrid and its application to a wide variety of problems can be found from Multigrid Methods [42] by Hackbusch, as well as in a more recent book Multigrid [78] by Trottenberg et al.

The efficiency of the geometric multigrid method is based on two main components: error smoothing and coarse grid (or level) correction. The classical iteration methods, Jacobi or Gauß-Seidel iterations, for instance, usually have a strong smoothing effect on the error of the approximate solution. Therefore these iteration methods are called *smoothers* in multigrid terminology. These iteration methods smoothen the error of approximate solution but fail to deduce the error itself. However, it is possible to project a smooth error on a coarser grid where it will be rough again, and we are able to use Jacobi or Gauß-Seidel efficiently. In

this way we obtain a coarse grid correction which can be interpolated back to the finer grid. By repeating this procedure recursively on a coarser and coarser grid or level, we get a multigrid *cycle*.

The geometric multigrid method has its limitations. The computational grid may be so complex that defining coarser (or finer) grids is impossible. This is a problem especially with irregular grids. Even though the discretization is uniform, geometric multigrid may fail if there are large jumps in coefficients or the coefficients are distributed in some particular patterns (see, for example, [2] and [25]). Some problems do not have any geometical background and, thus, no grid at all.

In their papers [12, 13, 11] Brandt et al. attacked these problems by attempting to automate the coarsening process by combining the *Galerkin-principle* and *operator-independent interpolation* in a geometric multigrid. This was the beginning of the development of the *algebraic multigrid method* (AMG). Stüben presented a special AMG algorithm in his paper [73], which was a basis for a famous publication by Ruge and Stüben [66], published in [55], which is often referred as a *classical AMG*.

The strategy in AMG is to use only algebraic data included in the problem at hand. In this context it means that *strong* and *weak* connections between unknowns are determined on the basis of coefficients in the system matrix.

After this is done, a multigrid hierarchy that includes coarse level system matrices and transfer operators between these levels is formed automatically. In this way a very robust method is obtained which could be applied to a wide range of problems. Furthermore, the method could be used as a black box solver, because it was enough to put system matrix and right hand side in to get the solution out and, thus, it could be applied to problems with no geometrical background at all. In [22] the performance of standard AMG is tested in number of numerical experiments.

Because of the wide range of applicability of classical AMG and of the fact that an AMG implementation, AMG1R5, was made available in the public domain, there was not much new in AMG development in the following years. However, the growing demand for black box solvers, especially for commercial simulation software capable of solving bigger and more complex problems, raised again interest in developing AMG in the early and mid 1990's.

In the classical AMG the *set up phase*, that is selecting coarse level unknowns and building transfer operators and coarse level matrices, is computationally quite demanding. A more efficient and more robust interpolation strategy was introduced in RAMG05, the successor of AMG1R5, described in detail in [78]. A variety of different strategies for selecting coarse level variables and smoothers and definition of the interpolation in AMG are introduced, for example, in [85, 84, 32, 14, 16, 15, 83, 6, 3, 4, 60, 82]. A number of these different strategies and exhaustive numerical comparisons of different combinations of these with the classical one can be found in [56]. A comprehensive reviews of AMG appeared in [74] and [79].

Another approach to the coarsening process is to use only the *graph* of the

system matrix. This approach is used, for example, in [9] by Braess, [47] by Kickinger and [7] by Beck. The graph of the system matrix is used to generate coarse systems, that is coarse system matrices and interpolation operators. The main advantage of only using the graph is the fast computation of the coarse level components that can be quite time-consuming in the method by Ruge and Stüben.

In his paper [47] Kickinger introduced two rather simple, yet efficient, algorithms to find coarse grids. The initial graph for a coarsening algorithm is extracted directly from the system matrix. Therefore, his method could be used as a pure black-box solver. We have modified the method so that any graph related to the original problem can be fed into the method. Thus, we have given up the notion of a black box solver, as we will also use other graphs than the one obtained from the original system matrix. Particularly, we have used a graph that contains information whether a node in the graph corresponds to a Dirichlet boundary node. This information is also carried to the coarser levels. Motivation for this is based on the following observation: The elimination of the equations corresponding to Dirichlet boundary conditions *after* the coarsening process increases the stability of the method.

Furthermore, when higher order elements are used in a finite element discretization, we construct an initial graph where a node is connected only to the nearest neighbouring nodes. The reason for doing this is that using the original graph of the system matrix would produce far too coarse systems which would degrade the convergence of the multigrid solver. Under each application in the later chapters it is explained more specifically how the initial graph for the coarsening process is chosen.

Our method is between a geometric and an algebraic multigrid. On the one hand, it is geometric, because the selection of coarse grid variables is more geometrical than algebraic and we change the smoother appropriately according to the problem at hand. On the other hand, our method is also algebraic multigrid, as only the finest discretization mesh and the related algebraic system are needed in order to construct components for the whole multigrid cycle.

The structure of the thesis is the following: In this introductory chapter our graph-based multigrid is discussed. In Section 1.1 technical details of our multigrid method are presented. This includes selecting coarse level variables and building necessary transfer operators between the levels. Furthermore, an efficient way to calculate coarse level system matrices by the Galerkin principle is introduced. Finally, an algorithm for a whole AMG cycle is presented. In Section 1.2 different smoothers for AMG are discussed. Only the smoothers that are used in this thesis are presented.

Chapter 2 focuses on the numerical solution of time-harmonic wave equations. In Section 2.1 the Helmholtz equation describing an acoustic pressure wave propagating in a fluid is presented. Two different approaches for the numerical solution of the Helmholtz equation are introduced. In Section 2.1.2 a (physical) damping preconditioner for the numerical solution of the Helmholtz equation is discussed. The same kind of preconditioner for the Navier equation is introduced

in Section 2.2.1. In these cases a graph-based multigrid studied in this thesis is used to obtain an approximation of the inverse of the preconditioner. Another approach for the numerical solution of Helmholtz and Navier equations is studied in Sections 2.1.3 and 2.2.2. In this approach, a time-harmonic wave equation is cast back to a time-dependent equation. By using control techniques, the wave problem is formulated as a least-squares optimization problem that is solved by the preconditioned conjugate gradient (PCG) method. At this stage, the multigrid method is utilized in the preconditioning step of the PCG method.

In Chapter 3 we test our multigrid method as a stand-alone solver for the computational fluid dynamic problems. In this chapter steady state Stokes and Navier-Stokes equations describing the motion of incompressible fluids are studied. They are discretized by the stabilized finite element method. The behaviour of our algorithm is investigated in numerical examples in the Section 3.4. Finally, in Chapter 4 we draw some conclusions about the method studied in this work and present some ideas for developing our method in the future.

1.1 The AMG algorithm

The algorithmic structure of the AMG initialization phase is presented in Algorithm 1. The coarsening process operates in a geometric fashion by sequentially choosing a coarse node and eliminating the neighbouring nodes of the graph. There is one important exception. If an interior node is selected as a coarse grid node, its neighbouring node is not eliminated from the graph if this neighbour is tagged as a Dirichlet boundary node. This way we can increase the stability of our method by assuring that a sufficient number of nodes corresponding to the Dirichlet boundary are selected to the coarse levels. However, this is relevant only when we are using unstructured meshes with an arbitrary numbering of nodes. With structured meshes our coarsening strategy produces hierarchical coarse 'grids'.

The primary criterion used here for selecting the next coarse grid node is to take the node with the minimum degree (taking into account the eliminations). The secondary criterion is to follow the original numbering. In Figure 1 there is an example of our coarsening strategy on an unstructured element mesh with first-order Lagrangian triangular elements. The Dirichlet boundary condition is imposed on every boundary. After choosing coarse nodes, a restriction operator and a coarse graph are formed. The restriction matrix can be defined easily by setting the elements corresponding to the neighbours of selected coarse node i in the i:th row to one. After this, every column sum in the restriction matrix is

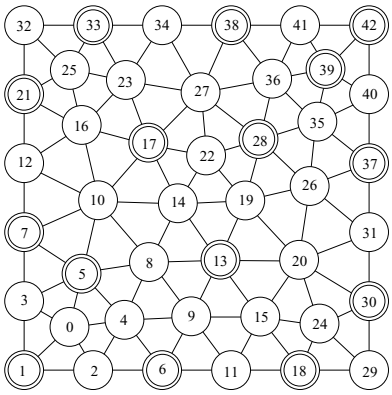

FIGURE 1 Graph nodes selected to coarse level. Nodes with double circle are selected to coarse level.

scaled to one. The elements of the restriction matrix are defined as follows:

$$\mathbf{R}_{ij} = \begin{cases} 1 & \text{for a fine grid point } j \text{ which is a coarse grid point } i, \\ \frac{1}{k} & \text{for a fine grid point } j \text{ which is a neighbor of coarse grid} \\ & \text{point } i \text{ and has } k \text{ neighboring coarse grid points,} \\ 0 & \text{otherwise.} \end{cases}$$

A coarse graph can be formed using the restriction matrix. Each coarse graph node corresponds to a row in the restriction matrix and two graph nodes are connected if, and only if, the corresponding rows of the restriction matrix have a nonzero element in the same column.

ALGORITHM 1 (AMG initialization)

Input: System matrix $\mathbf{A}_0$, initial graph G_0 and
maximum coarse system size n_c.
1. $k = 0$
2. **while** size of $\mathbf{A}_k$ is greater than n_c
3. Select the set of coarse nodes from the graph G_k
4. Form the restriction matrix $\mathbf{R}_k$
5. Create the graph G_{k+1}
6. Calculate the next system matrix $\mathbf{A}_{k+1} = \mathbf{R}_k \mathbf{A}_k \mathbf{R}_k^T$
7. k=k+1
8. **foreach** $i = 0, \ldots, k$
9. Eliminate the rows and columns of $\mathbf{A}_i$ marked in the graph G_i
10. **foreach** $i = 0, \ldots, k-1$
11. Eliminate the columns of $\mathbf{R}_i$ marked in the graph G_i
12. Eliminate the rows of $\mathbf{R}_i$ marked in the graph G_{i+1}
13. Factorize the coarsest matrix $\mathbf{A}_k$

Here the prolongation operator has been chosen as the transpose of the restriction and therefore it is not stored. The Galerkin formula $\mathbf{A}_{k+1} = \mathbf{R}_k \mathbf{A}_k \mathbf{R}_k^T$ is used in order to obtain the coarse system matrices. The coarsening process is continued until the size of the coarsest system is not greater than a given value, measured in numbers of unknowns. The coarsest level problem is solved by an LU-factorization. Because of this, the coarsest level matrix is stored either as a full matrix or as a band matrix, if the bandwidth is small enough. The factorization is then calculated 'in place'.

Let us study the practical implementation of steps (5) and (6) of Algorithm 1. In the implementation sparse matrices are stored in compressed row storage (CRS) format. There are two arrays for each row where the column indices and the values of the nonzero elements are stored. The elements are stored in the increasing column index order. Before step (5) an auxiliary indexing for $\mathbf{R}_k$ is created, which lists for each column the rows with a nonzero element in the column. Then, an arc in the graph G_{k+1} can be formed between any two nodes that are present in such a list. Especially, all nodes are linked to themselves.

This auxiliary indexing will also be used in step (6). The system matrix calculation according to the Galerkin formula is performed in two phases: first we calculate $\mathbf{T}_k = \mathbf{A}_k \mathbf{R}_k^T$ and then $\mathbf{A}_{k+1} = \mathbf{R}_k \mathbf{T}_k$. The matrix multiplication can be represented with (sparse) vector inner products. In the first phase we take row vectors of $\mathbf{A}_k$ and column vectors of $\mathbf{R}_k^T$, which are row vectors of $\mathbf{R}_k$. Thus, the storage format supports the inner product operation. The temporary result matrix $\mathbf{T}_k$ is stored as its transpose. In the second phase, we take row vectors of $\mathbf{R}_k$ and column vectors of $\mathbf{T}_k$, which are row vectors of $\mathbf{T}_k^T$. Still, we can not afford to calculate every element of $\mathbf{T}_k$ and $\mathbf{A}_{k+1}$. Here, we can reuse the auxiliary indexing formed in the beginning of step (5). For the row i we form a set of column indexes that is a union of all the auxiliary index lists of those columns j for which there is a nonzero element $(\mathbf{A}_k)_{ij}$. This is illustrated in the Figure 2.

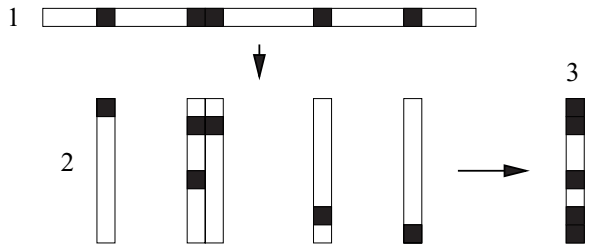

FIGURE 2 Finding out the sparsity pattern in the matrix multiplication.

In phase one we have the indexes of the nonzero elements of a single row i of the matrix $\mathbf{A}_k$. In phase two we collect the row indexes of nonzero elements of $\mathbf{R}_k$ in the corresponding locations and in phase three we form the union of these locations obtaining the final index set for nonzero entries in matrix $\mathbf{T}_k$ in row i. The same procedure can be used to find out the row indexes of all the possible nonzero elements of $\mathbf{A}_{k+1}$ in a column j. In case of the compressed column

storage (CCS) format, $\mathbf{R}_k^T$ are stored and $\mathbf{R}_k\mathbf{A}_k$ are calculated first to a temporary matrix, then $\mathbf{A}_{k+1}$.

The structure of AMG iteration is exactly the same as for any multigrid. It follows Algorithm 2 which has been written in recursive form. The algorithm presented here is the general μ-cycle algorithm. The choices $\mu = 1$ and $\mu = 2$ correspond to *V-cycle* and *W-cycle*, respectively. The initial call of the algorithm has to be made for the finest level.

ALGORITHM 2 Algorithm for the AMG iteration

Input: System matrix $\mathbf{A}_0$, current approximation $\mathbf{x}_0$,
right hand side vector $\mathbf{b}_0$
Output: The iterated approximate solution $\mathbf{x}^*$

if on the finest level
  while not converged
    Presmooth ν times: $\mathbf{x}_0 := \mathbf{x}_0 + \mathcal{S}_0^\nu(\mathbf{b}_0, \mathbf{A}_0, \mathbf{x}_0)$
    Restrict the residual: $\mathbf{b}_1 := \mathbf{R}_0(\mathbf{b}_0 - \mathbf{A}_0\mathbf{x}_0)$
    Set $\mathbf{x}_1 := 0$ and call cycle μ times for the next level
    Prolongate the correction: $\mathbf{x}_0 := \mathbf{x}_0 + \mathbf{R}_0^T\mathbf{x}_1$
    Postsmooth ν times: $\mathbf{x}_0 := \mathbf{x}_0 + \mathcal{S}_0^\nu(\mathbf{b}_0, \mathbf{A}_0, \mathbf{x}_0)$
if on an intermediate level l
  Presmooth ν times: $\mathbf{x}_l := \mathbf{x}_l + \mathcal{S}_l^\nu(\mathbf{b}_l, \mathbf{A}_l, \mathbf{x}_l)$
  Restrict the residual: $\mathbf{b}_{l+1} := \mathbf{R}_l(\mathbf{b}_l - \mathbf{A}_l\mathbf{x}_l)$
  Set $\mathbf{x}_{l+1} := 0$ and call cycle μ times for the level $l+1$
  Prolongate the correction: $\mathbf{x}_l := \mathbf{x}_l + \mathbf{R}_l^T\mathbf{x}_{l+1}$
  Postsmooth ν times: $\mathbf{x}_l := \mathbf{x}_l + \mathcal{S}_l^\nu(\mathbf{b}_l, \mathbf{A}_l, \mathbf{x}_l)$
if on the coarsest level k
  Solve $\mathbf{x}_k$ from $\mathbf{A}_k\mathbf{x}_k = \mathbf{b}_k$

In this thesis a finite element library written in C++ by Dr. Janne Martikainen is used. The library includes all the usual data structures needed in a finite element method, for instance vectors, matrices, mesh and different finite elements. Data structures are implemented as generic classes so that they can be initialized with different data types. After a finite element stiffness matrix is assembled, the user initializes a graph-based multigrid object with the stiffness matrix, an initial graph and the maximum number of unknowns on the coarsest level. After the setup phase is done, user invokes the solver method of the graph-based multigrid class with the stiffness matrix, a vector containing an initial guess, a right hand side vector, stopping criterion and the maximum number of the multigrid iterations. The W-cycle is used as a default iteration which can be naturally switched to the V-cycle. The result is stored in the vector that contained the initial guess. The finite element library and the graph-based multigrid solver code are available for academic purposes upon a request. A technical manual of the finite element library will be available in the near future in the report series of the Department of Mathematical Information Technology.

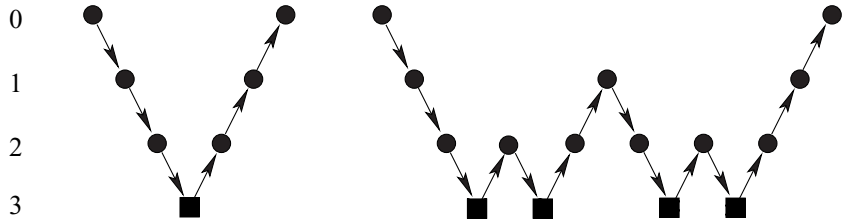

FIGURE 3 Multigrid cycles. V-cycle on the left and W-cycle on the right.

Two different multigrid iterations, called cycles, are illustrated in Figure 3. The cycles are called V-cycle and W-cycle, depending on how many times coarser levels are visited in one cycle. In Figure 3 black dots represent the action of the smoother on levels $0 \ldots 2$. On the coarsest level 3 a direct solver is applied to solve $\mathbf{A}_3\mathbf{x}_3 = \mathbf{b}_3$, marked by a black square. Arrows pointing downwards describe restriction and arrows pointing upwards describe prolongation.

1.2 Smoothers for AMG

In this thesis, different kinds of smoothers for graph-based multigrid iteration are used depending on the application area. With the damping preconditioner for the Helmholtz and Navier equations introduced in sections 2.1.2 and 2.2.1, respectively, we have used *weighted Jacobi* (ω-Jacobi) iterations as a smoother. Depending on the situation, different weight parameters ω are used.

In sections 2.1.3 and 2.2.2, where a controllability method for the Helmholtz and Navier equations is studied, a *Successive Over Relaxation* method is employed as a smoother. The weights for Jacobi ω-Jacobi and relaxation parameter for SOR are problem dependent; different parameters used in numerical experiments are shown in the included articles [PI]–[PIV].

A decomposition of matrix $\mathbf{A}$ to lower and upper triangular matrices $\mathbf{L}$ and $\mathbf{U}$, $\mathbf{A} = \mathbf{LU}$, usually produces much more dense matrices $\mathbf{L}$ and $\mathbf{U}$ than the original matrix $\mathbf{A}$, even if $\mathbf{A}$ is a sparse matrix. This can be avoided by applying a dropping rule, or zero pattern, in the decompositioning process, which means that we leave out some (off-diagonal) elements of the LU decomposition. In the following Algorithm 3 we present a general incomplete LU (ILU) decomposition of the matrix $\mathbf{A}$. Different kinds of incomplete factorizations have been proven to be good smoothers for multigrid methods. See, for example, a work by Wittum [87].

If matrix $\mathbf{A}$ is needed after the decomposition, then the space required to store elements of ILU decomposition must be reserved. One of the simplest dropping rule is to preserve the original non-zero pattern of matrix $\mathbf{A}$, i.e. no fill-in is allowed. This factorization is called zero fill-in ILU (ILU(0)) and its pseudo code is documented in Algorithm 4.

Reserving the original non-zero pattern of matrix $\mathbf{A}$ can be viewed as a static

ALGORITHM 3 General incomplete LU decomposition

Input: Matrix **A**, zero pattern P
Output: Incomplete LU decomposition of **A**
1. **foreach** $i = 2, \ldots, n$
2. **foreach** $k = 1, \ldots, i-1$ **&&** (i,k) $\notin$ P
3. $a_{ik} := a_{aik}/a_{kk}$
4. **foreach** $j = k+1, \ldots, n$ **&&** (i,j) $\notin$ P
5. $a_{ij} := a_{ij} - a_{ik} * a_{kj}$

ALGORITHM 4 ILU(0) decomposition

Input: Matrix **A**, non-zero pattern NZ of **A**
Output: Incomplete LU decomposition of **A**
1. **foreach** $i = 2, \ldots, n$
2. **foreach** $k = 1, \ldots, i-1$ **&&** (i,k) $\in$ NZ
3. $a_{ik} := a_{ik}/a_{kk}$
4. **foreach** $j = k+1, \ldots, n$ **&&** (i,j) $\in$ NZ
5. $a_{ij} := a_{ij} - a_{ik} * a_{kj}$

pattern for dropping elements in the factorization. One may obtain more robustness in the factorization by allowing some fill-in. It may be done statically, by using some pre-defined fill-in pattern, or dynamically, by introducing for example a threshold value as a dropping rule. This means that in the LU decomposition process we drop the elements that are smaller than a given tolerance. The problem here is that it is impossible to know how much fill-in there will be. One possibility is to apply, in addition to the dropping tolerance, a number for the largest elements per row that are finally stored. This strategy called ILUT was introduced by Saad in [67] and it is formulated here as in [68] of Saad:

ALGORITHM 5 ILUT decomposition

Input: Matrix **A**, dropping tolerance ϵ, maximum no. of elements per row p
Output: Incomplete LU decomposition of **A**
1. **foreach** $i = 1, \ldots, n$
2. $w := a_{i*}$
3. **foreach** $k = 1, \ldots, i-1$ **&&** $w(k) \neq 0$
4. $w_k := w_k/a_{kk}$
5. Apply dropping rule to w_k
6. **if** $w_k \neq 0$
7. $w := w - w_k * u_k$
8. Apply a dropping rule to row w
9. $l_{i,j} := w_j$ **for** $j = 1, \ldots, i-1$
10. $u_{i,j} := w_j$ **for** $j = i, \ldots, n$
11. $w := 0$

In Chapter 3, these ILU-type smoothers are applied when solving incompressible fluid flow problems with AMG. A great deal of different smoothers for multigrid methods can be found in the literature. For example, see the book by Trottenberg et al. [78] and the references therein.

2 AMG PRECONDITIONING IN WAVE SCATTERING

We are surrounded by waves. Our hearing is based on sensing pressure changes in the air, created by some sound source producing pressure waves with a suitable frequency for a human ear. Data are transmitted by electro-magnetic waves from a cell phone antenna to a base station, and vice versa, to let one hear someone else's voice from a more or less longer distance. Foetal development can be traced in ultrasonography using a device for sending and receiving ultrasonic waves. These are just a few examples of waves vibrating everywhere around us.

The numerical solution of wave equations has many challenges that have been under active research in a few past decades. A fundamental problem of the numerical wave simulation is related to the famous Nyquist–Shannon sampling theory [70]:

> "If a function $f(t)$ contains no frequencies higher than W hertz, it is completely determined by giving its ordinates at a series of points spaced $\frac{1}{2W}$ seconds apart."

It states that a signal can be reproduced from discrete sampling points if more than two samples per wave of the highest frequency is captured. In the space domain characterized by the wavelength λ it means that at least two mesh points per wavelength are needed.

However, this is only a lower limit for the resolution of the mesh. In practise, a much more dense mesh must be used for simulation of the waves to reach sufficient accuracy. It is shown (see papers [45] and [46] by Ihlenburg and Babuška) that for high frequency problems the error of the FEM approximation is proportional to $k^{p+1}h^p$, where p is the order of the basis functions, k is the wave number and h is the mesh step size. This *pollution effect* is the main reason that the number of grid points grows rapidly, especially in 3D problems at high frequencies. Thus, linear systems become very large and some sophisticated solution method is needed to solve the equation.

In this chapter we discuss two different solution methods for time-harmonic wave scattering in fluids and elastic materials. In both of the methods AMG described in the previous chapter is used as a preconditioner. One is applying

a (physical) damping preconditioner to time-harmonic wave equations. In this case AMG is used in approximating the inverse of damped operators. Erlangga et al. proposed in [30] to use an imaginary shifted-Laplacian operator as a preconditioner for the Helmholtz equation. They applied one cycle of geometric multigrid to approximate the inverse of the operator. In [PII] we extended this idea but instead of using geometric multigrid, we employed our graph-based multigrid. Furthermore, in [PIV] we applied this idea also to solving the Navier equation describing the behavior of linearly elastic material.

Another approach to solving wave equations is falling back from a time-harmonic case to a time-dependent case and then using control techniques to search for a time-periodic solution. We study here a method originally introduced by Bristeau, Glowinski and Périaux (see, for example [17, 18, 19, 39]), whose practical realization we have improved. In this approcah the AMG method is used as a preconditioner for the conjugate gradient method (CG) ([43], [48]). This technique, applied both to Helmholtz and Navier equations, is presented briefly in Sections 2.1.3 and 2.2.2. A detailed introduction to the method and numerical experiments can be found in papers [PI, PIII] as well as in [58] of Mönkölä. A comparison of these two different approaches applied to the Helmholtz equation can be found in [1] by Airaksinen and Mönkölä.

2.1 Wave scattering in fluids

As long as the changes in velocity, density and deformation are small compared to unity, mechanical waves can be described by *linear* wave equations [49]. Let $P(\vec{x},t)$ denote an acoustic pressure field, $\rho(\vec{x})$ the density of the medium in which the acoustic wave propagates and $F(\vec{x},t)$ a source term. The propagation of the wave in the medium is then described by the linear acoustic wave function:

$$\frac{1}{\rho(\vec{x})c(\vec{x})^2}\frac{\partial^2 P}{\partial t^2} - \nabla\cdot\left(\frac{1}{\rho(\vec{x})}\nabla P\right) = F. \tag{1}$$

Here, t is time and $c(\vec{x})$ is the speed of sound in the medium. We are aiming at solving time-harmonic wave equations. By substituting $P(\vec{x},t) = p(\vec{x})e^{-i\omega t}$, $F(\vec{x},t) = f(\vec{x})e^{-i\omega t}$ and $k = \frac{\omega}{c(\vec{x})}$, ω and c being strictly positive functions, we get the time-harmonic Helmholtz equation

$$-\nabla\cdot\frac{1}{\rho(\vec{x})}\nabla p(\vec{x}) - \frac{k(\vec{x})^2}{\rho(\vec{x})}p(\vec{x}) = f(\vec{x}). \tag{2}$$

The so-called wave number k denotes how many wave lengths there are per unit length, and it varies as the speed of sound $c(\vec{x})$ in the medium varies. ω is the angular frequency.

The boundary Γ of the domain Ω is composed of two non-overlapping sets, $\Gamma = \Gamma_d \bigcup \Gamma_i$. At least one of these sets is non-empty. On boundary Γ_d a sound-soft

boundary is described by the Dirichlet boundary condition

$$p = g_d \quad \text{on } \Gamma_d \tag{3}$$

The impedance boundary condition is expressed by

$$-i\gamma kp + \frac{\partial p}{\partial \vec{n}} = g_i, \quad \text{on } \Gamma_i \tag{4}$$

where $\vec{n}(\vec{x})$ is the outer normal vector and i is the imaginary unit. The absorbency coefficient $\gamma(\vec{x}) \in [0, 1]$ describes the amount of absorption on the boundary Γ_i.

Usually, phenomena like radar wave scattering from an obstacle (aeroplane, submarine, for example) take place in an infinite domain. It would be unreasonable, and even meaningless, to model the whole domain while we are actually interested in what happens near the obstacle. That is why these *exterior problems* in infinite domains are truncated into a finite computational domain. An example of a truncated domain Ω for the calculation of a wave scattering by an obstacle is depicted in Figure 4.

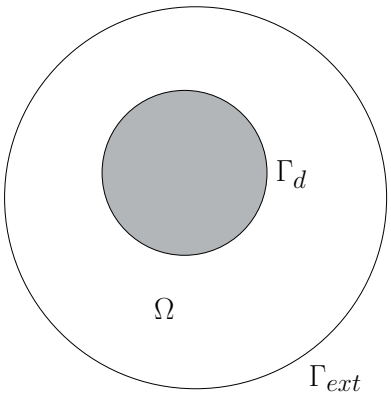

FIGURE 4 Computational domain Ω and artificial boundary Γ_{ext}. In the middle is an obstacle with sound-soft boundary Γ_d.

At a "sufficient" distance from the obstacle, on an artificial boundary, an absorbing boundary condition is imposed to avoid non-physical reflections from the boundary. Here we have done this by setting $\gamma = 1$ in the above impedance boundary condition (4). It describes a first-order approximation of the Sommerfeld condition [72] expressing the propagation of the wave through this boundary. It could be replaced by a second-order approximation to achieve better accuracy; see the work by Enquist and Majda [28]. There are also several other formulations of absorbing boundary conditions and other techniques (for instance, Perfectly Matched Layer) to reduce non-physical reflections. For these, see recent articles by Thompson [77] and Erlangga [29] and the references therein.

2.1.1 Discretization of the Helmholtz equation

The Helmholtz equation (2) is a partial differential equation (PDE), whose analytical solution is in general not known. However, it is possible to compute an

approximation of the solution by using suitable numerical methods. To do this, continuous functions in the PDE must be replaced by their discrete counterparts. There are different strategies to discretize partial differential equations: for example, finite differences [57], finite volume methods (FVM) [31], boundary element methods (BEM) [65] and finite element methods (FEM) [21, 88]. There are also some special discretizations that are related to the FEM, for example infinite element methods (IFEM) [8, 38], hp–FEM [5, 24] and spectral element methods (SEM) [63, 52, 64]. In this thesis we concentrate on FEM and SEM discretizations.

For the FEM discretization we need to introduce a weak (or variational) formulation of the Helmholtz equation. We choose a test function space V_0 and a solution space V_g as

$$V_g = \{q \in H^1(\Omega) : q = g(\mathbf{x}) \text{ on } \Gamma_d\}. \tag{5}$$

By multiplying equation (2) by test function $q \in V_0$, integrating over the domain Ω and using partial integration together with Green's formulae, we obtain a weak formulation for the Helmholtz equation: Find $p \in V_g$ such that

$$\int_\Omega \frac{1}{\rho}(\nabla p \cdot \nabla \bar{q} - k^2 p\bar{q})\,\mathrm{d}x - \int_{\Gamma_i} \frac{1}{\rho}(i\gamma k p\bar{q})\,\mathrm{d}s - \int_{\Gamma_a} \frac{1}{\rho}(ikp\bar{q})\,\mathrm{d}s = \int_\Omega f\bar{q}\,\mathrm{d}x \tag{6}$$

for all $q \in V_0$. For a finite element discretization of physical domain Ω, a mesh $\mathcal{T}$ is defined such that

$$\overline{\Omega}_h = \bigcup_{K\in\mathcal{T}} K,$$

that is, Ω_h is an approximation of Ω. If the boundary of the original domain Ω is polygonal, then $\Omega_h = \Omega$. The mesh $\mathcal{T}$ consists of either triangular or quadrilateral elements K in two-dimensional and of tetrahedra in three-dimensional problems.

2.1.2 A damping preconditioner for the Helmholtz equation

Discretization of weak the form (6) of equations (2)-(4) by the finite element method leads to a sparse and linear equation of form

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \quad \mathbf{A} \in \mathbb{C}^{N\times N},\ \mathbf{x}, \mathbf{b} \in \mathbb{C}^N, \tag{7}$$

where N refers to the number of grid points. Complex-valued coefficient matrix $\mathbf{A}$ is indefinite, symmetric and non-Hermitian. To maintain the accuracy of the discretization, a very high number of gridpoints is required, which results in a highly indefinite and large linear system. Direct methods suffocate because of substansial fill-in, especially in 3D. Modern iterative solution methods, like Krylov subspace iterative methods, for example Saad's GMRES [69] or Bi-CGSTAB [80] by van der Vorst, multigrid methods and domain decomposition methods, have low memory-usage (at least when compared to direct methods), but in Helmholtz-type problems they suffer from significant slow-down of convergence without a proper preconditioning.

A numerical solution of the Helmholtz equation has attracted a lot of researchers working on the field of computational science. Numerous different strategies for discrete Helmholtz equations have been introduced in recent years. A good review of the development of this topic can be found in [77] of Thompson. An introductory book of the topic is [44] by Ihlenburg, which handles a reliable simulation of scattering problems using finite element methods for the Helmholtz equation with moderate and large wave numbers.

Here we have chosen to use right preconditioning, where the original equation (7) is replaced by the equation

$$\mathbf{A}\mathbf{B}^{-1}\mathbf{y} = \mathbf{b}, \quad \mathbf{y} = \mathbf{B}\mathbf{x}. \tag{8}$$

The preconditioner matrix $\mathbf{B}$ is chosen so that the condition number of the preconditioned system $\mathbf{A}\mathbf{B}^{-1}$ will be reduced. We will use a preconditioner introduced by Erlangga et al. in [30] that is based on a discretized shifted-Laplacian operator

$$\mathcal{B} = -\nabla \cdot \frac{1}{\rho(\vec{x})}\nabla - (\beta_1 + \beta_2 i)\frac{k(\vec{x})^2}{\rho(\vec{x})}, \quad \beta_1,\, \beta_2 \in \mathbb{R}, \tag{9}$$

where parameters β_1 and β_2 can be freely chosen. A damped Helmholtz operator will be obtained by choosing $\beta_1 = 1$ and β_2 to be positive. The boundary conditions and the discretization of the operator $\mathcal{B}$ are identical to those of the original problem (2). The computation of the exact inverse of the preconditioner $\mathbf{B}$ would require as much computational effort as solving the original problem. Thus, we use one W-cycle of multigrid presented in the previous chapter with one pre- and one post-smoothing sweep by under-relaxed Jacobi iteration to approximate the inversion.

2.1.3 Controllability method for the Helmholtz equation

The exact controllability method for partial differential equations is based on the Hilbert Uniqueness Method (HUM), introduced by J.L. Lions in [50]. The method we use here to find time periodic solutions for the Helmholtz equation was introduced in [17]. Instead of solving time-harmonic equations (2)–(4) directly, they can be substituted for solving an exact controllability problem

$$\frac{\partial^2 P}{\partial t^2} - \nabla^2 P = f, \quad \text{in } \Omega \times [0,T], \tag{10}$$

$$P = G_d, \quad \text{on } \Gamma_d \times [0,T], \tag{11}$$

$$\frac{\partial P}{\partial t} + \frac{\partial P}{\partial \vec{n}} = G_i, \text{ on } \Gamma_i \times [0,T], \tag{12}$$

$$P(\vec{x},0) = e_0, \qquad \frac{\partial P}{\partial t}(\vec{x},0) = e_1, \quad \text{in } \Omega, \tag{13}$$

$$P(\vec{x},T) = e_0, \qquad \frac{\partial P}{\partial t}(\vec{x},T) = e_1, \quad \text{in } \Omega. \tag{14}$$

The idea is to find such initial conditions $\mathbf{e} = (e_0, e_1)$ that the solution P and its time derivative at time T would coincide with the initial conditions. The numerical solution for this is achieved by minimizing the difference between the initial

conditions and values of the corresponding variables after time period T. Thus, the problem is formulated as a least-squares optimization problem

$$\min\left\{\frac{1}{2}\int_\Omega \left|\frac{\partial P(\vec{x},T)}{\partial t} - e_1\right|^2 \mathrm{d}x + \frac{1}{2}\int_\Omega |\nabla(P(\vec{x},T) - e_0|^2 \,\mathrm{d}x\right\}. \tag{15}$$

The space discretization of the weak formulation of the wave equation (10)–(12) is done by the Spectral Element Method. The discretized objective function of the least-squares problem (15) reads

$$\frac{1}{2}(u^N - e_0)^T \mathbf{K}(u^N - e_0) + \frac{1}{2}\left(\frac{\partial u^N}{\partial t} - e_1\right)^T \mathbf{M}\left(\frac{\partial u^N}{\partial t} - e_1\right) \tag{16}$$

where $\mathbf{M}$ is the mass matrix and $\mathbf{K}$ is the stiffness matrix arising from the discretization. Time is discretized by the fourth-order Runge-Kutta method (see, for example [35]).

According to Maday et al. [51], using the conjugate gradient (CG) method without any preconditioning to solve discrete system (16) would lead to a growing number of iterations when the order of the elements grows. Here we have used preconditioned CG described in Algorithm 6 to avoid a slow-down of the convergence.

ALGORITHM 6 (Preconditioned CG algorithm)

1. Compute the initial values e_1 and e_0.
2. Solve the state equation $s(\mathbf{e}, \mathbf{u}(\mathbf{e})) = 0$.
3. Solve the adjoint state equation $\left(\frac{\partial s(\mathbf{e},\mathbf{u}(\mathbf{e}))}{\partial \mathbf{u}(\mathbf{e})}\right)^T \mathbf{p} = \left(\frac{\partial J(\mathbf{e},\mathbf{u}(\mathbf{e}))}{\partial \mathbf{u}(\mathbf{e})}\right)^T$.
4. Compute the gradient vectors g_0 and g_1.
5. Solve the linear system with the preconditioner $\mathbf{L}\mathbf{w} = -\mathbf{g}$.
6. Set $c_0 = -(\mathbf{w}, \mathbf{g})$, $c = c_0$ and $i = 1$.
7. Repeat until $\sqrt{\frac{c}{c_0}} < \varepsilon$:
8. Solve the state equation $s(\mathbf{w}, \mathbf{u}(\mathbf{w})) = 0$.
9. $\left(\frac{\partial s(\mathbf{w},\mathbf{u}(\mathbf{w}))}{\partial \mathbf{u}(\mathbf{w})}\right)^T \mathbf{p} = \left(\frac{\partial J(\mathbf{w},\mathbf{u}(\mathbf{w}))}{\partial \mathbf{u}(\mathbf{w})}\right)^T$.
10. Compute the gradient updates v_0 and v_1.
11. Compute $\beta = \frac{c}{(\mathbf{w},\mathbf{v})}$.
12. $\mathbf{e}^i = \mathbf{e}^{i-1} + \beta\mathbf{w}$.
13. $\mathbf{g} = \mathbf{g} + \beta\mathbf{v}$.
14. Solve linear system with the preconditioner $\mathbf{L}\mathbf{v} = -\mathbf{g}$.
15. $\gamma = \frac{1}{c}$, $c = -(\mathbf{v}, \mathbf{g})$, $\gamma = c\gamma$.
16. $\mathbf{w} = v + \gamma w$, $i = i + 1$.

In the preconditioning steps (5) and (14) in the Algorithm 6 we use the block-diagonal preconditioner

$$\mathbf{L} = \begin{bmatrix} \mathbf{K} & 0 \\ 0 & \mathbf{M} \end{bmatrix}, \tag{17}$$

where the matrices **K** and **M** are the stiffness and mass matrix, respectively. In spectral elements, the nodes of the element are located at the Gauss-Lobatto quadrature points, so the mass matrix is diagonal and the calculation of the inversion is straightforward. The preconditioning with the stiffness matrix **K** is computed by using the AMG method presented in Section 1.1. In practical simulations where dozens of CG iterations are needed to reach sufficient accuracy it is highly important to have a fast solver like AMG for the preconditioner.

2.2 Wave scattering in elastic materials

A time-harmonic wave $\vec{U}(\vec{x},t) = \vec{u}(\vec{x})e^{-i\omega t}$, where t is time and ω is angular velocity, propagating in an elastic medium is modelled by the Navier equation

$$-\omega^2\rho\vec{u} - \nabla\cdot\bar{\bar{\sigma}}(\vec{u}) = \vec{f}, \quad \text{in } \Omega. \tag{18}$$

In the above equation (18) $\vec{f}$ is the force term and ρ is the density of the material. Stress and strain forces are bound together through Hooke's law describing the stress tensor

$$\bar{\bar{\sigma}}(\vec{u}) = \lambda(\nabla\cdot\vec{u}) + 2\mu\bar{\bar{e}}(\vec{u}) \tag{19}$$

and the strain tensor

$$\bar{\bar{e}}(\vec{u}) = \frac{1}{2}(\nabla\vec{u} + (\nabla\vec{u})^T). \tag{20}$$

The Lamé parameters λ and μ are defined as follows:

$$\lambda(\vec{x}) = \frac{E}{2(1+\nu)},\ \mu(\vec{x}) = \frac{E\nu}{(1+\nu)(1-2\nu)}. \tag{21}$$

The stiffness of an isotropic material is measured by tensile (Young's) modulus $E(\vec{x})$. Poisson's ratio $\nu(\vec{x})$ is the ratio of transverse contraction strain to longitudinal extension strain in the direction of stretching force. Furthermore, the speed of pressure and shear waves, c_p and c_s respectively, can be expressed as functions of Lamé parameters and of the density ρ:

$$c_p = \sqrt{\frac{\lambda+2\mu}{\rho}},\ c_s = \sqrt{\frac{\mu}{\rho}}. \tag{22}$$

The wavelengths and wave numbers for pressure and shear waves are

$$\lambda_i = c_i\frac{2\pi}{\omega}, \quad k_i = \frac{\omega}{c_i}, \quad i = p,s. \tag{23}$$

To solve the equation (18) boundary conditions must be expressed. Here we have used two kinds of boundary conditions by dividing the boundary $\partial\Omega$ of an elastic material into two non-overlapping sets $\partial\Omega = \Gamma_d \cup \Gamma_i$ such that at least one or the other of these sets is not empty. By the Dirichlet boundary condition

$$\vec{u} = \vec{g_d} \quad \text{on } \Gamma_d, \tag{24}$$

we can describe, for example, a non-movable part of the body by setting $\vec{g} = \vec{0}$, or a vibrating source by setting $\vec{g}$ as something different from zero. The absorbing boundary condition for the Navier equation is

$$i\omega\rho\mathbf{B}\vec{u} + \bar{\bar{\sigma}}(\vec{u})\vec{n} = \vec{0} \quad \text{on } \Gamma_i, \tag{25}$$

where the components of the matrix $\mathbf{B}$ are defined by

$$\mathbf{B}_{ij} = c_p n_i n_j + c_s t_i t_j, \text{ in 2D, and} \tag{26}$$

$$\mathbf{B}_{ij} = c_p n_i n_j + c_s t_i t_j, + c_s s_i s_j \text{ in 3D,} \tag{27}$$

where c_p and c_s are speeds of pressure from the Equation (22) and shear wave and $\vec{s}$ and $\vec{t}$ are tangentials of the boundary. The vector $\vec{n}$ is an outer normal vector of the boundary.

2.2.1 A damping preconditioner for the Navier equation

At first it is convenient to introduce some matrices arising from a FEM discretization of the Navier equation (18) and its boundary conditions (24) – (25). Here the matrices introduced in this order are called mass matrix, stiffness matrix and a matrix arising from the absorbing boundary condition (25):

$$\mathbf{M} = \int_\Omega \rho\omega^2 \mathbf{u} \cdot \mathbf{v} \, \mathrm{d}x, \tag{28}$$

$$\mathbf{K} = \int_\Omega \bar{\bar{\sigma}}(\mathbf{u}) : \bar{\bar{\varepsilon}}(\mathbf{v}) \, \mathrm{d}x, \tag{29}$$

$$\mathbf{C} = -\int_{\Gamma_i} \gamma\omega\rho\bar{\bar{B}}\mathbf{u} \cdot \mathbf{v} \, \mathrm{d}s. \tag{30}$$

Using the notation above the discretization of the Navier equation in the matrix form reads:

$$\mathbf{K} + i\mathbf{C} - \mathbf{M} \tag{31}$$

Our hypothesis is that the same kind of a physically damped operator as was used for the Helmholtz equation in the previous section results in an efficient preconditioner also for the Navier equation. Physical damping in an elastic material is added by using a complex Young's modulus, i.e. by multiplying the original Young's modulus by a complex shift $z = \alpha_2 + \beta_2 i$. This leads to a preconditioning operator

$$\mathcal{B} = -\omega^2\rho - z\nabla \cdot \bar{\bar{\sigma}}(\cdot), \tag{32}$$

where α_2 and β_2 are freely chosen parameters. The complex shift z also affects the absorbing boundary condition (25):

$$i\omega\rho\sqrt{z}\mathbf{B}\vec{u} + z\bar{\bar{\sigma}}(\vec{u})\vec{n} = \vec{0} \quad \text{on } \Gamma_i. \tag{33}$$

Using the matrix notation (31) the discretization of the operator (32) produces a damped Navier preconditioner

$$\mathbf{P} = z\mathbf{K} + \sqrt{z}i\mathbf{C} - \mathbf{M}. \tag{34}$$

2.2.2 Controllability method for the Navier equation

As well as for the Helmholtz equation in Section 2.1.3, the exact controllability method can be applied to the Navier equation. The time-harmonic Navier equation (18) with the boundary conditions (24)–(25) transformed back to the corresponding time-dependent wave equation is

$$\rho \frac{\partial^2 \vec{U}}{\partial t^2} - \nabla \cdot \bar{\bar{\sigma}}(\vec{U}) = \vec{0}, \text{ in } Q = \Omega \times (0, T), \tag{35}$$

$$\vec{U} = \vec{G}_d, \text{ on } \gamma_0 = \Gamma_d \times (0, T), \tag{36}$$

$$\rho \bar{\bar{B}} \frac{\partial \vec{U}}{\partial t} + \bar{\bar{\sigma}}(\vec{U})\vec{n} = \vec{G}_i, \text{ on } \gamma_i = \Gamma_i \times (0, T), \tag{37}$$

with the initial conditions

$$\vec{U}(\vec{x}, 0) = \vec{e}_0, \quad \frac{\partial \vec{U}(\vec{x}, 0)}{\partial t} = \vec{e}_1. \tag{38}$$

In the same manner as for the Helmholtz equation in Section 2.1.3, the exact controllability problem (35)–(38) is formulated as a least-squares problem that is solved by the preconditioner Conjugate Gradient algorithm 6. The blocks of the preconditioner $\mathbf{L}$ in this case are of the type

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix}, \quad \mathbf{M} = \begin{bmatrix} \mathbf{M}_{11} & 0 \\ \mathbf{0} & \mathbf{M}_{22} \end{bmatrix}, \tag{39}$$

due to the vector-valued variable $\vec{u} = (u_1, u_2)$. Because of the spectral element discretization the mass matrix $\mathbf{M}$ is naturally diagonal. Since we are solving a vector-valued problem, specific care is needed with the AMG algorithm to keep different displacement components disconnected on coarser levels. A solution is to use an initial graph where the sets of graph nodes corresponding to different types of unknowns are not interconnected and the restriction matrix is built blockwise.

2.2.3 Forming the graph for higher-order FEM or SEM

In this thesis, different kinds of higher-order finite element methods have been used for the discretization of Helmholtz and Navier equations. Damped preconditioning operators for both Helmholtz (2) and Navier equation (18), as well as weak forms of these equations, are discretized using first, second, and third order Lagrange finite elements. In 2-D triangular elements are employed and tetraehedral elements are used in 3-D examples. In Figure 5 a second-order triangular Lagrange element is depicted on the left.

In controllability method investigations for both Helmholtz and Navier equations rather simple geometries in 2-D were studied. Instead of higher-order Lagrangian elements, a so-called Spectral Element Method (SEM) was used to discretize the equations. In spectral elements, nodes of the element are located at the

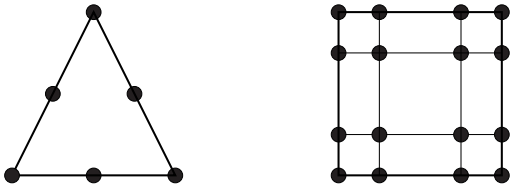

FIGURE 5 A second-order triangular Lagrange element on the left and a third-order spectral element on the right.

Gauss-Lobatto qudrature points. Rectangular spectral elements up to the fifth order were used. In Figure 5 a third-order spectral element is depicted on the right. Higher-order finite element discretization is discussed in general in book [71] by Šolín et al.

In Section 1.1 it was shown how to select coarse graph nodes for the problems discretized with first-order elements. If we were to use the original graph of higher-order finite element discretization to select coarse graph nodes, we would end up in way too coarse grids. This is because the graph would contain connections between every node in a higher-order element. In this case, choosing a coarse node from a higher-order element would eliminate all the other nodes from the element. This is prevented by feeding to the method an initial graph that contains only connections between the closest neighbours of a node. This is analogous to the situation that we would form a finite element grid of first-order elements using the same nodes as in a higher-order element grid.

3 APPLICATION OF THE AMG SOLVER IN CFD

In this chapter we test our graph-based multigrid in computational fluid dynamics (CFD) applications. We will use it as a solver in the numerical simulation of incompressible fluid flows modelled by Stokes and Navier-Stokes equations.

3.1 Stokes equations

Stationary, incompressible and slow fluid flow, strongly dominated by the viscous effects, is modelled by the Stokes equation with suitable boundary conditions on the boundary $\partial\Omega$ of the domain $\Omega \subset \mathbb{R}^d$, $d = 2$ or 3.

$$\begin{cases} -\nu\Delta\vec{u} + \nabla p &= \vec{f} \\ \nabla \cdot \vec{u} &= 0. \end{cases} \qquad \text{in } \Omega, \tag{40}$$

where $\vec{u}$ is the velocity of the fluid, p is the pressure and $\vec{f}$ is the body force. Here the kinematic viscosity ν is assumed to be equal to unity; therefore it is dropped in the further notations.

The boundary $\partial\Omega$ is assumed to be polygonal or polyhedral and it consists of two non-overlapping sets $\partial\Omega = \Gamma_D \cup \Gamma_N$. Here it is assumed that the Dirichlet part of the boundary is non-trivial, as otherwise uniqueness of the velocity solution would not be assured. The boundary conditions associated with equation (40) are

$$\vec{u} = \vec{g} \text{ on } \Gamma_D, \qquad \frac{\partial\vec{u}}{\partial\vec{n}} - \vec{n}p = \vec{0} \text{ on } \Gamma_N, \tag{41}$$

where g is a given function and $\vec{n}(\vec{x})$ is the outer normal vector. The left condition is the Dirichlet boundary condition and the right one is the Neumann boundary condition. In this thesis, on Neumann parts of the boundaries a "natural outflow" boundary condition is used, which is the reason why we have set the Neumann boundary condition equal to $\vec{0}$.

For the finite element discretization we need to introduce a weak formula-

tion for the Stokes equations. We choose function spaces V_g and Q to be

$$V_{\vec{g}} = \left\{ \vec{v} \in H^1(\Omega)^d : \vec{v} = \vec{g} \text{ on } \Gamma_D \right\}, \qquad Q = L^2(\Omega). \tag{42}$$

The space V_0 is chosen as a test function space for the velocity field and a solution for the velocity is sought from the space V_g. For the pressure, both the test function and solution spaces are the same space Q. By multiplying Equation (40) with test functions $\vec{v} \in V_0$ and $q \in Q$, integrating over the domain Ω and taking into account boundary conditions (41), we obtain a weak formulation for the Stokes problem: Find $\vec{u} \in V_g$ and $p \in Q$ such that

$$\begin{cases} \displaystyle\int_\Omega \nabla \vec{u} : \nabla \vec{v} - p \cdot \nabla \vec{v} \, \mathrm{d}x = \int_\Omega \vec{f} \cdot \vec{v} \, \mathrm{d}x \\ \displaystyle\int_\Omega q \nabla \cdot \vec{u} \, \mathrm{d}x = 0. \end{cases} \tag{43}$$

for all $\vec{v} \in V_0$ and $q \in Q$. Furthermore, let us define discrete function spaces

$$\begin{aligned} V_{gh} &= \left\{ \vec{v} \in H^1(\Omega)^d : \vec{v} \mid_K \in P^m, \vec{v} = \vec{g} \text{ on } \Gamma_D \right\}, \\ Q_h &= \left\{ q \in L^2(\Omega) : q \mid_K \in P^n \right\}, \end{aligned} \tag{44}$$

where P^k denotes polynomials of order k or less. Here we have used linear triangular elements in 2D problems and linear tetrahedral elements in 3D problems. That is, the polynomial degrees in discrete function spaces (44) are $m = n = 1$. In order to be able to use the same piecewise linear approximations for both velocity and pressure, some stabilization is needed to prevent oscillating pressure modes. Here we have used the stabilized finite element method for the Stokes problem introduced in [37] by Franca, Hughes and Stenberg. For the piecewise linear approximations of the velocity field the method reduces to the following form. Find $\vec{u}_h \in V_{gh}$ and $p_h \in Q_h$ such that

$$\begin{cases} \displaystyle\int_\Omega \nabla \vec{u_h} : \nabla \vec{v_h} - p \cdot \nabla \vec{v_h} \, \mathrm{d}x = \int_\Omega \vec{f} \cdot \vec{v_h} \, \mathrm{d}x \\ \displaystyle\int_\Omega q_h \nabla \cdot \vec{u_h} \, \mathrm{d}x - \sum_{K \in \mathcal{T}} \int_K \alpha h_k^2 \nabla p_h \cdot \nabla q_h \, \mathrm{d}x = - \sum_{K \in \mathcal{T}} \int_K \alpha h_k^2 \vec{f} \cdot \nabla q_h \, \mathrm{d}x. \end{cases} \tag{45}$$

for all $\vec{v_h} \in V_{0h}$ and $q_h \in Q_h$. Here $\mathcal{T}$ is a non-overlapping partition of the domain Ω into elements K. The stabilization parameter α is a given constant and h_k is the diameter of element K. A choice of the stabilization parameter α to be too large may lead to overstabilization of the Stokes problem (see, for example discussion by Elman et al. in [27]). Aware of this, we remain within the stabilized method (45), for it is very easy to implement and it has a favourable affect on the convergence of iterative solvers.

However, it is worthwhile to mention other possibilities for dealing with the stability issues. In *mixed methods* polynomial approximations of different order for the velocity and pressure are used to form a stable discretization. For example, in *Taylor-Hood* elements (see, for example, [23]), a quadratic approximation

is used for velocity components and a linear one for the pressure. Other stable pairings of polynomial approximations for the Stokes equations can be found for example in the book [27] by Elman et al.

A parameter-free stabilization scheme for the Stokes problem is introduced in [26] by Dohrmann and Bochev. In this method a local projection of the pressure is used to correct the mismatch between the discrete divergence of the velocity field and the discrete pressure space. In this way a stable linear approximation for both the pressure and the velocity field is achieved without introducing any "arbitrary" stabilization parameter.

3.2 Navier-Stokes equations

In order to take into account momentum effects in a stationary incompressible flow, the Navier-Stokes equations are

$$\begin{cases} -\nu\Delta\vec{u} + \rho(\vec{u}\cdot\nabla)\vec{u} + \nabla p = \vec{f} \\ \nabla\cdot\vec{u} = 0 \end{cases} \quad \text{in } \Omega, \tag{46}$$

where ν is the kinematic viscosity of the fluid. We assume that the viscosity and the density ρ of the fluid are given constants. Furthermore, it is assumed that the density is equal to unity so that it is dropped from the following notations. The properties of a particular flow are often related to the Reynolds number, which is defined by $\mathrm{Re} = \rho v l/\nu$, where v and l are the characteristic velocity and length scale of the flow.

The Navier-Stokes equations are nonlinear due to the convective term $(\vec{u}\cdot\nabla)\vec{u}$ and they must be linearized before using our linear solver. We have chosen to use the Picard linearization because it has a larger convergence radius than the Newton linearization. A disadvantage is that the Picard linearization converges only linearly, whereas in the Newton method convergence is quadratic. By the Picard linearization a solution to the nonlinear problem is sought as follows: select $\vec{u}_0$ (e.g. $\vec{u}_0 = 0$) and proceeding from u_i, subsequently solve u_{i+1} and p_{i+1} from the equations

$$\begin{cases} -\nu\Delta\vec{u}_{i+1} + (\vec{u}_i\cdot\nabla)\vec{u}_{i+1} + \nabla p_{i+1} = \vec{f} \\ \nabla\cdot\vec{u}_{i+1} = 0 \end{cases} \quad \text{in } \Omega, \tag{47}$$

until the solution $(\vec{u}_{i+1}, p_{i+1})$ satisfies equations (46) well enough. In practice the iteration may need some underrelaxation. Thus, instead of taking $\vec{u}_{i+1}$ directly as the solution of (47) we assign $\vec{u}_{i+1}$ to be a convex combination of $\vec{u}_{i+1}$ solved from (47) and the previous velocity $\vec{u}_i$. Giving a relaxation parameter $\eta \in [0,1]$ the new velocity approximation is computed by

$$\vec{u}_{i+1} = \eta\vec{u}_{i+1} + (1-\eta)\vec{u}_i.$$

This linear problem is a special case of the so-called Oseen problem where, instead of the old approximation $\vec{u}_i$, there is a given sufficiently smooth convective vector field $\vec{a}$. In this context a pseudo time integration can be used to stabilize the iteration and to make the linear systems easier to solve, as suggested by Wabro in [81]. The pseudo time integration consists of adding the terms

$$\kappa\vec{u}_{i+1}, \quad \kappa\vec{u}_i, \tag{48}$$

to the left and right hand side of the first equation in (47), respectively. Here κ is a suitable (small) parameter which can also be a function of the position, such that the pseudo time integration is used only in some parts of the computational domain.

We choose function spaces V_G and Q as in (42) and obtain a weak formulation for the Oseen (linearized Navier-Stokes) problem: Find $\vec{u} \in V_g$ and $p \in Q$ such that

$$\begin{cases} \displaystyle\int_\Omega \nu\nabla\vec{u} : \nabla\vec{v} + (\vec{a}\cdot\nabla)\vec{u}\cdot\vec{v} - p\nabla\cdot\vec{v}\,\mathrm{d}x = \int_\Omega \vec{f}\cdot\vec{v}\,\mathrm{d}x \\ \displaystyle\int_\Omega q\nabla\cdot\vec{u}\,\mathrm{d}x = 0. \end{cases} \tag{49}$$

for all $\vec{v} \in V_0$ and $q \in Q$. Here $\vec{a}$ denotes some known vector field.

The discretization of the weak Oseen problem is done by the finite element method using equal-order linear approximations for the velocity components and pressure. This approximation is unstable, so we will use a stabilization method introduced in [36] by Franca and Frezzi. For linear approximations of the velocity field, the stabilization method reduces to a streamline-upwind/Petrov-Galerkin (SUPG) type of formulation (described for example in [20] by Brooks and Hughes). The stabilized weak formulation for the Oseen problem reads: Find $\vec{u}_h \in V_{0h}$ and $p_h \in Q_h$ such that

$$\begin{cases} \displaystyle\int_\Omega \nu\nabla\vec{u}_h : \nabla\vec{v}_h + (\vec{a}\cdot\nabla)\vec{u}_h\cdot\vec{v}_h - p_h\nabla\cdot\vec{v}_h + \delta\nabla\cdot\vec{u}_h\nabla\cdot\vec{v}_h\,\mathrm{d}x \\ \qquad \displaystyle + \sum_{K\in\mathcal{T}}\int_K R_O\cdot T_O\,\mathrm{d}x = \int_\Omega \vec{f}\cdot\vec{v}_h\,\mathrm{d}x + \sum_{K\in\mathcal{T}}\int_K \vec{f}\cdot T_O\,\mathrm{d}x, \\ \displaystyle\int_\Omega q_h\nabla\cdot\vec{u}_h - \sum_{K\in\mathcal{T}}\int_K \tau\nabla p_h\cdot\nabla q_h\,\mathrm{d}x = -\sum_{K\in\mathcal{T}}\int_K \tau\vec{f}\cdot\nabla q_h\,\mathrm{d}x \end{cases} \tag{50}$$

holds for all $\vec{v}_h \in V_{0h}$ and $q_h \in Q_h$. In Equation (50) appears many different parameters which are discussed next. At first, the term $\delta\nabla\cdot\vec{u}_h\nabla\cdot\vec{v}_h\,\mathrm{d}x$ is used for the continuity stabilization. In article [36] by Franca et al. the parameter δ is defined by $\delta = \lambda||\vec{a}(\vec{x})||h_K\min(Re_K(\vec{x}),1)$, where $\lambda > 0$ is a positive parameter. In the numerical tests we have chosen $\lambda = 1$. An element-specific Reynolds number $Re_K(\vec{x})$ is defined as

$$Re_K(\vec{x}) = \frac{m_K||\vec{a}(\vec{x})||h_K}{4\nu}, \tag{51}$$

where it is sufficient to choose $m_K = \frac{1}{3}$ when linear interpolations are used. Notations R_O and T_O are used for abbreviating the following stabilization terms:

$$R_O := (\vec{a} \cdot \nabla)\vec{u}_h + \nabla p_h, \quad T_O := \tau((\vec{a} \cdot \nabla)\vec{v}_h - \nabla q_h), \tag{52}$$

where τ is the stabilization parameter

$$\tau = \frac{h_K \min(Re_K(\vec{x}), 1)}{2||\vec{a}(\vec{x})||}. \tag{53}$$

Other stabilization methods for the incompressible Navier-Stokes equations can be found, for example, in [23, 61, 62, 75, 76, 86].

3.3 Adjusting the AMG for incompressible flow problems

For incompressible flow problems we form the initial graph using a finite element mesh. There is a graph node for each type of unknown in each nodal point of the mesh. The graph nodes corresponding to the same type of unknown are connected if the corresponding nodal points in the finite element mesh share at least one common element. Nodes for the coarser level are chosen as describes in Section 1.1. Moreover, we eliminate the degrees of freedom corresponding to Dirichlet type boundary conditions after the coarsening process. This way we do not have to use a 'double coarsening strategy' suggested in [81] by Wabro for avoiding stability problems.

In the numerical experiments smoothers for the AMG are the standard incomplete LU factorization without fill-in (ILU(0)) presented in Algorithm 4 in Section 1.2 and a variant of ILUT decomposition presented in Algorithm 5 in the same section. In smoothing steps we use relaxation. The relaxation parameter is either given by the user for every level of the AMG except for the coarsest level, where a direct solver is used, or an optimal relaxation parameter is calculated to minimize the 2-norm of the residual. In the second case the relaxation parameter is calculated for a discrete problem $\mathbf{Ax} = \mathbf{f}$ as follows: First we set $\mathbf{d}_k = \mathbf{Sr}_k$, where $\mathbf{d}_k$ is the new search direction (as in a minimization procedure), $\mathbf{S}$ corresponds to the action of the smoother and $\mathbf{r}_k$ is the residual in the iteration step k defined by $\mathbf{r}_k = \mathbf{f} - \mathbf{Ax}_k$. The approximate solution is updated by

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \beta_k \mathbf{d}_k, \quad \text{where} \quad \beta_k = \frac{(\mathbf{r}_k, \mathbf{Ad}_k)}{(\mathbf{Ad}_k, \mathbf{Ad}_k)}. \tag{54}$$

Unless stated otherwise, we use W-cycles with one pre- and one post-smooth on any other level except the coarsest. The numbering of the degrees of freedom in the discrete system has some effect on the ILU-smoother of the AMG. Also, it changes the bandwidth, and thus the storage format, of the coarsest system. A resonable requirement is that the degrees of freedom corresponding to a specific nodal point are numbered sequentially.

3.3.1 Stability issues

The stabilization of the pressure in both Stokes and Navier-Stokes problems, as well as of the convection term in Navier-Stokes, calls for elementwise added terms, which have an h-dependent coefficient in front of them. With AMG this causes a problem, since this coefficient transfers to the coarse levels as such, which will impair the stability. As analyzed in [81], the stability of the coarse grid systems can be assured, for example, by postprocessing the corresponding matrix elements in the coarse grid systems. However, when using an unstructured mesh, postprocessing is not straightforward as we do not know how to scale the h-dependent coefficients correctly, because the sizes of the "elements" in coarse levels are unknown. Additionally, we want to construct a method which is as close to a black box algorithm as possible, and therefore we will do no postprocessing. Instead, only a few coarser levels are used such that the stability is not lost. While this is admittedly in contradiction with the multigrid ideology, we will show that the constructed method is still good for practical purposes.

3.4 Numerical examples

3.4.1 A Stokes flow in 2D lid-driven cavity

We start by solving the stabilized Stokes problem in a classic lid-driven cavity setting. The computational domain of the problem is the unit square. On the top of the square we set the velocity field of the flow equal to $(1, 0)$ and it vanishes on the other boundaries. The iteration is continued starting from a zero initial guess until the 2-norm of the residual is below 10^{-6}. The problem is solved on a sequence of uniform meshes. The smoother here is ILU(0).

First, as a comparison, we solve the Stokes problem so that degrees of freedom related to the Dirichlet boundary are eliminated before the coarsening process. Additionally, no relaxation is used in the smoothing steps. Finite element meshes used in discretization are structured and they consist of linear triangular elements. The element sizes vary from $1/32$ to $1/256$ for four different meshes.

The results are shown in Table 1. Starting from the left, the mesh step size h, the size of the discrete system (after eliminating Dirichlet boundary conditions), the discretization time, the number of levels in the AMG and its initialization time, the ILU factorization time, the system solve time, number of iterations and the averaged residual contraction per cycle are shown. The various times are in CPU-seconds.

As can be seen from Table 1 the results are quite discouraging. The AMG solver fails to achieve convergence as the number of levels of AMG increases. Even though we restrict the number of levels to three, the biggest problem does not converge at all and the second biggest problem converges very slowly.

We next solve the same problem as above, but this time we use elimina-

TABLE 1 The Stokes flow in a lid-driven cavity, pre-elimination, no relaxation.

h	size	Discr.	levels	AMG init.	ILU init.	Solve	Iter.	Rate
1/32	3011	0.03 s	2	0.04 s	0.01 s	0.09 s	9	0.154
1/64	12163	0.15 s	3	0.22 s	0.05 s	0.49 s	9	0.158
1/128	48899	–	5	–	–	–	–	–
1/128	48899	–	4	–	–	–	–	–
1/128	48899	0.67 s	3	2.03 s	0.15 s	14.46 s	54	0.770
1/256	196099	–	5	–	–	–	–	–
1/256	196099	–	4	–	–	–	–	–
1/256	196099	–	3	–	–	–	–	–

tion after coarsening discussed in Section 1.1, and optimal relaxation (54) in the smoothing steps of Algorithm 2. Same meshes as above are used. The results are gathered in Table 2, where the explanations of the columns are equivalent to those for the previous table.

TABLE 2 The Stokes flow in a lid-driven cavity with optimal relaxation, elimination after coarsening.

h	size	Discr.	levels	AMG init.	ILU init.	Solve	Iter.	Rate
1/32	3011	0.03 s	2	0.05 s	0.01 s	0.05 s	7	0.094
1/64	12163	0.15 s	3	0.21 s	0.04 s	0.31 s	8	0.102
1/128	48899	0.81 s	4	1.55 s	0.20 s	3.07 s	9	0.151
1/256	196099	–	5	–	–	–	–	–
1/256	196099	2.55 s	4	6.91 s	0.63 s	11.1 s	10	0.196

The results of this experiment are much more positive than those of the previous one, but we can still observe the slowing of the convergence with respect to the number of levels in AMG. There are two explanations for this. First, any multigrid method applying a direct solver for the coarsest problem converges better (also theoretically) for a smaller number of levels. Secondly, the further we coarsen the system the more we have problems with stability. This manifests itself in the largest problem, where the solver fails. The method works well again when the number of coarse levels is smaller.

This phenomenon can be futher confirmed by increasing the stabilization parameter. We choose now the pressure stabilization parameter $\alpha = 1/12$ and compute the corresponding results. Here optimal relaxation is used and elimination is carried out after the coarsening. For the higher stabilization parameter (Table 3), we observe better convergence rates and we do not experience any problems even during the largest simulation.

In Figure 6 is illustrated the memory usage of our AMG solver for meshes used above. The memory usage of the whole running program is shown on the

TABLE 3 The Stokes flow in a lid-driven cavity, $\alpha = 1/12$.

h	size	Discr.	levels	AMG init.	ILU init.	Solve	Iter.	Rate
1/32	3011	0.04 s	2	0.04 s	0.01 s	0.07 s	7	0.008
1/64	12163	0.15 s	3	0.24 s	0.03 s	0.39 s	7	0.078
1/128	48899	0.62 s	4	1.23 s	0.15 s	1.84 s	7	0.077
1/256	196099	2.59 s	5	6.85 s	0.60 s	9.12 s	7	0.117

left side of the figure and on the right is shown the portion of the AMG components, including fine level graph, coarse level matrices and vectors, restriction matrices, ILU and LU factorizations.

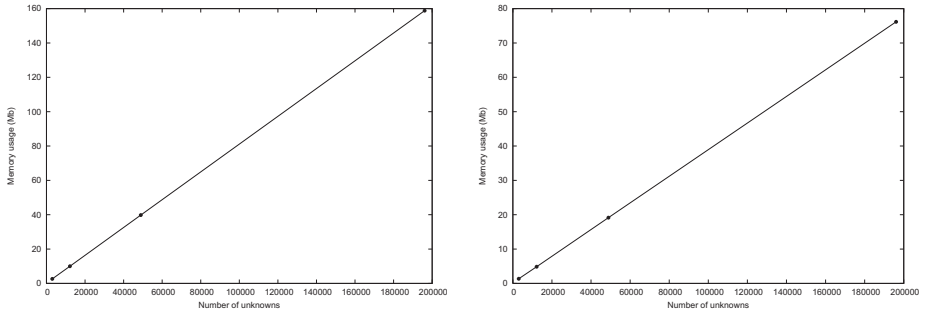

FIGURE 6 Memory usage in Stokes problem in 2D driven cavity on different meshes. Whole running program (left) and AMG components (right).

Finally for this section, we test our algorithm with an unstructured mesh. For comparison we will use the coarsening strategy proposed by Kickinger in the paper [47], in which the selection of coarse nodes is carried out in a straightforward manner according to the numbering of the unknowns. For the test we generated two unstructured meshes with arbitrary numbering of the nodes. The smaller mesh contains 3007 nodes and 5812 triangular elements, while the larger one has 14697 nodes and 28416 elements. The results are shown in Table 4. The labeling of the columns is equivalent to that in the previous tables except for the first column where we mark our coarsening strategy by 'I' and the strategy used in [47] by 'II'. In these cases, slightly better convergence rates can be achieved with our coarsening strategy. With structured meshes both coarsening strategies produce similar coarse levels and, thus, similar convergence rates.

3.4.2 A Navier-Stokes flow in 2D lid-driven cavity

Next we solve the Navier-Stokes equations with the Picard-iteration in the same test setting as in the previous examples. Using the current approximate solution $\vec{u}_k$, p_k of the Picard iteration step k and the Oseen system (50), we can calculate the residual $\mathbf{r}_{N-S}$ of the nonlinear equations. The Picard iteration is continued until

TABLE 4 The Stokes flow in a lid-driven cavity, unstructured mesh

	size	Discr.	levels	AMG init.	ILU init.	Solve	Iter.	Rate
I	8621	0.11 s	3	0.32 s	0.04 s	0.67 s	10	0.183
II	8621	0.11 s	3	0.30 s	0.03 s	0.73 s	11	0.235
I	42139	0.64 s	4	4.99 s	0.17 s	4.85 s	14	0.282
II	42139	0.64 s	4	4.70 s	0.20 s	5.61 s	16	0.339

the residual norm $\|\mathbf{r}_{N-S}\|_2$ is smaller than a prescribed parameter. The presented AMG method is used as an inner iteration and it is stopped when the residual of the Oseen problem $\mathbf{r}_{Os}$ satisfies the following condition:

$$\|\mathbf{r}_{Os}\|_2 \leq 10^{-3}\|\mathbf{r}_{N-S}\|_2.$$

The initial guess for the inner iteration is the current solution $\vec{u}_k$, p_k. The strategy for choosing the relaxation parameter η for the outer Picard iteration is the following. We start with the value $\eta = 1$. If, in some step, the outer iteration diverges, the new parameter is $\eta = \max(0.1, \eta/2)$. In the case of convergence, η is updated to $\min(1.0, 1.1\eta)$. In this way, the problems are usually solved without an intervention from the user.

In the AMG method, ILU(0) is used as a smoother with the optimized relaxation parameter (54). The density of the fluid is $\rho = 1.0$ and the viscosity is $\nu = 2.0 \cdot 10^{-3}$, giving the Reynolds number Re=500. For the outer Picard iteration the initial guess for the velocity is $\vec{u}_0 = 0$ and for the pressure $p_0 = 0$. The nonlinear iteration was stopped when $\|\mathbf{r}_{N-S}\|_2 \leq 10^{-6}$ was reached. The method is tested with respect to the mesh step size. The results are reported in Table 5.

TABLE 5 The Navier-Stokes flow in a lid-driven cavity.

h	size	lvls	P. it.	A. it.	Discr.	A. in.	I. in.	Solve	Total
1/32	3011	2	10	4.0	0.05 s	0.06 s	0.01 s	0.05 s	1.94 s
1/64	12163	3	11	4.0	0.19 s	0.33 s	0.05 s	0.30 s	10.44 s
1/128	48899	4	11	4.0	0.83 s	1.66 s	0.24 s	1.41 s	49.72 s
1/256	196099	5	–	–	–	–	–	–	–
1/256	196099	4	10	3.9	3.27 s	8.57 s	1.01 s	5.74 s	204.26 s

The table shows the mesh step size, the size of the discretized system, the number of levels in the AMG, the number of outer iterations, the average number of inner iterations per outer iteration and the total CPU-time of the program. Solve time refers to time of one outer iteration. Since the structures of the discrete problems are the same as for the Stokes flow, the individual initialization times are comparable to those of the Stokes flow results.

Again, we notice the lack of convergence for the largest problem using five levels in the AMG. This is due to the loss of stability and it can be circumvented by

taking one less coarse level. Otherwise the method behaves well and the number of inner iterations required for each outer iteration is very low.

In the next sequence of tests the effect of viscosity on the method is under consideration. Therefore we set the fluid density $\rho = 1$ and measure the computational times with the viscosity parameter ν ranging from 10^{-1} to 10^{-5}. We are well aware that these simulations are not physically realistic without turbulence modelling and these experiments are performed only to test the method. The setting of the problem is the same as before and the mesh is uniform with $h = 1/128$. The results are reported in Table 6.

TABLE 6 The Navier-Stokes flow in a lid-driven cavity.

ν	P. iter.	A. iter.	Discr.	A. in.	I. in.	Solve	Total
1.0e-1	4	4.50	0.86 s	1.67 s	0.25 s	1.63 s	20.93 s
1.0e-2	7	4.14	0.84 s	1.70 s	0.24 s	1.50 s	33.63 s
1.0e-3	13	4.00	0.85 s	1.71 s	0.25 s	1.46 s	60.01 s
5.0e-4	15	4.27	0.79 s	1.59 s	0.23 s	1.45 s	65.38 s
2.0e-4	17	5.76	0.78 s	1.58 s	0.23 s	2.06 s	83.87 s
1.0e-4	30	8.63	0.79 s	1.59 s	0.23 s	3.23 s	181.55 s
5.0e-5	44	11.66	0.83 s	1.67 s	0.24 s	4.71 s	336.80 s
2.0e-5	80	27.85	0.79 s	1.59 s	0.23 s	11.14 s	1114.15 s
1.0e-5	—	—	—	—	—	—	—

As expected, viscosity affects the convergence of the outer iteration and the solution phase of the AMG inner iteration. The last row of the table reveals that the inner iteration did not converge for $\nu = 10^{-5}$. This problem can be circumvented by using pseudo time integration (48). Table 7 shows the results of the same experimets with the pseudo time integration weight $\kappa = 0.01$.

TABLE 7 The Navier-Stokes flow in a lid-driven cavity, pseudo time integration.

ν	P. iter.	A. iter.	Discr.	A. in.	I. in.	Solve	Total
1.0e-1	4	4.50	0.78 s	1.58 s	0.23 s	1.51 s	19.37 s
1.0e-2	7	4.14	0.82 s	1.59 s	0.23 s	1.40 s	31.68 s
1.0e-3	15	4.00	0.81 s	1.59 s	0.23 s	1.35 s	64.13 s
5.0e-4	17	4.24	0.80 s	1.60 s	0.24 s	1.45 s	74.41 s
2.0e-4	23	6.48	0.84 s	1.66 s	0.24 s	2.47 s	125.92 s
1.0e-4	29	7.72	0.84 s	1.67 s	0.24 s	3.02 s	174.37 s
5.0e-5	55	10.64	0.85 s	1.68 s	0.25 s	4.25 s	396.88 s
2.0e-5	118	20.75	0.83 s	1.65 s	0.24 s	8.50 s	1343.97 s
1.0e-5	210	36.00	0.83 s	1.65 s	0.24 s	14.88 s	3727.18 s

This time the problem with $\nu = 10^{-5}$ could also be solved. However, the

pseudo time integration seems to have a tendency of considerably slowing down the convergence of the outer iteration.

Figure 7 shows scaled velocity vectors near the lower right corner of the computational domain at viscosities ν=1.0e-3, ν=5.0e-5 and ν=2.0e-5. Secondary and tertiary vortices can be observed.

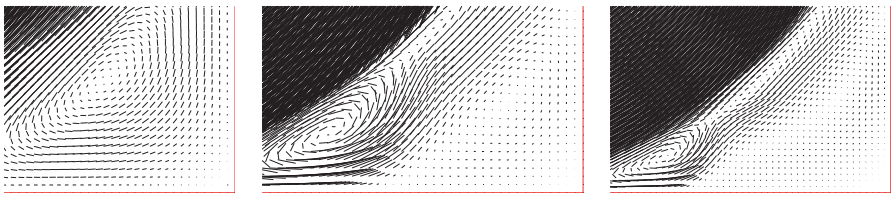

FIGURE 7 Scaled velocity vectors at viscosities ν=1.0e-3, ν=5.0e-5 and ν=2.0e-5.

3.4.3 A 2D backward facing step

Let us study next the well known backward facing step problem setting. Here the computational domain is the union of two rectangles, $]0,2[\times]0.5,1[$ and $]2,16[\times]0,1[$, which depict a two-dimensional 'pipe' with a sudden expansion in the direction of the flow. The velocity field at the inlet has a parabolic distribution $u(0,y) = 16.0(y-0.5)(1.0-y)$.

We solve the problem on an unstructured finite element mesh which consists of 12867 nodal points and 24818 linear triangular elements. The discrete problem has a total of 36811 unknowns. The results are reported in Table 8.

TABLE 8 The Navier-Stokes flow past a backward facing step.

ν	P. iter.	A. iter.	Discr.	A. in.	I. in.	Solve	Total
1.0e-1	5	5.00	0.60 s	1.87 s	0.22 s	1.80 s	25.55 s
1.0e-2	11	5.00	0.60 s	1.87 s	0.23 s	1.78 s	53.14 s
1.0e-3	—	—	—	—	—	—	—

The method diverges already at the viscosity value ν=1.0e-3. The main problem with the method is the lack of robustness in the settings where a large vortex reaches the computational boundary. This is precisely what happens in this test case and in several other tests not reported here. A partial solution to this problem is to use pseudo time integration or to increase the length of the computational domain.

Next we use pseudo time integration in the whole computational domain with the weight parameter $\kappa = 0.1$. The corresponding results are shown in Table 9. The drawback of the pseudo time integration is its negative influence on the convergence of the outer and inner iterations.

Next we use a variant of ILUT by Saad [67] as a smoother of AMG. In ILUT the dropping threshold of the matrix elements is relative to the maximum norm

TABLE 9 The Navier-Stokes flow past a backward facing step.

ν	P. iter.	A. iter.	Discr.	A. in.	I. in.	Solve	Total
1.0e-1	5	5.60	0.61 s	1.78 s	0.22 s	1.99 s	25.96 s
1.0e-2	13	5.38	0.59 s	1.76 s	0.22 s	1.89 s	61.92 s
1.0e-3	138	38.41	0.61 s	1.83 s	0.23 s	15.81 s	2567.83 s
5.0e-4	–	—	—	—	—	—	—

of the particular row of the matrix to be factored. However, since there is a possibility that a diagonal element of the triangular factors vanishes, all diagonal elements smaller than a given constant are replaced with the same constant. Results are shown in Table 10. We can observe better convergence rates when using ILUT as a smoother. A disadvantage is that the computation of the ILUT factorization is much more expensive than the standard ILU(0).

TABLE 10 The Navier-Stokes flow past a backward facing step.

ν	P. iter.	A. iter.	Discr.	A. in.	I. in.	Solve	Total
1.0e-1	5	4.40	0.59 s	1.79 s	0.91 s	1.66 s	27.67 s
1.0e-2	11	4.09	0.58 s	1.78 s	0.91 s	1.53 s	56.34 s
1.0e-3	97	6.81	0.59 s	1.80 s	0.91 s	2.78 s	602.90 s
5.0e-4	224	7.94	0.59 s	1.80 s	0.92 s	3.27 s	1500.28 s

3.4.4 A 3D lid-driven cavity

As the first three-dimensional problem we consider the previously described lid-driven cavity problem. The computational domain is a unit cube $[0,1]^3$ with Dirichlet boundary conditions $\vec{u} = (1,0,0)$ on the top face of the cube, and $\vec{u} = (0,0,0)$ on other faces. For the computations, we have created a structured finite element mesh consisting of 35937 nodal points and 196608 tetrahedra elements. The number of unknowns on the finest level is 125310 and the number of levels in AMG is five. As we can see from the Table 11, the number of outer iterations increases substantially when viscocity decreases, but the number of inner iterations remains reasonably small.

3.4.5 A 3D square pipe with cylindrical obstacle

In this example we simulate the flow in a pipe with a square cross section in which a cylindrical obstacle has been introduced. The exact domain for the computations is

$$]0,2[\times]0,1[^2 \backslash B(0.75,0.5,0.2)\times]0,1[,$$

TABLE 11 The Navier-Stokes in 3D lid-driven cavity.

ν	P. iter.	A. iter.	Discr.	A. in.	I. in.	Solve	Total
1.0e-1	4	5.00	14.93 s	37.20 s	3.41 s	9.23 s	313.72 s
1.0e-2	10	5.50	15.11 s	37.37 s	3.45 s	10.36 s	721.11 s
5.0e-3	21	5.86	15.77 s	38.84 s	3.59 s	11.60 s	1532.60 s
2.0e-3	61	7.30	15.61 s	38.38 s	3.54 s	14.64 s	4490.53 s
1.0e-3	254	8.92	15.62 s	38.30 s	3.54 s	18.25 s	19429.60 s

where $B(0.75, 0.5, 0.2)$ represents a ball on the x-y –plane at $(0.75, 0.5)$ with a radius 0.2. The computational domain is illustrated in Figure 8.

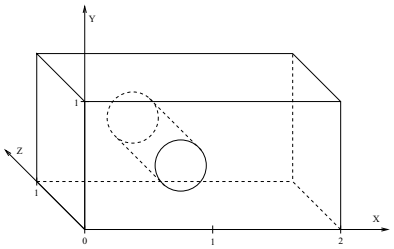

FIGURE 8 Pipe and a cylindrical obstacle.

The flow comes into the domain from the left boundary ($x = 0$) with a biquadratic profile $\vec{u} = 16.0y(1.0 - y)z(1.0 - z)$ and leaves it at the right boundary ($x = 2$). An unstructured finite element mesh has been generated for the geometry having 18672 nodal points and 94304 tetrahedra. The results are gathered in Table 12. As the results show, our method is still capable of convergence at the viscosity value $\nu = 1.0e - 3$.

TABLE 12 The Navier-Stokes flow past a cylindrical obstacle.

ν	P. iter.	A. iter.	Discr.	A. in.	I. in.	Solve	Total
1.0e-1	5	5.60	9.93 s	28.03 s	2.65 s	7.05 s	278.67 s
1.0e-2	8	6.25	10.20 s	28.79 s	2.73 s	8.26 s	443.50 s
5.0e-3	11	8.64	10.02 s	28.35 s	2.69 s	11.67 s	625.70 s
2.0e-3	19	10.05	10.10 s	28.43 s	2.69 s	13.86 s	1106.20 s
1.0e-3	42	10.67	10.06 s	28.26 s	2.68 s	14.77 s	2449.26 s

With these test settings, the finest level system matrix has about 4 million nonzero elements. The running program uses approximately 163 megabytes of memory and 56% of this is reserved for the AMG components, such as fine level graph, coarse level matrices and vectors, restriction matrices, ILU and LU factorizations.

4 CONCLUSIONS

In this thesis a graph-based multigrid method for solving large and sparse linear equations was studied. The main goal was to construct a simple and easy to implement, yet efficient, algebraic multigrid method to be used in the numerical simulation of different physical phenomena.

The starting point for the study was a graph-based multigrid introduced by Kickinger in [47]. It was improved by processing the Dirichlet boundary conditions after the whole multigrid hierarchy was constructed. Additionally, the method was modified so that any graph related to the problem could be fed into the coarsening process. In practice, the graph is extracted from the mesh used in the discretization or from the system matrix. Along with this modification we gained more flexibility in our method at the cost of losing a pure black box feature that is often associated with the AMG methods. The fast multiplication of sparse matrices introduced in Section 1.1, along with the straightforward graph-based coarsening, enabled a rapid computation of coarse levels.

In Chapter 2 we considered our method as a preconditioner for linear time-harmonic wave equations. The propagation of acoustic waves was modelled by the Helmholtz equation and that of elastic waves by the Navier equations. We studied a physical damping preconditioner for both Helmholtz and Navier equations. The numerical results in papers [PII] and [PIV] show that our method is suitable for approximating the inverse of the damped preconditioner with higher-order discretizations in complicated domains. Especially for low-frequency and mid-frequency Helmholtz problems our approach is well suited. In high-frequency Helmholtz problems a doubling of GMRES iterations was observed.

Another approach for solving wave scattering problems using control techniques was studied in Section 2.1.3 for the Helmholtz equation and in Section 2.2.2 for the Navier equation. In this case a time-harmonic wave equation was represented as an exact controllability problem for the time-dependent equation. The exact controllability problem was formulated as a least squares optimization problem whose solution was sought using the preconditioned conjugate gradient method. Here we used our multigrid method to compute the preconditioning step of the CG algorithm. Higher-order spectral methods were used to discretize

the equations. Numerical tests in paper [PI] show that the use of the AMG preconditioner in acoustic scattering problems keeps the number of CG iterations independent of the order of the spectral element basis, which confirms that the proposed method is suitable for higher orders. The percentage of overall computational costs remained very low for the AMG preconditioner. For the Navier equation, proportion of the AMG preconditioner of computational costs also remained very low but more fine-tuning was needed to keep the number of AMG iterations low. For instance, the relaxation parameter of the Successive Over Relaxation (SOR) smoother had to be adjusted according to the order of discretization, as can be seen from the numerical results in article [PIV].

In Chapter 3 we applied our multigrid method as a solver for applications in computational fluid dynamics. An incompressible fluid flow was modelled with both Stokes and Navier-Stokes equations. In both cases the discretization was done by the stabilized finite element method. In the numerical tests in Section 3.4 we showed that it is necessary to process the Dirichlet boundary conditions after the coarse level matrices are computed. Furthermore, the calculation of the optimal relaxation of the ILU smoother was discovered to be an efficient way to reduce the number of AMG iterations. Overall, taking into account that our method is constructed from quite simple algorithms, it was shown that the method is a quite efficient solver even for flows with the Reynolds number of order 10^4.

One improvement for solving the Navier-Stokes equations would be to use Newton-Raphson linearization instead of that of Picard. It has quadratic convergence but it needs a better initial guess than Picard linearization. An option would be to use one or two steps of Picard iteration and then start the Newton iteration. In both Newton and Picard linearizations the stiffness matrix must be computed again at the beginning of the iteration. This could be avoided by using so called quasi-Newton methods that are widely used in numerical optimization (see, for instance [59], a book by Nocedal and Wright).

Altogether, the graph-based multigrid method studied here was shown to be a suitable solver and preconditioner for quite a wide range of physical applications. However, some work remains to be done. Now, when there are dozens of cores even in a single processor, parallelization and scalability of algorithms are more and more of great importance. Parallelization of the solution phase is quite straightforward, especially when the smoother is parallel by construct. For example, in the Jacobi method the components of the new iteration may be computed simultaneously and, thus, it is naturally parallel. For ILU-type smoothers, an alternative is to use graph-coloring techniques (see, for example, [68] by Saad) to color adjacent nodes with different colors. Then, unknowns of the same color can be determined simultaneously. This approach is quite attrative to us since the adjacency graph of the unknowns is already at hand. A more challenging task is to parallelize the setup phase of AMG that is inherently sequential. Many approaches to make the setup phase parallel can already be found from the literature (see, for example [40] by Griebel et al. and references therein). For us, this is the next step to be taken to improve our algorithm.

REFERENCES

[1] T. Airaksinen and S. Mönkölä. Comparison between the shifted-Laplacian preconditioning and the controllability methods for computational acoustics. *Journal of Computational and Applied Mathematics*, 234(6):1796–1802, 2009.

[2] R.E. Alcouffe, A. Brandt, J.E. Dendy, and J.W. Painter. The multi-grid method for the diffusion equation with strongly discontinuous coefficients. *SIAM J. Sci. Statist. Comput.*, 2:430–454, 1981.

[3] O. Axelsson and P.S. Vassilevski. Algebraic multilevel preconditioning methods. I. *Numerische Mathematik*, 56(2–3):157–177, 1989.

[4] O. Axelsson and P.S. Vassilevski. Algebraic multilevel preconditioning methods. II. *SIAM Journal on Numerical Analysis*, 27(6):1569–1590, 1990.

[5] I. Babuska and B.Q. Guo. The h, p and h-p version of the finite element method: basis theory and applications. *Advances in Engineering Software*, 15(3–4), 1992.

[6] R.E. Bank and C. Wagner. Multilevel ILU decomposition. *Numerische Mathematik*, 82(4):543–576, 1997.

[7] R. Beck. Graph-based algebraic multigrid for Lagrange-type finite elements on simplicial meshes. Preprint SC 99-22, Konrad-Zuse-Zentrum fur Informationstechnik, Berlin, 1999.

[8] P. Bettess. *Infinite elements*. Penshaw Press, 1992.

[9] D. Braess. Towards algebraic multigrid for elliptic problems of second order. *Computing*, 55(4):379–393, 1995.

[10] A. Brandt. Multi–level adaptive technique (MLAT) for fast numerical solution to boundary value problems. In H. Cabannes and R. Teman, editors, *Proceedings of the Third International Conference on Numerical Methods in Fluid Mechanics*, volume 18 of *Lecture Notes in Physics*, pages 82–89, Berlin, 1973. Springer–Verlag.

[11] A. Brandt. Algebraic multigrid theory: the symmetric case. *Appl. Math. Comput.*, 19:23–56, 1986.

[12] A. Brandt, S.F. McCormick, and J. Ruge. Algebraic multigrid (AMG) for automatic multigrid solution with application to geodetic computations. Technical report, Institute for Computational Studies, Fort Collins, Colorado, 1982.

[13] A Brandt, S.F. McCormick, and J. Ruge. Algebraic multigrid (AMG) for sparse matrix equations. In D.J. Evans, editor, *Sparsity and its Applications*, pages 257–284. Cambridge University Press, Cambridge, 1984.

[14] M. Brezina, A.J. Cleary, R.D. Falgout, V.E. Henson, J.E. Jones, T.A. Manteuffel, S.F. McCormick, and J.W. Ruge. Algebraic multigrid based on element interpolation (AMGe). *SIAM J. Sci. Comput.*, 22(5):1570–1592 (electronic), 2000.

[15] M. Brezina, R. Falgout, S. MacLachlan, T. Manteuffel, S.F. McCormick, and J.W. Ruge. Adaptive smoothed aggregation (αSA). *SIAM J. Sci. Comput.*, 25(6):1896–1920, 2004.

[16] M. Brezina, R. Falgout, S. MacLachlan, T. Manteuffel, S.F. McCormick, and J.W. Ruge. Adaptive algebraic multigrid. *SIAM J. Sci. Comput.*, 27(4):1261–1286 (electronic), 2006.

[17] M. O. Bristeau, R. Glowinski, and J. Périaux. Using exact controllability to solve the Helmholtz equation at high wave numbers. In *Second International Conference on Mathematical and Numerical Aspects of Wave Propagation (Newark, DE,1993)*, pages 113–127. SIAM, Philadelphia, PA, 1993.

[18] M. O. Bristeau, R. Glowinski, and J. Périaux. On the numerical solution of the Helmholtz equation at large wave numbers using exact controllability methods. Application to scattering. In *Domain decomposition methods in science and engineering (Como, 1992)*, volume 157 of *Contemp. Math.*, pages 399–419. Amer. Math. Soc., Providence, RI, 1994.

[19] M. O. Bristeau, R. Glowinski, and J. Périaux. Controllability methods for the computation of time-periodic solutions; application to scattering. *J. Comput. Phys.*, 147(2):265–292, 1998.

[20] A.N. Brooks and T.J.R. Hughes. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Comput. Methods Appl. Mech. Engrg.*, 32:199–259, 1982.

[21] P.G. Ciarlet. *The Finite Element Method for Elliptic Problems*. North-Holland, New York, 1978.

[22] A.J. Cleary, R.D. Falgout, V.E. Henson, J.E. Jones, T.A. Manteuffel, G.N. Miranda, J.W. Ruge, and S.F. McCormick. Robustness and scalability of algebraic multigrid. *SIAM J. Sci. Comput*, 21:1886–1908, 1998.

[23] C. Cuvelier, A. Segal, and A.A. van Steenhoven. *Finite Element Methods and Navier-Stokes Equations*. D. Reidel Publishing Company, 1986.

[24] L. Demkowicz, J. Kurtz, D. Pardo, W. Rachowicz, M. Paszynski, and A. Zdunek. *Computing with hp-Adaptive Finite Elements*. CRC Press, 2007.

[25] J. E. Dendy Jr. Black box multigrid. *J. Comput. Phys.*, 48:366–386, 1982.

[26] C. Dohrmann and P. Bochev. A stabilized finite element method for the Stokes problem based on polynomial pressure projections. *Int. J. Numer. Methods Fluids*, 46:183–201, 2004.

[27] H.C. Elman, D.J. Silvester, and A.J. Wathen. *Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics*. Oxford University Press, 2005.

[28] B. Enquist and A. Majda. Absorbing boundary conditions for the numerical simulation of waves. *Math. Comp.*, 31:629–651, 1977.

[29] Y. A. Erlangga. Advances in iterative methods and preconditioners for the Helmholtz equation. *Arch. Comput. Methods Eng.*, 15(1):37–66, 2008.

[30] Y. A. Erlangga, C. W. Oosterlee, and C. Vuik. A novel multigrid based preconditioner for heterogeneous helmholtz problems. *SIAM J. Sci. Comput.*, 27(4):1471–1492, 2006.

[31] R. Eymard, T. Galloüet, and R. Herbin. Finite Volume Methods. In P.G. Ciarlet and J.L. Lions, editors, *Handbook of Numerical Analysis*, volume VII, pages 713–1020. Elsevier, 2000.

[32] R.D. Falgout and P.S. Vassilevski. On generalizing the algebraic multigrid framework. *SIAM J. Numer. Anal.*, 42(4):1669–1693, 2004.

[33] R.P. Fedorenko. A relaxation method for solving elliptic difference equations. *USSR Comput. Math. Math. Phys.*, 1(1092), 1961.

[34] R.P. Fedorenko. The speed of convergence of one iterative process. *USSR Comput. Math. Math. Phys.*, 4(227), 1964.

[35] G.E. Forsythe, M.A. Malcolm, and C.B. Moler. *Computer Method for Mathematical Computations*. Prentice–Hall, 1977.

[36] L.P. Franca and S.L. Frey. Stabilized finite element methods. II. The incompressible Navier-Stokes equations. *Comput. Methods Appl. Mech. Engrg.*, 99(2-3):209–233, 1992.

[37] L.P. Franca, T.J.R. Hughes, and R. Stenberg. Stabilized finite element methods for the stokes problem. In M. Gunzburger and R.A. Nicolaides, editors, *Incompressible Computational Fluid Dynamics*, pages 87–107. Cambridge University Press, 1993.

[38] K. Gerdes. A review of infinite element methods for exterior helmholtz problems. *Journal of Computational Acoustics*, 8(1):43–62, 2000.

[39] R. Glowinski and J.L. Lions. Exact and approximate controllability for distributed parameter systems. *Acta Numerica*, 3(-1):269–378, 1994.

[40] M. Griebel, B. Metsch, D. Oeltz, and M.A. Schweitzer. Coarse grid classification: a parallele coarsening scheme for algebraic multigrid methods. *Numerical Linear Algebra with Applications*, 13:193–214, 2006.

[41] W. Hackbusch. Ein iteratives Verfahren zur schnellen Auflösung elliptischer Randwertprobleme. Technical Report 76-12, Universität Köln, 1976.

[42] W. Hackbusch. *Multigrid Methods and Applications*. Springer-Verlag, Berlin, Germany, 1985.

[43] M.R. Hestenes and Stiefel E.L. Methods of conjugate gradients for solving linear systems. *J. Res. Natl. Bur. Stand.*, 49(409), 1952.

[44] F. Ihlenburg. *Finite Element Analysis of Acoustic Scattering*, volume 132 of *Applied Mathematical Sciences*. Springer-Verlag, New York, 1998.

[45] F. Ihlenburg and I. Babuška. Finite element solution of the Helmholtz equation with high wave number part I: The *h*-version of the FEM. *Comput. Mat. Appl.*, 30(9):9–37, 1995.

[46] F. Ihlenburg and I. Babuška. Finite element solution of the Helmholtz equation with high wave number part II: The $h - p$-version of the FEM. *SIAM Journal on Numerical Analysis*, 34(1):315–358, 1997.

[47] F. Kickinger. Algebraic multi-grid for discrete elliptic second-order problems. In *Multigrid methods V (Stuttgart, 1996)*, pages 157–172. Springer, Berlin, 1998.

[48] C. Lanczos. Solutions of systems of linear equations by minimized iterations. *J. Res. Natl. Bur. Stand.*, 49(33), 1952.

[49] L.D. Landau and E.M. Lifshitz. *Theory of Elasticity*. Pergamon Press, Oxford, 1975.

[50] J.L. Lions. Exact controllability, stabilization and perturbations for distributed systems. *SIAM Review*, 30(1):1–68, 1988.

[51] Y. Maday and A.T. Patera. Spectral element method for the incompressible Navier-Stokes equations. In A.K. Noor and J.T. Oden, editors, *State-of-the-art Surveys on Computational Mechanics*, pages 71–143. American Society of Mechanicaal Engineering, New York, 1989.

[52] Y. Maday and J.T. Patera. Spectral element methods for the incompressible Navier-Stokes equations. In A.K. Noor and J.T. Oden, editors, *State-of-the-art Surveys on Computational Mechanics*, pages 71–143. American Society of Mechanical Engineering, 1989.

[53] J. Martikainen. Numerical study of two sparse AMG-methods. *Math. Model. Numer. Anal.*, 37(1):133–142, 2003.

[54] J. Martikainen, A. Pennanen, and T. Rossi. Application of an algebraic multigrid method to incompressible flow problems. Reports of the Department of Mathematical Information Technology, Series B. Scientific Computing, 2006.

[55] S.F. McCormick, editor. *Multigrid Methods*. SIAM, Philadelphia, Pennsylvania, 1987.

[56] G.A. Meurant. Numerical experiments with algebraic multilevel preconditioners. *Elect. Trans. Numer. Anal.*, 12:1–65, 2001.

[57] K.W. Morton and D.F. Mayers. *Numerical Solution of Partial Differential Equations: An Introduction*. Cambridge University Press, 2 edition, 2005.

[58] S. Mönkölä. *Spectral Element Method and Controllability Approach for Time-Harmonic Wave Propagation*. Licentiate thesis, University of Jyväskylä, Jyväskylä, Finland, 2008.

[59] J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer, 1999.

[60] A Nägel, R.D. Falgout, and G. Wittum. Filtering algebraic multigrid and adaptive strategies. *Computing and Visualization in Science*, 11(3):159–167, 2008.

[61] S. Nägele and G. Wittum. On the influence of different stabilisation methods for the incompressible navier-stokes equations. *Journal of Computational Physics*, 224(1):100–116, 2007.

[62] M.A. Olshanskii. A low order Galerkin finite element method for the Navier–Stokes equations of steady incompressible flow: a stabilization issue and iterative methods. *Comput. Methods Appl. Mech. Engrg.*, 191:5515–5536, 2002.

[63] A.T. Patera. Spectral element method for fluid dynamics: laminar flow in a channel expansion. *Journal of Computational Physics*, 15:468–488, 1984.

[64] D. Pathria and G.E. Karniadakis. Spectral element methods for elliptic problems in nonsmooth domains. *Journal of Computational Physics*, 122(1):83–95, 1995.

[65] C. Pozrikidis. *A Practical Guide to Boundary element methods with the software library BEMLIB*. CRC Press, 2002.

[66] J.W. Ruge and K. Stüben. Algebraic Multigrid (AMG). In S.F. McCormick, editor, *Multigrid Methods*, Frontiers in Applied Mathematics, pages 73–130. SIAM, Philadelphia, Pennsylvania, 1987.

[67] Y. Saad. ILUT: a dual threshold incomplete *LU* factorization. *Numer. Linear Algebra Appl.*, 1(4):387–402, 1994.

[68] Y. Saad. *Iterative methods for sparse linear systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, second edition, 2003.

[69] Y. Saad and M.H. Schultz. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Statist. Comput.*, 7(3):856–869, 1986.

[70] C.E. Shannon. Communication in the presence of noise. *Proc. Institute of Radio Engineers*, 37(1):10–21, 1949.

[71] P. Šolín, K. Segeth, and I. Doležel. *Higher-order finite element methods*. Studies in Advanced Mathematics. Chapman & Hall/CRC, Boca Raton, FL, 2004. With 1 CD-ROM (Windows, Macintosh, UNIX and LINUX).

[72] A. Sommerfeld. Die Greensche Funktion der Schwingungsgleichung. *Jahresber. Deutsch. Math.*, 21:309–353, 1912.

[73] K. Stüben. Algebraic multigrid (AMG): Experiences and comparisons. *Appl. Math. Comput.*, 13:419–452, 1983.

[74] K. Stüben. A review of algebraic multigrid. *J. Comput. Appl. Math.*, 128(1-2):281–309, 2001.

[75] T.E. Tezduyar, S. Mittal, S.E. Ray, and R. Shih. Incompressible flow computations with stabilized bilinear and linear equal-order-interpolation velocity-pressure elements. *Comput. Methods Applied Mech. and Engrg.*, 95:221–242, 1992.

[76] T.E. Tezduyar and Y. Osawa. Finite element stabilization parameters computed from element matrices and vectors. *Comput. Methods Applied Mech. and Engrg.*, 190:411–430, 2000.

[77] L.L. Thompson. A review of finite-element methods for time-harmonic acoustics. *Journal of the Acoustical Society of America*, 119(3):1315–1330, 2006.

[78] U. Trottenberg, C. Oosterlee, and A. Schüller. *Multigrid*. Academic Press, San Diego, 2001.

[79] U. Trottenberg, C. Oosterlee, and A. Schüller. *Multigrid*. Academic Press, San Diego, 2001. Appendix A: An introduction to algebraic multigrid by K. Stüben, pages 413–532.

[80] H.A. van der Vorst. Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM J. Sci Statis. Comput.*, 13(2):631–644, 1992.

[81] M. Wabro. Coupled algebraic multigrid methods for the Oseen problem. *Comput. Visual. Sci.*, 7(3–4):141–151, 2004.

[82] C. Wagner. Introduction to algebraic multigrid. Technical report, Interdisziplinäres Zentrum für wissenschaftliches Rechnen, Universität Heidelberg, 1999.

[83] C. Wagner. On the algebraic construction of multilevel transfer operators. *Computing*, 65(1):73–95, 1999.

[84] W. L. Wan, T.F. Chan, and B. Smith. An energy-minimizing interpolation for robust multigrid methods. *SIAM J. Sci. Comput.*, 21(4):1632–1649, 2000.

[85] W.L. Wan. An energy-minimizing interpolation for multigrid methods. Technical report, Stanford University, Juli, 1997.

[86] C.H. Whiting and K.E. Jansen. A stabilized finite element method for the incompressible navier–stokes equations using a hierarchical basis. *International Journal for Numerical Methods in Fluids*, 35(1):93–116, 2001.

[87] G. Wittum. On the robustness of ILU smoothing. *SIAM Journal on Scientific and Statistical Computing*, 10(4):699–717, 1989.

[88] O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu. *The finite element method: its basis and fundamentals*. Butterworth-Heinemann, 2005.

YHTEENVETO (FINNISH SUMMARY)

Tässä väitöskirjassa tarkastellaan graafipohjaista monihilamenetelmää, joka on suurten, lineaaristen ja harvojen yhtälöryhminen ratkaisemiseen käytettävä iteratiivinen ratkaisumenetelmä. Väitöskirjan suomenkielinen otsikko on "Graafipohjainen monihilamenetelmä sovelluksineen". Väitöskirja koostuu johdanto-osiosta ja neljästä kansainvälisestä, vertaisarvioidusta lehtiartikkelista. Johdanto-osiossa esitellään monihilamenetelmä sekä eri sovellusalueet, joihin monihilamenetelmää on sovellettu tässä tutkimustyössä. Tutkimustyön päämääränä on ollut kehittää yksinkertaisista ja helposti toteutettavista algoritmeista koostuva monihilamenetelmä, joka kuitenkin soveltuu käytettäväksi hyvinkin erilaisten fysikaalisten ilmiöiden numeerisessa simuloinnissa.

Fysikaalisten ilmiöiden matemaattiset mallit ovat usein osittaisdifferentiaaliyhtälöitä (ODY), joiden tarkkaa ratkaisua ei yleensä tunneta kuin yksinkertaisissa erikoistapauksissa. Yhtälön ratkaisulle on kuitenkin mahdollista etsiä numeerinen likiarvo tietokonetta ja numeerisia menetelmiä apuna käyttäen. ODY:t on kuitenkin diskretisoitava, ennen kuin niitä voidaan ratkoa tietokoneella.

Osittaisdifferentiaaliyhtälöiden diskretointimenetelmiä on useita. Tässä väitöskirjassa käytettävät diskretointimenetelmät ovat elementtimenetelmä ja spektraalielementtimenetelmä. Muita yleisiä menetelmiä ovat esimerkiksi differenssimenetelmä ja äärellisen tilavuuden menetelmä. Yhteinen piirre eri diskretointimenetelmille on se, että ne tuottavat suuren yhtälöryhmän, jossa voi olla jopa miljardeja tuntemattomia ratkaistavana. Tällaisen yhtälöryhmän ratkaiseminen on työlästä perinteisillä ratkaisumenetelmillä. Esimerkiksi LU-hajotelman muodostaminen tällaiselle suurelle ja harvalle matriisille on erittäin raskas ja tietokoneen muistiresursseja runsaasti kuluttava operaatio, etenkin jos taustalla oleva ilmiö mallinnetaan monimutkaisessa, kolmiulotteisessa geometriassa. Toisaalta taas yksinkertaisten iteratiivisten ratkaisumenetelmien, kuten vaikkapa Jacobin-menetelmän, konvergenssi voi hidastua merkittävästi, jolloin riittävän tarkan likiarvon saavuttaminen voi kestää kohtuuttoman pitkään.

Eräitä tehokkaimpia ratkaisumenetelmiä tällä hetkellä ovat niin kutsutut monihilamenetelmät, jotka jakautuvat algebrallisiin ja geometrisiin monihilamenetelmiin. Monihilamenetelmien tehokkuus perustuu perinteisten iteratiivisten menetelmien virhettä silottavaan ominaisuuteen ja harvemmalla hilalla tai tasolla laskettavaan approksimaation korjaukseen. Geometrisissä monihilamenetelmissä nämä harvemmat hilat ja siirto-operaattorit hilojen välillä muodostetaan eksplisiittisesti, kun taas algebrallisissa menetelmissä harvemmat tasot ja operaattorit muodostetaan automaattisesti kerroinmatriisin avulla.

Tässä väitöskirjassa esiteltävä monihilamenetelmä pohjautuu F. Kickingerin esittelemään menetelmään. Kickingerin menetelmässä algebrallisen monihilamenetelmän (AMG) harvempien tasojen muuttujat valitaan kerroinmatriisin naapuruusgraafin avulla. Graafin käytöstä saadaan se etu, että harvan tason muuttujien valinta on nopeaa ja suoraviivaista. Tässä työssä on paranneltu Kickingerin menetelmää. Tärkein parannus on Dirichlet'n reunaehtojen käsittely, toisin sa-

noen niiden eliminointi yhtälöryhmästä vasta kun kaikki harvempien tasojen matriisit on muodostettu. Näin on saatu huomattavasti lisää vakautta erilaisten yhtälöryhmien ratkaisemiseen. Lisäksi menetelmää on muunnettu siten, että käytettävän graafin ei välttämättä tarvitse olla matriisin naapuruusgraafi, vaan menetelmään on mahdollista syöttää jokin muukin ratkaistavaan tehtävään liittyvä graafi. Käytännössä tarvittavat graafit saadaan tehtävään liittyvästä elementtiverkosta. Näin on saatu lisää joustavuutta menetelmään, mutta samalla on vähennetty ratkaisumenetelmän niin kutsuttua musta laatikko -ominaisuutta, joka yleensä liitetään algebrallisiin monihilamenetelmiin. Lisäksi on esitelty nopea tapa laskea harvan tason systeemimatriisi. Kaksi viimeksi mainittua ominaisuutta nopeuttavat harvempien tasojen muodostamista, joka on työläin operaatio algebrallisissa monihilamenetelmissä.

Graafipohjaista monihilamenetelmää on tässä väitöskirjassa käytetty aikaharmonisten aaltoyhtälöiden ratkaisemisessa ja laskennallisessa virtausdynamiikassa. Akustisten ja elastisten aaltojen etenemistä kuvaavavien Helmholtzin ja Navierin yhtälöiden ratkaisemiseen on käytetty kahdenlaista lähestymistapaa. Ensimmäisessä lähestymistavassa on käytetty vaimennettua pohjustinta helpottamaan aikaharmonisten aaltoyhtälöiden ratkaisemista. Pohjustinmatriisi on muodostettu diskretoimalla aaltoyhtälö, johon on lisätty keinotekoista vaimennusta. Näin saatava pohjustettu yhtälöryhmä on ratkaistu iteratiivisella GMRES-menetelmällä, jossa pohjustinmatriisin käänteismatriisia approksimoidaan yhdellä monihila-iteraatiolla.

Toisessa lähestymistavassa aikaharmonisten aaltoyhtälöiden ratkaisemiseen käytetään säätöteoriaan perustuvaa tekniikkaa. Aikaharmoninen aaltoyhtälö muunnetaan ajasta riippuvaksi tarkan säädettävyyden tehtäväksi, joka edelleen muotoillaan pienimmän neliösumman optimointitehtäväksi. Optimointitehtävä ratkaistaan käyttäen pohjustettua liittogradienttimenetelmää. Pohjustin on diagonaalinen lohkopohjustin, jonka muodostamisessa syntyy diskretoitujen aaltoyhtälöiden jäykkyysmatriisin sisältävät yhtälöryhmät. Nämä yhtälöryhmät ratkaistaan graafipohjaisella monihilamenetelmällä. Numeeriset tulokset artikkeleissa [PI]–[PIV] osoittavat, että monihilamenetelmä on tehokas pohjustin molemmissa lähestymistavoissa.

Lisäksi tässä väitöskirjassa on tarkasteltu monihilamenetelmän toimintaa virtausta kuvaavien Stokesin ja Navier-Stokesin yhtälöiden diskretoinnista syntyvien yhtälöryhmien ratkaisumenetelmänä. Numeeriset tulokset luvussa 3.4 osoittavat, että esitelty menetelmä kykenee ratkaisemaan virtaustehtäviä, joissa Reynoldsin luku on jopa kertaluokkaa 10^4. Näin ollen menetelmämme soveltuu hyvin jopa erityisen vaativien yhtälöryhmien ratkaisemiseen.

JYVÄSKYLÄ STUDIES IN COMPUTING

1 Ropponen, Janne, Software risk management - foundations, principles and empirical findings. 273 p. Yhteenveto 1 p. 1999.
2 Kuzmin, Dmitri, Numerical simulation of reactive bubbly flows. 110 p. Yhteenveto 1 p. 1999.
3 Karsten, Helena, Weaving tapestry: collaborative information technology and organisational change. 266 p. Yhteenveto 3 p. 2000.
4 Koskinen, Jussi, Automated transient hypertext support for software maintenance. 98 p. (250 p.) Yhteenveto 1 p. 2000.
5 Ristaniemi, Tapani, Synchronization and blind signal processing in CDMA systems. - Synkronointi ja sokea signaalinkäsittely CDMA järjestelmässä. 112 p. Yhteenveto 1 p. 2000.
6 Laitinen, Mika, Mathematical modelling of conductive-radiative heat transfer. 20 p. (108 p.) Yhteenveto 1 p. 2000.
7 Koskinen, Minna, Process metamodelling. Conceptual foundations and application. 213 p. Yhteenveto 1 p. 2000.
8 Smolianski, Anton, Numerical modeling of two-fluid interfacial flows. 109 p. Yhteenveto 1 p. 2001.
9 Nahar, Nazmun, Information technology supported technology transfer process. A multi-site case study of high-tech enterprises. 377 p. Yhteenveto 3 p. 2001.
10 Fomin, Vladislav V., The process of standard making. The case of cellular mobile telephony. - Standardin kehittämisen prosessi. Tapaustutkimus solukkoverkkoon perustuvasta matkapuhelintekniikasta. 107 p. (208 p.) Yhteenveto 1 p. 2001.
11 Päivärinta, Tero, A genre-based approach to developing electronic document management in the organization. 190 p. Yhteenveto 1 p. 2001.
12 Häkkinen, Erkki, Design, implementation and evaluation of neural data analysis environment. 229 p. Yhteenveto 1 p. 2001.
13 Hirvonen, Kullervo, Towards better employment using adaptive control of labour costs of an enterprise. 118 p. Yhteenveto 4 p. 2001.
14 Majava, Kirsi, Optimization-based techniques for image restoration. 27 p. (142 p.) Yhteenveto 1 p. 2001.
15 Saarinen, Kari, Near infra-red measurement based control system for thermo-mechanical refiners. 84 p. (186 p.) Yhteenveto 1 p. 2001.
16 Forsell, Marko, Improving component reuse in software development. 169 p. Yhteenveto 1 p. 2002.
17 Virtanen, Pauli, Neuro-fuzzy expert systems in financial and control engineering. 245 p. Yhteenveto 1 p. 2002.
18 Kovalainen, Mikko, Computer mediated organizational memory for process control. Moving CSCW research from an idea to a product. 57 p. (146 p.) Yhteenveto 4 p. 2002.
19 Hämäläinen, Timo, Broadband network quality of service and pricing. 140 p. Yhteenveto 1 p. 2002.
20 Martikainen, Janne, Efficient solvers for discretized elliptic vector-valued problems. 25 p. (109 p.) Yhteenveto 1 p. 2002.
21 Mursu, Anja, Information systems development in developing countries. Risk management and sustainability analysis in Nigerian software companies. 296 p. Yhteenveto 3 p. 2002.
22 Seleznyov, Alexandr, An anomaly intrusion detection system based on intelligent user recognition. 186 p. Yhteenveto 3 p. 2002.
23 Lensu, Anssi, Computationally intelligent methods for qualitative data analysis. 57 p. (180 p.) Yhteenveto 1 p. 2002.
24 Ryabov, Vladimir, Handling imperfect temporal relations. 75 p. (145 p.) Yhteenveto 2 p. 2002.
25 Tsymbal, Alexey, Dynamic integration of data mining methods in knowledge discovery systems. 69 p. (170 p.) Yhteenveto 2 p. 2002.
26 Akimov, Vladimir, Domain decomposition methods for the problems with boundary layers. 30 p. (84 p.). Yhteenveto 1 p. 2002.
27 Seyukova-Rivkind, Ludmila, Mathematical and numerical analysis of boundary value problems for fluid flow. 30 p. (126 p.) Yhteenveto 1 p. 2002.
28 Hämäläinen, Seppo, WCDMA Radio network performance. 235 p. Yhteenveto 2 p. 2003.
29 Pekkola, Samuli, Multiple media in group work. Emphasising individual users in distributed and real-time CSCW systems. 210 p. Yhteenveto 2 p. 2003.
30 Markkula, Jouni, Geographic personal data, its privacy protection and prospects in a location-based service environment. 109 p. Yhteenveto 2 p. 2003.
31 Honkaranta, Anne, From genres to content analysis. Experiences from four case organizations. 90 p. (154 p.) Yhteenveto 1 p. 2003.
32 Raitamäki, Jouni, An approach to linguistic pattern recognition using fuzzy systems. 169 p. Yhteenveto 1 p. 2003.
33 Saalasti, Sami, Neural networks for heart rate time series analysis. 192 p. Yhteenveto 5 p. 2003.
34 Niemelä, Marketta, Visual search in graphical interfaces: a user psychological approach. 61 p. (148 p.) Yhteenveto 1 p. 2003.
35 You, Yu, Situation Awareness on the world wide web. 171 p. Yhteenveto 2 p. 2004.
36 Taatila, Vesa, The concept of organizational competence – A foundational analysis. - Perusteanalyysi organisaation kompetenssin käsitteestä. 111 p. Yhteenveto 2 p. 2004.

37 LYYTIKÄINEN, VIRPI, Contextual and structural metadata in enterprise document management. - Konteksti- ja rakennemetatieto organisaation dokumenttien hallinnassa. 73 p. (143 p.) Yhteenveto 1 p. 2004.
38 KAARIO, KIMMO, Resource allocation and load balancing mechanisms for providing quality of service in the Internet. 171 p. Yhteenveto 1 p. 2004.
39 ZHANG, ZHEYING, Model component reuse. Conceptual foundations and application in the metamodeling-based systems analysis and design environment. 76 p. (214 p.) Yhteenveto 1 p. 2004.
40 HAARALA, MARJO, Large-scale nonsmooth optimization variable metric bundle method with limited memory. 107 p. Yhteenveto 1 p. 2004.
41 KALVINE, VIKTOR, Scattering and point spectra for elliptic systems in domains with cylindrical ends. 82 p. 2004.
42 DEMENTIEVA, MARIA, Regularization in multistage cooperative games. 78 p. 2004.
43 MAARANEN, HEIKKI, On heuristic hybrid methods and structured point sets in global continuous optimization. 42 p. (168 p.) Yhteenveto 1 p. 2004.
44 FROLOV, MAXIM, Reliable control over approximation errors by functional type a posteriori estimates. 39 p. (112 p.) 2004.
45 ZHANG, JIAN, Qos- and revenue-aware resource allocation mechanisms in multiclass IP networks. 85 p. (224 p.) 2004.
46 KUJALA, JANNE, On computation in statistical models with a psychophysical application. 40 p. (104 p.) 2004.,
47 SOLBAKOV, VIATCHESLAV, Application of mathematical modeling for water environment problems. 66 p. (118 p.) 2004.
48 HIRVONEN, ARI P., Enterprise architecture planning in practice. The Perspectives of information and communication technology service provider and end-user. 44 p. (135 p.) Yhteenveto 2 p. 2005.
49 VARTIAINEN, TERO, Moral conflicts in a project course in information systems education. 320 p. Yhteenveto 1p. 2005.
50 HUOTARI, JOUNI, Integrating graphical information system models with visualization techniques. - Graafisten tietojärjestelmäkuvausten integrointi visualisointitekniikoilla. 56 p. (157 p.) Yhteenveto 1p. 2005.
51 WALLENIUS, EERO R., Control and management of multi-access wireless networks. 91 p. (192 p.) Yhteenveto 3 p. 2005.
52 LEPPÄNEN, MAURI, An ontological framework and a methodical skeleton for method engineering – A contextual approach. 702 p. Yhteenveto 2 p. 2005.
53 MATYUKEVICH, SERGEY, The nonstationary Maxwell system in domains with edges and conical points. 131 p. Yhteenveto 1 p. 2005.
54 SAYENKO, ALEXANDER, Adaptive scheduling for the QoS supported networks. 120 p. (217 p.) 2005.
55 KURJENNIEMI, JANNE, A study of TD-CDMA and WCDMA radio network enhancements. 144 p. (230 p.) Yhteenveto 1 p. 2005.
56 PECHENIZKIY, MYKOLA, Feature extraction for supervised learning in knowledge discovery systems. 86 p. (174 p.) Yhteenveto 2 p. 2005.
57 IKONEN, SAMULI, Efficient numerical methods for pricing American options. 43 p. (155 p.) Yhteenveto 1 p. 2005.
58 KÄRKKÄINEN, KARI, Shape sensitivity analysis for numerical solution of free boundary problems. 83 p. (119 p.) Yhteenveto 1 p. 2005.
59 HELFENSTEIN, SACHA, Transfer. Review, reconstruction, and resolution. 114 p. (206 p.) Yhteenveto 2 p. 2005.
60 NEVALA, KALEVI, Content-based design engineering thinking. In the search for approach. 64 p. (126 p.) Yhteenveto 1 p. 2005.
61 KATASONOV, ARTEM, Dependability aspects in the development and provision of location-based services. 157 p. Yhteenveto 1 p. 2006.
62 SARKKINEN, JARMO, Design as discourse: Representation, representational practice, and social practice. 86 p. (189 p.) Yhteenveto 1 p. 2006.
63 ÄYRÄMÖ, SAMI, Knowledge mining using robust clustering. 296 p. Yhteenveto 1 p. 2006.
64 IFINEDO, PRINCELY EMILI, Enterprise resource planning systems success assessment: An integrative framework. 133 p. (366 p.) Yhteenveto 3 p. 2006.
65 VIINIKAINEN, ARI, Quality of service and pricingin future multiple service class networks. 61 p. (196 p.) Yhteenveto 1 p. 2006.
66 WU, RUI, Methods for space-time parameter estimation in DS-CDMA arrays. 73 p. (121 p.) 2006.
67 PARKKOLA, HANNA, Designing ICT for mothers. User psychological approach. – Tieto- ja viestintätekniikoiden suunnittelu äideille. Käyttäjäpsykologinen näkökulma. 77 p. (173 p.) Yhteenveto 3 p. 2006.
68 HAKANEN, JUSSI, On potential of interactive multiobjective optimization in chemical process design. 75 p. (160 p.) Yhteenveto 2 p. 2006.
69 PUTTONEN, JANI, Mobility management in wireless networks. 112 p. (215 p.) Yhteenveto 1 p. 2006.
70 LUOSTARINEN, KARI, Resource , management methods for QoS supported networks. 60 p. (131 p.) 2006.
71 TURCHYN, PAVLO, Adaptive meshes in computer graphics and model-based simulation. 27 p. (79 p.) Yhteenveto 1 p.
72 ZHOVTOBRYUKH, DMYTRO, Context-aware web service composition. 290 p. Yhteenveto 2 p. 2006.

73 KOHVAKKO, NATALIYA, Context modeling and utilization in heterogeneous networks. 154 p. Yhteenveto 1 p. 2006.
74 MAZHELIS, OLEKSIY, Masquerader detection in mobile context based on behaviour and environment monitoring. 74 p. (179 p). Yhteenveto 1 p. 2007.
75 SILTANEN, JARMO, Quality of service and dynamic scheduling for traffic engineering in next generation networks. 88 p. (155 p.) 2007.
76 KUUVA, SARI, Content-based approach to experiencing visual art. - Sisältöperustainen lähestymistapa visuaalisen taiteen kokemiseen. 203 p. Yhteenveto 3 p. 2007.
77 RUOHONEN, TONI, Improving the operation of an emergency department by using a simulation model. 164 p. 2007.
78 NAUMENKO, ANTON, Semantics-based access control in business networks. 72 p. (215 p.) Yhteenveto 1 p. 2007.
79 WAHLSTEDT, ARI, Stakeholders' conceptions of learning in learning management systems development. - Osallistujien käsitykset oppimisesta oppimisympäristöjen kehittämisessä. 83 p. (130 p.) Yhteenveto 1 p. 2007.
80 ALANEN, OLLI, Quality of service for triple play services in heterogeneous networks. 88 p. (180 p.) Yhteenveto 1 p. 2007.
81 NERI, FERRANTE, Fitness diversity adaptation in memetic algorithms. 80 p. (185 p.) Yhteenveto 1 p. 2007.
82 KURHINEN, JANI, Information delivery in mobile peer-to-peer networks. 46 p. (106 p.) Yhteenveto 1 p. 2007.
83 KILPELÄINEN, TURO, Genre and ontology based business information architecture framework (GOBIAF). 74 p. (153 p.) Yhteenveto 1 p. 2007.
84 YEVSEYEVA, IRYNA, Solving classification problems with multicriteria decision aiding approaches. 182 p. Yhteenveto 1 p. 2007.
85 KANNISTO, ISTO, Optimized pricing, QoS and segmentation of managed ICT services. 45 p. (111 p.) Yhteenveto 1 p. 2007.
86 GORSHKOVA, ELENA, A posteriori error estimates and adaptive methods for incompressible viscous flow problems. 72 p. (129 p.) Yhteenveto 1 p. 2007.
87 LEGRAND, STEVE, Use of background real-world knowledge in ontologies for word sense disambiguation in the semantic web. 73 p. (144 p.) Yhteenveto 1 p. 2008.
88 HÄMÄLÄINEN, NIINA, Evaluation and measurement in enterprise and software architecture management. - Arviointi ja mittaaminen kokonais- ja ohjelmistoarkkitehtuurien hallinnassa. 91 p. (175 p.) Yhteenveto 1 p. 2008.
89 OJALA, ARTO, Internationalization of software firms: Finnish small and medium-sized software firms in Japan. 57 p. (180 p.) Yhteenveto 2 p. 2008.
90 LAITILA, ERKKI, Symbolic Analysis and Atomistic Model as a Basis for a Program Comprehension Methodology. 321 p. Yhteenveto 3 p. 2008.
91 NIHTILÄ, TIMO, Performance of Advanced Transmission and Reception Algorithms for High Speed Downlink Packet Access. 93 p. (186 p.) Yhteenveto 1 p. 2008.
92 SETÄMAA-KÄRKKÄINEN, ANNE, Network connection selection-solving a new multiobjective optimization problem. 52 p. (111p.) Yhteenveto 1 p. 2008.
93 PULKKINEN, MIRJA, Enterprise architecture as a collaboration tool. Discursive process for enterprise architecture management, planning and development. 130 p. (215 p.) Yhteenveto 2 p. 2008.
94 PAVLOVA, YULIA, Multistage coalition formation game of a self-enforcing international environmental agreement. 127 p. Yhteenveto 1 p. 2008.
95 NOUSIAINEN, TUULA, Children's involvement in the design of game-based learning environments. 297 p. Yhteenveto 2 p. 2008.
96 KUZNETSOV, NIKOLAY V., Stability and oscillations of dynamical systems. Theory and applications. 116 p. Yhteenveto 1 p. 2008.
97 KHRIYENKO, OLEKSIY, Adaptive semantic Web based environment for web resources. 193 p. Yhteenveto 1 p. 2008.
98 TIRRONEN, VILLE, Global optimization using memetic differential evolution with applications to low level machine vision. 98 p. (248 p.) Yhteenveto 1 p. 2008.
99 VALKONEN, TUOMO, Diff-convex combinations of Euclidean distances: A search for optima. 148 p. Yhteenveto 1 p. 2008.
100 SARAFANOV, OLEG, Asymptotic theory of resonant tunneling in quantum waveguides of variable cross-section. 69 p. Yhteenveto 1 p. 2008.
101 POZHARSKIY, ALEXEY, On the electron and phonon transport in locally periodical waveguides. 81 p. Yhteenveto 1 p. 2008.
102 AITTOKOSKI, TIMO, On challenges of simulation-based globaland multiobjective optimization. 80 p. (204 p.) Yhteenveto 1 p. 2009.
103 YALAHO, ANICET, Managing offshore outsourcing of software development using the ICT-supported unified process model: A cross-case analysis. 91 p. (307 p.) Yhteenveto 4 p. 2009.
104 KOLLANUS, SAMI, Tarkastuskäytänteiden kehittäminen ohjelmistoja tuottavissa organisaatioissa. - Improvement of inspection practices in software organizations. 179 p. Summary 4 p. 2009.
105 LEIKAS, JAANA, Life-Based Design. 'Form of life' as a foundation for ICT design for older adults. - Elämälähtöinen suunnittelu. Elämänmuoto ikääntyville tarkoitettujen ICT tuotteiden ja palvelujen suunnittelun lähtökohtana. 218 p. (318 p.) Yhteenveto 4 p. 2009.

106 Vasilyeva, Ekaterina, Tailoring of feedback in web-based learning systems: Certitude-based assessment with online multiple choice questions. 124 p. (184 p.) Yhteenveto 2 p. 2009.

107 Kudryashova, ElenaV., Cycles in continuous and discrete dynamical systems. Computations, computer assisted proofs, and computer experiments. 79 p. (152 p.) Yhteenveto 1 p. 2009.

108 Blackledge, Jonathan, Electromagnetic scattering and inverse scattering solutions for the analysis and processing of digital signals and images. 297 p. Yhteenveto 1 p. 2009.

109 Ivannikov, Andriy, Extraction of event-related potentials from electroencephalography data. - Herätepotentiaalien laskennallinen eristäminen EEG-havaintoaineistosta. 108 p. (150 p.) Yhteenveto 1 p. 2009.

110 kalyakin, Igor, Extraction of mismatch negativity from electroencephalography data. - Poikkeavuusnegatiivisuuden erottaminen EEG-signaalista. 47 p. (156 p.) Yhteenveto 1 p. 2010.

111 Heikkilä, Marikka, Coordination of complex operations over organisational boundaries. 265 p. Yhteenveto 3 p. 2010.

112 Fekete, Gábor, Network interface management in mobile and multihomed nodes. 94 p. (175 p.) Yhteenveto 1 p. 2010.

113 Kujala, Tuomo, Capacity, workload and mental contents - Exploring the foundations of driver distraction. 146 p. (253 p.) Yhteenveto 2 p. 2010.

114 Lugano, Giuseppe, Digital community design - Exploring the role of mobile social software in the process of digital convergence. 253 p. (316 p.) Yhteenveto 4 p. 2010.

115 Kampylis, Panagiotis, Fostering creative thinking. The role of primary teachers. - Luovaa ajattelua kehittämässä. Alakoulun opettajien rooli. 136 p. (268 p.) Yhteenveto 2 p. 2010.

116 Toivanen, Jukka, Shape optimization utilizing consistent sensitivities. - Muodon optimointi käyttäen konsistentteja herkkyyksiä. 55 p. (130p.) Yhteenveto 1 p. 2010.

117 Mattila, Keijo, Implementation techniques for the lattice Boltzmann method. - Virtausdynamiikan tietokonesimulaatioita Hila-Boltzmann -menetelmällä: implementointi ja reunaehdot. 177 p. (233 p.) Yhteenveto 1 p. 2010.

118 Cong, Fengyu, Evaluation and extraction of mismatch negativity through exploiting temporal, spectral, time-frequency, and spatial features. - Poikkeavuusnegatiivisuuden (MMN) erottaminen aivosähkönauhoituksista käyttäen ajallisia, spektraalisia, aikataajuus - ja tilapiirteitä. 57 p. (173 p.) Yhteenveto 1 p. 2010.

119 Liu, Shenghua, Interacting with intelligent agents. Key issues in agent-based decision support system design. 90 p. (143 p.) Yhteenveto 2 p. 2010.

120 Airaksinen, Tuomas, Numerical methods for acoustics and noise control. - Laskennallisia menetelmiä akustisiin ongelmiin ja melunvaimennukseen. 58 p. (133 p.) Yhteenveto 2 p. 2010.

121 Weber, Matthieu, Parallel global optimization Structuring populations in differential evolution. - Rinnakkainen globaalioptimointi. Populaation rakenteen määrittäminen differentiaalievoluutiossa. 70 p. (185 p.) Yhteenveto 2 p. 2010.

122 Väärämäki, Tapio, Next generation networks, mobility management and appliances in intelligent transport systems. - Seuraavan sukupolven tietoverkot, liikkuvuuden hallinta ja sovellutukset älykkäässä liikenteessä. 50 p. (111 p.) Yhteenveto 1 p. 2010.

123 Viukari, Leena, Tieto- ja viestintätekniikka-välitteisen palvelun kehittämisen kolme diskurssia. - Three discourses for an ICT-service development . 304 p. Summary 5 p. 2010.

124 Puurtinen, Tuomas, Numerical simulation of low temperature thermal conductance of corrugated nanofibers. - Poimutettujen nanokuitujen lämmönjohtavuuden numeerinen simulointi matalissa lämpötiloissa . 114 p. Yhteenveto 1 p. 2010.

125 Hiltunen, Leena, Enhancing web course design using action research . - Verkko-opetuksen suunnittelun kehittäminen toimintatutkimuksen keinoin . 192 p. Yhteenveto 2 p. 2010.

126 Aho, Kari, Enhancing system level performance of third generation cellular networks through VoIP and MBMS services. 121 p. (221 p.). Yhteenveto 2 p. 2010.

127 Häkkinen, Markku, Why alarms fail. A cognitive explanatory model. 102 p. (210 p.). Yhteenveto 1 p. 2010.

128 Pennanen, Anssi, A graph-based multigrid with applications. - Graafipohjainen monihilamenetelmä sovelluksineen. 52 p. (128 p.). Yhteenveto 2 p. 2010.