UPPSALA
UNIVERSITET

Digital Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology 45

Numerical Computations with Fundamental Solutions

PER SUNDQVIST

ACTA
UNIVERSITATIS
UPSALIENSIS
UPPSALA
2005

ISSN 1651-6214
ISBN 91-554-6227-8
urn:nbn:se:uu:diva-5757

Dissertation presented at Uppsala University to be publicly examined in room 2446, MIC, Uppsala, Friday, May 13, 2005 at 13:15 for the degree of Doctor of Philosophy. The examination will be conducted in English.

Abstract

Sundqvist, P. 2005. Numerical Computations with Fundamental Solutions. (Numeriska beräkningar med fundamentallösningar). Acta Universitatis Upsaliensis. *Digital Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology* 45. viii+51 pp. Uppsala. ISBN 91-554-6227-8.

Two solution strategies for large, sparse, and structured algebraic systems of equations are considered. The first strategy is to construct efficient preconditioners for iterative solvers. The second is to reduce the sparse algebraic system to a smaller, dense system of equations, which are called the boundary summation equations.

The proposed preconditioners perform well when applied to equations that are discretizations of linear first order partial differential equations. Analysis shows that also very simple iterative methods converge in a number of iterations that is independent of the number of unknowns, if our preconditioners are applied to certain scalar model problems. Numerical experiments indicate that this property holds also for more complicated cases, and a flow problem modeled by the nonlinear Euler equations is treated successfully.

The reduction process is applicable to a large class of difference equations. There is no approximation involved in the reduction, so the solution of the original algebraic equations is determined exactly if the reduced system is solved exactly. The reduced system is well suited for iterative solution, especially if the original system of equations is a discretization of a first order differential equation. The technique is used for several problems, ranging from scalar model problems to a semi-implicit discretization of the compressible Navier-Stokes equations.

Both strategies use the concept of fundamental solutions, either of differential or difference operators. An algorithm for computing fundamental solutions of difference operators is also presented.

Keywords: fundamental solution, partial differential equation, partial difference equation, iterative method, preconditioner, boundary method

Per Sundqvist, Department of Information Technology, Box 337, Uppsala University, SE-75105 Uppsala, Sweden

ISSN 1651-6214
ISBN 91-554-6227-8
urn:nbn:se:uu:diva-5757 (http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-5757)

To Lina and our child

List of Papers

This thesis is based on the following papers, which are referred to in the text by their Roman numerals.

I Brandén, H., Sundqvist, P. (2005) Preconditioners Based on Fundamental Solutions, *Submitted to BIT*.

II Brandén, H., Sundqvist, P. (2004) An Algorithm for Computing Fundamental Solutions of Difference Operators, *Numer. Algorithms 36 (4): 331-343*.

III Brandén, H., Holmgren, S., Sundqvist, P. (2005) Discrete Fundamental Solution Preconditioning for Hyperbolic Systems of PDE, *Submitted to J. Sci. Comput.*

IV Sundqvist, P. (2004) Boundary Summation Equations, *Submitted to SIAM J. Sci. Comput.*

V Sundqvist, P., Holmgren, S. (2005) Navier–Stokes Equations for Low Mach Number Flows Solved by Boundary Summation, *Submitted to Num. Meth. Partial Differential Equations*

Reprints were made with permission from the publishers.

FMB *This work was funded by*
the Graduate School in Mathematics and Computing.

Contents

1 Introduction

Partial differential equations, PDEs, are important, since they are used to mathematically model many problems in science and industry. For some problems, it is either impossible or very expensive to make experiments, and then mathematical models provide a necessary or economical alternative. Colliding galaxies is an example of a problem for which experiments are impossible, and wind tunnel tests for optimizing aircraft design is an example of an expensive set of experiments. The PDEs considered in this thesis are mostly of a kind that describes transport phenomena, such as flow or wave propagation.

Discretizations are important, since most interesting PDEs cannot be solved analytically. In order to obtain an approximate solution, the equations must be discretized, e.g. by finite difference or finite element methods. The choice of discretization scheme depends on e.g. the type of the PDE, the geometry of the problem, and personal taste. In this thesis, finite difference discretizations are considered, but the algorithms we present are applicable also to finite element and finite volume discretizations, as long as the grid is structured.

Efficient solution methods for discretized problems are also important, since the computational time and memory requirements of the solver decides how accurately a given PDE can be solved. Both the development of the computers and of the algorithms continue to push the limit on how complex PDEs that can be solved at an acceptable accuracy. It is to this field, devising efficient solution methods for discretized problems, that the present thesis contributes.

Both time-dependent and time-independent PDEs are often discretized in such a way that the main computational task is the solution of a large sparse linear system of algebraic equations. These systems are often so large that direct solvers are not feasible, due to their memory requirements. Instead, iterative solvers are used, with the effect that the algebraic system is solved approximately.

Usually, the standard iterative solvers do not converge rapidly enough, or even at all, due to the nature of discretized PDEs. If they converge, the convergence depends on the number of unknowns in the discretization in an unfavorable way. Iterative methods that do guarantee convergence, such as GMRES, often require a significant amount of memory and work.

By the use of preconditioners, both time and memory requirements of iterative solvers can be reduced. In Papers I and III, preconditioners are constructed using fundamental solutions, either of differential or difference operators. These papers are presented in sections 2.1 and 2.3. Using GMRES, increased convergence speed also implies a reduction of memory requirement, at least if the convergence criterion is fulfilled before the method is restarted.

Boundary methods provide another tool used to circumvent the difficulties with solving PDEs numerically. The term ”boundary method” refers to methods that reduce a problem posed on some domain, to an equivalent problem where the unknowns are related only to the boundary of the same domain. A common property of boundary methods is that they reduce the number of unknowns that need to be determined by the solver. A boundary method defined for difference equations is introduced in Paper IV and applied to flow problems in Paper V. These papers are presented in sections 2.4 and 2.5. *Boundary summation equations* is the term used for the reduced system.

Even if the number of unknowns is reduced by a boundary method, it might still be large enough to make iterative solvers worthwhile, especially if the dimensionality of the original domain is high, or if there is a good initial guess available. One drawback of boundary methods is that they all seem to generate dense matrices. The residual evaluation becomes a nontrivial task, compared to the finite difference case. In Paper IV, an efficient residual evaluation algorithm for the boundary summation equations is proposed. If GMRES is used, any reduction in the number of unknowns is itself valuable, since a larger Krylov subspace can be stored at the same memory cost. The restarting length can hence be increased.

Low time and memory costs are not the only valuable properties of algorithms. Implementational aspects are also important, first of all from the programmer’s point of view. If the programmer does not understand the algorithm well, it is unlikely that the program will be efficiently implemented. As an example, we can take the residual evaluation when using the boundary summation equations. The main part of computational work consists of discrete Fourier transforms, for which there exist highly optimized codes. The rest of the work is an application of a sparse matrix, and the composition of the two tasks should not be a serious challenge to any programmer.

1.1 Computations with Fundamental Solutions

The main topic of this thesis is the solution of large, sparse systems of equations that represent discretizations of partial differential equations. Two strategies are used. The first strategy is to construct preconditioners with the purpose to speed up iterative solvers. The second strategy is to reduce a constant coefficient difference equation to the boundary of the original domain. Common to both strategies is that they use fundamental solutions.

Usually, fundamental solutions are considered as a tool used in theory for PDEs, e.g. to prove properties of solutions such as existence and differentiability [33]. Then, a fundamental solution E of a differential operator P is defined to be a distribution (acting on infinitely differentiable test functions with compact support) that satisfies

$$PE = \delta, \tag{1.1}$$

where δ is the Dirac distribution. The idea in Paper I is to use fundamental solutions of differential operators with the purpose to construct preconditioners for discretized PDEs. The numerically troublesome fact that fundamental solutions are distributions is dealt with by using E only outside its singular support. The existence of fundamental solutions of differential operators relies on that the operator is linear with constant coefficients, but the resulting preconditioner can be applied also to variable coefficient problems.

Fundamental solutions of differential operators are used also in other methods for PDEs, e.g. to reformulate them into boundary integral equations. This topic is reviewed in chapter 4, which gives a survey on boundary methods. The survey is intended to put the boundary summation equations in perspective. Also the fundamental solution preconditioners are put in perspective, by the review of related preconditioners in chapter 3.

Fundamental solutions are known for many differential operators, but for difference operators very few can be found in the literature. The definition used in this thesis can be written exactly as the definition of fundamental solutions of differential operators in (1.1), by letting P be a constant coefficient difference operator and letting δ be a grid function that is zero everywhere, except for at the origin, where it equals one. An existence proof is given in [9], but the proof is not constructive. An algorithm for computing fundamental solutions of difference operators is introduced in Paper II, which is presented in section 2.2. This algorithm is first used in Paper III, where the approach is conceptually similar to the one in Paper I. A preconditioner is constructed, but this time it uses a fundamental solution of a difference operator instead of a differential operator.

The boundary summation equations are presented in paper IV. There, fundamental solutions computed by the algorithm in Paper II are used to reduce difference equations, posed on some domain, to equations on its boundary, by eliminating all unknowns in the interior of the domain. The method is easy to implement in a general fashion, so that the application to new problems is straight forward.

Even if the preconditioner or the reduction to the boundary are easily implemented, there is a considerable amount of work related to testing them on advanced problems. Therefore, simplified model problems are used extensively in the development of new numerical algorithms. More advanced equations can be attacked with confidence only after that the algorithm is working as expected for simpler ones. Also, the analysis of a preconditioned iterative method is often hard also for simplified problems, and real application problems are almost never within reach.

One more difficulty with application problems is that there can be many parameters that must be properly tuned. They can be physical parameters that need to lie in certain intervals to assure the existence of a solution, or parameters of the discretization, such as a time step that can cause instability if chosen too large. This difficulty is present independently of what kind of preconditioner one would like to use.

Regardless of how complicated a problem is, there is a need for efficient solvers of linear systems. Many algorithms for solving nonlinear problems are constructed such that most of the computations consist of solving a sequence of linear problems. If the linear solver can be made faster, the over all performance of the nonlinear algorithm can be substantially improved. The preconditioner based on fundamental solutions of difference operators is used in a nonlinear solver in Paper III, with the purpose to find steady state solutions of a flow problem. There, the linear problems in the sequence are not actually solved, since it shows that applying the preconditioner once for each linear problem is sufficient. The Boundary Summation Equations are used in a semi-implicit algorithm [22] for solving a time-dependent flow problem in Paper V. For that particular semi-implicit solver, a constant coefficient, linear difference equation needs to be solved in each time step, which is exactly what the Boundary Summation Equations are designed to handle.

Image deblurring is an example of a completely different application where the algorithms of the thesis might be useful. The reason is that the process of deblurring an image involves the solution of a large, sparse system of equations. These equations can often be identified with a constant coefficient difference equation. In that case, both the fundamental solution preconditioners and the boundary summation equations are applicable.

2 Our contributions

This chapter contains presentations of the five papers I–V. The order of the presentation, and of the papers, is the same as the order in which the research was performed. The work was done between late 1999 and early 2005. Several slightly different notations are used, and each section uses the same notation as the paper to which it refers.

2.1 Paper I

Paper I describes a way to use fundamental solutions of differential operators to construct preconditioners for difference equations that are discretizations of PDEs. The differential operator P is scalar, linear, and has constant coefficients. The differential equation is

$$\begin{cases} Pu(x) = f(x), & x \in \Omega, \\ Bu(x) = g(x), & x \in \Gamma \subset \partial\Omega, \end{cases} \tag{2.1}$$

where Ω is a convex and bounded subset of $\mathbb{R}^2$. Here, the preconditioner is a discretization of an integral operator K, given by

$$(Ku)(x) = \int_\Omega E(x-y)u(y)dy, \quad x \in \Omega,$$

where E is a fundamental solution of P.

The discretization of (2.1) is represented by a finite difference approximation denoted by

$$\begin{cases} P_h u_i = f_i, & i \in \Omega_h, \\ B_h u_i = g_i, & i \in \Gamma_h, \end{cases}$$

where $\Omega_h = \{(i_1,i_2)|(i_1h_1,i_2h_2) \in \Omega\}$, $\Gamma_h = \{(i_1,i_2)|(i_1h_1,i_2h_2) \in \Gamma\}$, and h_1 and h_2 are the discretization steps in the two space dimensions. These definitions suppose that the grid is uniform and that the boundary Γ is aligned with the grid.

The integral operator is discretized by

$$K_h u_i = \sum_{i\in\Omega_h} \widetilde{E}_{i-j} u_j h_1 h_2, \quad i \in \Omega_h,$$

where $\widetilde{E}$ equals E on the grid, except for in a neighborhood ω of the set where E is singular. Inside ω, we use $\widetilde{E}_i = E_i$, where E_i is the solution to

$$P_h E_i = \begin{cases} 1, & i = 0, \\ 0, & i \in \omega \setminus 0, \end{cases} \tag{2.2}$$

with boundary conditions given by E on $\partial\omega$.

The analysis in Paper I suggests that grid independent convergence could be obtained for first order problems. Numerical experiments also verifies that it is possible. However, by analyzing the differential and integral operators, we also find that it is impossible to achieve grid independent convergence if the order of the differential operator is higher than one, at least when using the fixed point method.

The main contribution of Paper I is to introduce the idea of approximating the inverse of a difference operator by discretizing a truncated convolution operator. The most important experimental result is that grid independent convergence is achieved for a first order PDE on the unit square, with

$$P = \frac{\partial}{\partial x_1} + \frac{\partial}{\partial x_2}. \tag{2.3}$$

It is discretized using second order centered finite differences and second order artificial viscosity,

$$P_h = -\varepsilon_h \left(D_+^{x_1} D_-^{x_1} + D_+^{x_2} D_-^{x_2} \right) + D_0^{x_1} + D_0^{x_2}.$$

The parameter ε_h is chosen proportional to $h = h_1 = h_2$, and the resulting difference operator is first order accurate. The solution is prescribed at the left and bottom boundaries, and numerical boundary conditions that approximate the differential operator are used at the right and upper boundaries.

It is required that the artificial viscosity in the discretization is taken into account when choosing the formula for the fundamental solution. Although P_h is a consistent approximation of P in (2.3), it also approximates

$$P' = \frac{\partial}{\partial x_1} + \frac{\partial}{\partial x_2} - \varepsilon_h \left(\frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2} \right),$$

for every finite ε_h. A fundamental solution of P' is

$$E(x) = \frac{1}{2\pi\varepsilon_h} e^{(x_1+x_2)/2\varepsilon_h} K_0 \left(\frac{|x|}{\sqrt{2}\varepsilon_h} \right),$$

where K_0 is the modified Bessel function of the second kind. This function is singular at the origin. Therefore, $\widetilde{E}_{i_1,i_2} = E(i_1 h, i_2 h)$ is employed at all grid points, except for in a small square neighborhood of the origin, where the linear system (2.2) is solved. Table 2.1 shows average numbers of preconditioned fixed point iterations required to reduce the spectral norm of the initial random residual by a factor 10^7.

Table 2.1: *Average numbers of iterations for the scalar convection problem. The preconditioner is based on a fundamental solution of the convection-diffusion operator.*

h^{-1}	$\varepsilon_h = h$	$\varepsilon_h = h/2$	$\varepsilon_h = h/3$	$\varepsilon_h = h/4$
16	17	5	13	27
32	17	5	13	25
64	19	5	12	24
128	20	5	11	23
256	19	5	11	22
512	19	5	10	20

2.2 Paper II

Paper II differs from the other papers in that it does not explicitly deal with the solution of algebraic systems of equations. Instead, it provides an algorithm for computing the fundamental solutions that are needed in Papers III, IV, and V.

Here, P denotes a d-dimensional partial difference operator acting on vector valued grid functions with n_c components in each grid point,

$$(Pu)_j = \sum_{k\in\mathbb{Z}^d} B_k u_{j-k}, \quad j \in \mathbb{Z}^d,$$

where B_k are $n_c \times n_c$ matrices that are zero for all k outside a neighborhood of the origin. We are interested in computing a fundamental solution of P on $\Omega_m = (-m_1, \dots, m_1 - 1) \times \cdots \times (-m_d, \dots, m_d - 1)$, i.e. a grid function E that satisfies

$$(PE)_j = \delta_j I \equiv \begin{cases} I, & j = 0, \\ 0, & j \in \Omega_m \setminus 0, \end{cases} \tag{2.4}$$

where I is the identity matrix of size n_c.

The symbol of P is the discrete Fourier transform of B_k and is denoted $\widehat{B}$. It is defined on the domain Ω_m,

$$\widehat{B}_k = \sum_{j\in\Omega_m} B_j e^{-\pi i\left(\frac{j_1 k_1}{m_1} + \cdots + \frac{j_d k_d}{m_d}\right)}, \quad k \in \Omega_m. \tag{2.5}$$

If $\widehat{B}_k$ is nonsingular for all $k \in \Omega_m$, it is easy to determine a fundamental solution by solving

$$\widehat{B}_k\widehat{E}_k = \hat{\delta}_k I, \quad k \in \Omega_m,$$

and computing the inverse discrete Fourier transform

$$E_j = \frac{1}{2^d m_1 \cdots m_d} \sum_{k\in\Omega_m} \widehat{E}_k e^{\pi i\left(\frac{j_1k_1}{m_1}+\cdots+\frac{j_dk_d}{m_d}\right)}, \quad j \in \Omega_m.$$

Both the transform and its inverse can be computed by the fast Fourier transform algorithm in $\mathscr{O}(m_1\cdots m_d \log(m_1\cdots m_d))$ arithmetic operations.

Paper II deals with the problem that occurs when $\widehat{B}_k$ is singular for some k. We formulate an algorithm that makes it possible to compute a fundamental solution using the fast Fourier transform also when $\widehat{B}$ has singularities.

We show that existence of a fundamental solution that is periodic in all dimensions is equivalent to that the symbol is nonsingular everywhere. However, de Boor, Höllig, and Riemenschneider have shown that a well behaved fundamental solution exists for all difference operators with constant coefficients [9], also for operators whose symbols have zeros.

Our idea is to compute a fundamental solution by imposing periodic boundary conditions in all dimensions except one. This leads to that we can prove the existence of a fundamental solution in terms of well established one dimensional difference equation theory. Also, a fast solution algorithm is still available.

In the algorithm for computing E when the symbol has singularities, the discrete Fourier transform is first applied to the periodic dimensions of the difference equation. This is easily performed analytically. The partly transformed problem consists of a set of ordinary difference equations, each with $2m_p$ unknowns, where p is the dimension that was not transformed. Different fundamental solutions are obtained by imposing different conditions instead of periodicity in dimension p. Either some standard boundary conditions can be used, in which case the resulting linear systems are solved using a direct method. Or the equations can be solved as underdetermined systems by the use of QR-factorizations. In either case, the fundamental solution is obtained by applying a $(d-1)$-dimensional inverse Fourier transform to the solution of the partly transformed problem.

2.3 Paper III

A modification of the preconditioning method in Paper I is to use a fundamental solution of the difference operator instead of one to the differential operator. This approach is used in Paper III. The algorithm developed in Paper II is used to compute the fundamental solutions.

One advantage of the approach in Paper III, compared to that in Paper I, is that no explicit a-priori knowledge about the fundamental solution is required. In Paper I we use both an explicit formula for the fundamental solution and information on its singular support. In Paper III, the initialization of the preconditioner involves the computation of a fundamental solution of a difference operator. The new approach is therefore easier to apply to more advanced problems.

In Paper III, the steady state two dimensional Euler equations are solved in a narrowing channel. These PDEs have variable coefficients, nonlinearities and are solved on a nonuniform grid. The differential operator is given by

$$A_1(\mathbf{v})\frac{\partial}{\partial x_1}+A_2(\mathbf{v})\frac{\partial}{\partial x_2}, \tag{2.6}$$

where $\mathbf{v}=(\rho,v_1,v_2)^T$ and

$$A_1(\mathbf{v})=\begin{pmatrix} v_1 & \rho & 0 \\ c^2/\rho & v_1 & 0 \\ 0 & 0 & v_1 \end{pmatrix}, \qquad A_2(\mathbf{v})=\begin{pmatrix} v_2 & 0 & \rho \\ 0 & v_2 & 0 \\ c^2/\rho & 0 & v_2 \end{pmatrix}.$$

Here, v_1 and v_2 are velocity components and ρ is the density. The local speed of sound is denoted by c. The domain on which the PDE is posed is shown in Figure 2.1. More information on the problem setting, such as boundary conditions and discretization schemes is found in Paper III.

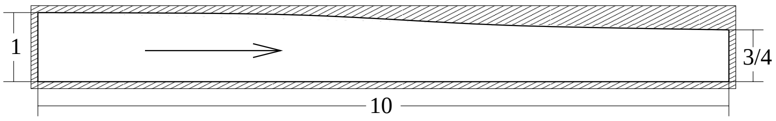


Figure 2.1: The narrowing channel. The length is ten times the width of the inflow at the left, and the width of the outflow is 3/4 of the inflow.

The discretized equations can be written

$$B(v)v=g, \tag{2.7}$$

which indicates that the coefficient matrix depends on the solution. To obtain the steady state solution, a time stepping procedure is applied to a preconditioned version of (2.7),

$$v^{i+1}=(I-K(v^i)B(v^i))v^i+K(v^i)g, \tag{2.8}$$

where $K(v^i)$ is the preconditioner that is based on a fundamental solution of a constant coefficient difference operator

$$\mathscr{B}v_{j_1,j_2} = \sum_{(l_1,l_2)\in\ell} B^{l_1,l_2} v_{j_1-l_1,j_2-l_2},$$

which in each iteration is constructed as an average of the variable coefficient difference operator that defines $B(v^i)$ in (2.7) on interior grid points. Then, the fundamental solution of $\mathscr{B}$ is computed using the algorithm in Paper II, and used for preconditioning.

Grid refinement studies are performed for two discretizations, one first order accurate approximation of upwind type and one second order accurate centered approximation with artificial viscosity. A semicirculant preconditioner [28, 29] was used for comparison.

Table 2.2 shows iteration counts for the upwind discretization. The F-column contains results for the fundamental solution preconditioner and the SC-column for the semicirculant and m_1 is the number of grid points in the dimension along the channel.

Table 2.2: *Numbers of iterations for two discretizations of the nonlinear Euler equations.*

	upwind		centered	
m_1	F	SC	F	SC
80	107	27	222	56
160	105	27	221	58
320	106	33	222	62
640	108	38	227	65

The numbers of iterations does not depend significantly on the grid size. However, we also see that the semicirculant preconditioner performs better than the fundamental solution preconditioner. One conclusion is that the fundamental solution preconditioner needs more research to become competitive for this kind of problems.

A possible interpretation on another level is that the semicirculant preconditioner is a better approximation of the inverse of $B(v^i)$, but that would not be a completely satisfactory explanation. The reason is that the problem is nonlinear, and the coefficient matrix corresponding to the steady state solution is unknown. If the inverse of $B(v^i)$ is used as a preconditioner, it still takes several iterations to reach convergence for the upwind discretization, sometimes actually more than with the semicirculant preconditioner.

2.4 Paper IV

The concept of boundary summation equations is introduced in Paper IV. The idea is to reduce a large sparse system, representing a partial difference equation with boundary conditions, to a much smaller, dense system. The similarities to boundary integral equations and some other related methods are described in chapter 4.

Boundary summation equations are defined for difference equations, and one important application of such equations is PDEs. Here, a constant coefficient PDE in d dimensions is denoted

$$\begin{cases} \boldsymbol{P_\Omega u}(x) = \boldsymbol{f_\Omega}(x), & x \in \boldsymbol{\Omega} \subset \mathbb{R}^d, \\ \boldsymbol{P_\Gamma}(x)\boldsymbol{u}(x) = \boldsymbol{f_\Gamma}(x), & x \in \boldsymbol{\Gamma} \subset \partial\boldsymbol{\Omega}, \end{cases} \tag{2.9}$$

where $\boldsymbol{P_\Omega}$ and $\boldsymbol{P_\Gamma}$ are differential operators, $\boldsymbol{u}$ is an unknown, vector valued function with n_c components, $\boldsymbol{f_\Omega}$ is a forcing function, $\boldsymbol{f_\Gamma}$ specifies boundary data, and $\boldsymbol{\Gamma}$ is the subset of $\partial\boldsymbol{\Omega}$ where boundary conditions are imposed. Note that the operator $\boldsymbol{P_\Gamma}$ is allowed to have varying coefficients.

Using a finite difference approximation on a uniform grid and a suitable ordering of the unknowns, the discretization of (2.9) is represented by the linear system

$$\begin{pmatrix} P_\Gamma & P_{\Gamma\Omega} \\ P_{\Omega\Gamma} & P_\Omega \end{pmatrix} \begin{pmatrix} u_\Gamma \\ u_\Omega \end{pmatrix} = \begin{pmatrix} f_\Gamma \\ f_\Omega \end{pmatrix}, \tag{2.10}$$

where the block row $\begin{pmatrix} P_\Gamma & P_{\Gamma\Omega} \end{pmatrix}$ represents the equations that can be referred to as boundary conditions. The block row $\begin{pmatrix} P_{\Omega\Gamma} & P_\Omega \end{pmatrix}$ represents a difference operator with constant coefficients, denoted by $\hat{P}$. In (2.10), u_Γ contains the unknowns u_i for $i \in \Gamma$, and u_Ω the unknowns for $i \in \Omega$, where Γ contains those grid points where boundary conditions are imposed and Ω is the set of grid points where $\hat{P}$ is applied without modification. Any equation used to close the system defined by $\hat{P}$ is referred to as a boundary condition, even if it is a lower order discretization of $\boldsymbol{P_\Omega}$.

The difference operator is defined by

$$\hat{P}v_i = \sum_{j \in S} B_j v_{i-j}, \quad i \in \Omega, \tag{2.11}$$

where B_j, $j \in S$ are n_c-matrices, so $\hat{P}$ is a mapping from grid functions defined on $\overline{\Omega} = \Omega \cup \Gamma$ to grid functions defined on Ω.

An auxiliary vector of unknowns v is introduced by

$$\begin{pmatrix} K_\Gamma & K_{\Gamma\Omega} \\ K_{\Omega\Gamma} & K_\Omega \end{pmatrix} \begin{pmatrix} v_\Gamma \\ v_\Omega \end{pmatrix} = \begin{pmatrix} u_\Gamma \\ u_\Omega \end{pmatrix}, \tag{2.12}$$

where

$$\begin{pmatrix} K_\Gamma & K_{\Gamma\Omega} \\ K_{\Omega\Gamma} & K_\Omega \end{pmatrix} \begin{pmatrix} v_\Gamma \\ v_\Omega \end{pmatrix}_i = \sum_{j\in\bar{\Omega}} E_{i-j} v_j, \quad i \in \bar{\Omega}. \tag{2.13}$$

Here E is a fundamental solution of $\hat{P}$, i.e. a grid function that satisfies

$$\hat{P} E_j = \delta_j I.$$

The indices i and j in (2.13) are the same as in (2.11) and refers to grid points, not rows and columns of the K-matrix. Using this convention, we can write

$$\begin{pmatrix} P_{\Omega\Gamma} & P_\Omega \end{pmatrix} u_i = \sum_{j\in S} B_j u_{i-j} = f_i, \quad i \in \Omega.$$

Inserting (2.12) in (2.10) yields

$$\begin{pmatrix} P_\Gamma & P_{\Gamma\Omega} \\ P_{\Omega\Gamma} & P_\Omega \end{pmatrix} \begin{pmatrix} K_\Gamma & K_{\Gamma\Omega} \\ K_{\Omega\Gamma} & K_\Omega \end{pmatrix} \begin{pmatrix} v_\Gamma \\ v_\Omega \end{pmatrix} = \begin{pmatrix} A & C \\ 0 & I \end{pmatrix} \begin{pmatrix} v_\Gamma \\ v_\Omega \end{pmatrix} = \begin{pmatrix} f_\Gamma \\ f_\Omega \end{pmatrix},$$

since

$$\begin{aligned} \begin{pmatrix} P_{\Omega\Gamma} & P_\Omega \end{pmatrix} \begin{pmatrix} K_\Gamma & K_{\Gamma\Omega} \\ K_{\Omega\Gamma} & K_\Omega \end{pmatrix} v_i &= \sum_{j\in S} B_j v_{i-j} \sum_{k\in\bar{\Omega}} E_{i-j-k} v_{i-j} \\ = \sum_{k\in\bar{\Omega}} \sum_{j\in S} B_j E_{i-j-k} v_{i-j} &= \sum_{k\in\bar{\Omega}} \delta_{i-k} I v_k, \quad i \in \Omega. \end{aligned} \tag{2.14}$$

The last block row is eliminated, and the resulting reduced equations are

$$A v_\Gamma = f_\Gamma - C f_\Omega, \tag{2.15}$$

where

$$A = P_\Gamma K_\Gamma + P_{\Gamma\Omega} K_{\Omega\Gamma},$$

and

$$C = P_\Gamma K_{\Gamma\Omega} + P_{\Gamma\Omega} K_\Omega.$$

The system (2.15) is square and has a number of unknowns that is equal to the number of grid points in Γ. It is this system that we call *boundary summation equations*. Once it is solved, the solution to (2.10) is computed by an application of (2.13),

$$\begin{pmatrix} u_\Gamma \\ u_\Omega \end{pmatrix} = \begin{pmatrix} K_\Gamma & K_{\Gamma\Omega} \\ K_{\Omega\Gamma} & K_\Omega \end{pmatrix} \begin{pmatrix} v_\Gamma \\ f_\Omega \end{pmatrix}. \tag{2.16}$$

The K-matrix is by construction a reordered Toeplitz matrix, and it is hence possible to apply it, and its submatrices, using a standard technique where the main part of the computations is performed by FFT. All blocks in the P-matrix are supposed to be sparse, implying that all matrices needed in an iterative method for (2.15) can be applied in $\mathscr{O}(n \log n)$ arithmetic operations, where n is the number of grid points in $\bar{\Omega}$.

In Paper IV, the algorithm is first applied to discretizations of a number of scalar PDEs on the unit square. The boundary summation equations are solved by GMRES. The results indicate that grid independent convergence occurs for first order scalar and hyperbolic systems of PDE. There is also a proof of this for a specific discretization of the scalar convection equation, stating the strong result that the norm of A is strictly less than one. The convergence is not grid independent for discretizations of second order equations, but the numbers of iterations are moderate. Also, quite many GMRES iterations can be performed without restarting, since the number of unknowns is reduced.

The definition of the boundary summation equations does not demand that the domain is a square. Experiments are performed successfully also for PDEs posed on two dimensional domains of L- and circular shape. These are embedded in rectangles, so for domains of very complicated shape, there is a need for an alternative treatment.

Other directions of generalization is to systems of difference equations and to higher dimensions. In Paper IV, experiments are performed for first order scalar equations posed on hypercubes of dimension up to six, and for a discretization of a linearized version of the Euler equations with three components in two dimensions. Both kinds of experiments show grid independent convergence.

2.5 Paper V

In paper V, the boundary summation equations are used to solve the isentropic Navier–Stokes equations as posed in [22], modeling time-dependent fluid flow,

$$V_t + P_0 V + P_1(V)V = P_2(V)V, \tag{2.17}$$

where $V = (\phi, u, v)^T$ is the unknown function with u and v as velocity components and ϕ derived from the speed of sound. The differential operators are

$$
\begin{aligned}
P_0 &= \frac{1}{\varepsilon}\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}\frac{\partial}{\partial x_1} + \frac{1}{\varepsilon}\begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}\frac{\partial}{\partial x_2}, \\
P_1(V) &= \begin{pmatrix} u & \gamma_1\phi & 0 \\ \gamma_1\phi & u & 0 \\ 0 & 0 & u \end{pmatrix}\frac{\partial}{\partial x_1} + \begin{pmatrix} v & 0 & \gamma_1\phi \\ 0 & v & 0 \\ \gamma_1\phi & 0 & v \end{pmatrix}\frac{\partial}{\partial x_2}, \text{ and} \\
\rho Re P_2(V) &= \begin{pmatrix} 0 & 0 & 0 \\ 0 & \lambda+2\mu & 0 \\ 0 & 0 & \mu \end{pmatrix}\frac{\partial^2}{\partial x_1^2} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & \mu & 0 \\ 0 & 0 & \lambda+2\mu \end{pmatrix}\frac{\partial^2}{\partial x_2^2} \\
&\quad + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & \lambda+\mu \\ 0 & \lambda+\mu & 0 \end{pmatrix}\frac{\partial^2}{\partial x_1 \partial x_2}.
\end{aligned} \tag{2.18}
$$

Here, ε is the Mach number, Re is the Reynolds number, ρ is the density and the Lamé coefficients are given by $\lambda = -2/3$ and $\mu = 1$.

The problem is difficult, especially for small ε, since then different time scales are involved. Even if we are interested only in the slower scale, the faster will impose a strict stability condition if an explicit time stepping is used. This criterion can be relaxed, e.g. by using the semi-implicit scheme as in [22],

$$
\frac{U^{n+1} - U^{n-1}}{2k} + Q_0 U^{n+1} + Q_1(U^n)U^n = Q_2(U^{n-1})U^{n-1}. \tag{2.19}
$$

Here, we use standard, centered, second order accurate finite difference operators Q_0, Q_1, and Q_2 for P_0, P_1, and P_2 and a uniform grid. Hence, to take one time step of size k, we solve

$$
(I + 2kQ_0)U^{n+1} = (I + 2kQ_2(U^{n-1}))U^{n-1} - 2kQ_1(U^n)U^n, \tag{2.20}
$$

and to apply the boundary summation equations to the left hand side, we need a fundamental solution of the difference operator

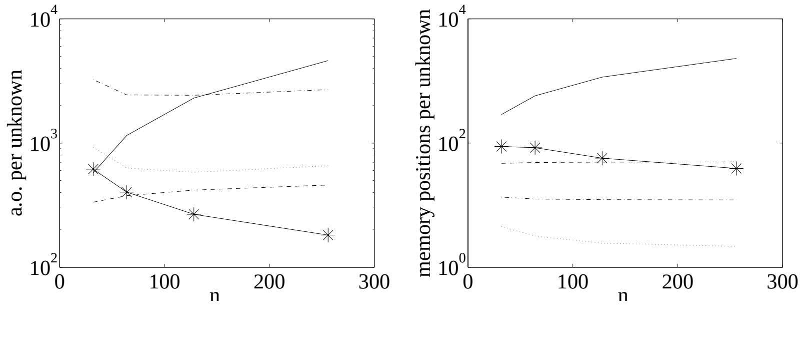
$$
I - \frac{2k}{\varepsilon}(A_1 D_0^{x_1} + A_2 D_0^{x_2}), \tag{2.21}
$$

where A_1 and A_2 are the first two matrices in (2.18). The symbol of the operator (2.21) turns out to be invertible everywhere, or in other words: for this specific case, there exists a fundamental solution that is periodic in both dimensions.

In Paper V, the boundary summation equations are applied to two problems;

the driven cavity and the backward facing step. The driven cavity is a unit square where the top boundary moves to the right at unit speed. The other three boundaries are fixed. Due to the no-slip conditions, the flow will rotate clockwise, and in the bottom corners there will be recirculation in the counter-clockwise direction. The backward facing step is a section of a channel whose width increases discontinuously. Recirculation will occur behind the step.

First, we compare various solution methods for the driven cavity, with respect to arithmetic complexity and memory requirements. The coefficient matrix for the boundary summation equations is called A, and the large, sparse matrix representing the original equations is called P. The methods we compare are a direct band solver applied to P, GMRES applied to P, a direct solver applied to A, and GMRES applied to A, both for the periodic fundamental solution, denoted by E^p, and for the more general fundamental solution, denoted by E^g, computed by the algorithm in Paper II. Figure 2.2 shows the results, where the domain is discretized with n grid points in each dimension.



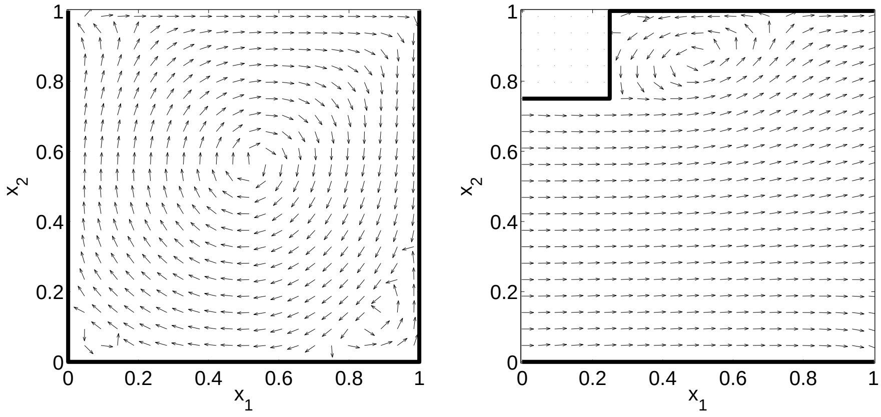
(a) Arithmetic cost per time step, excluding the evaluation of the explicit part.

(b) Memory requirement.

Figure 2.2: Arithmetic complexities and memory requirements for band Gaussian elimination on P (solid), Gaussian elimination on A with E^p (dashed), GMRES on A with E^g (dash-dot), GMRES on A with the E^p (dotted), and GMRES on P (solid with star).

It shows that the boundary summation equations solved by GMRES are competitive, especially if the periodic fundamental solution is used. Compared to direct methods, the gain is that in the time stepping, we have access to a relatively good initial guess for the iterative method. The comparison to GMRES on P shows that each time step is more expensive for the boundary summation equations, but consumes considerably less memory. When using the boundary summation equations with the periodic fundamental solution, the number of arithmetic operations required to solve the implicit part each timestep is only about three times as large as the requirement for evaluating the explicit part.

Figures 2.3(a) and 2.3(b) shows solutions to the driven cavity and the backward facing step problems.



(a) The driven cavity, $T = 35$.

(b) The backward facing step, $T = 2$.

Figure 2.3: Normalized flow-field for $Re = 1000$, $\varepsilon = 0.01$.

3 Review of Related Preconditioners

This chapter is a review of some classes of preconditioners that are related to the ones based on fundamental solutions, and is meant to put papers I and III in perspective. The first three classes are preconditioners for Toeplitz matrices. A one-level Toeplitz matrix T is a matrix that is constant along its diagonals, i.e.

$$T = \begin{bmatrix} B_0 & B_{-1} & \cdots & B_{-n+2} & B_{-n+1} \\ B_1 & B_0 & B_{-1} & & B_{-n+2} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ B_{n-2} & & B_1 & B_0 & B_{-1} \\ B_{n-1} & B_{n-2} & \cdots & B_1 & B_0 \end{bmatrix}. \tag{3.1}$$

It is called a block Toeplitz matrix if the B_k are matrices. In the generalization to the multilevel case, each B_k is itself a Toeplitz matrix, with elements that are Toeplitz matrices, and so on.

A finite difference discretization of a differential operator with constant coefficients on a uniform grid can be represented by a difference operator with constant coefficients. The corresponding coefficient matrix would have (multilevel) Toeplitz structure, if it were not for the boundary conditions. Even though they impose only a low rank disturbance, they can have large impact on the behavior of the iterative method. Also, for matrices arising from discretizations of differential equations, the entries will depend on the grid size. If there are derivatives of different order involved, this dependence can not be removed by a scaling. Other important properties are that if there are derivatives of odd order in the PDE, the coefficient matrix will be non-Hermitian, and if the discretization is not centered, the matrix will be nonnormal. In spite of some differences, our method relates to preconditioning strategies for Toeplitz matrices and block Toeplitz matrices with Toeplitz blocks.

There are many papers treating the iterative solution of Toeplitz systems. Many follow the idea by Strang [59] and use the preconditioned conjugate gradient method, PCG. For a large number of references, see the book [43] by Ng. There, several preconditioners are considered and a number of applications are mentioned, for which Toeplitz matrices are important.

When dealing with Toeplitz matrices, the function that generates the matrix is an important tool. In the one-level case, it is defined as a complex function $f : [-\pi, \pi) \to \mathbb{C}$ whose Fourier coefficients are the elements of T,

$$B_j = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(\theta) e^{-ij\theta} d\theta, \quad j = -n+1, \dots n-1, \tag{3.2}$$

and it gives important information on e.g. the spectrum of T, see [19] by Grenander and Szegő. If f is real, then T is Hermitian, and if f is even, then T is symmetric. The concept in (3.2) is easily generalized to several dimensions, corresponding to multilevel block Toeplitz matrices, and important theorems in [19] are generalized in different directions in e.g. [53].

The discrete fundamental solution, as defined in (2.4), is related to f in (3.2) since the symbol of P in (2.5) both generates the same Toeplitz matrix as f, and is used to compute the fundamental solution E. In other words, $\hat{E}_k$ is equal to $1/f$ evaluated at discrete points, if f is nonzero everywhere. That $\hat{B}$ generates the same matrix as f is seen by considering $\hat{B}_k$ not as a function defined on a grid, but as a function of a continuous variable, $\hat{B}(\theta), \theta \in [-\pi, \pi)$, using the substitution $\theta = \pi k/n$. Integrating as in (3.2) and using (2.5) for $d = 1$,

$$\begin{aligned} \frac{1}{2\pi} \int_{-\pi}^{\pi} \hat{B}(\theta) e^{-ij\theta} d\theta &= \frac{1}{2\pi} \int_{-\pi}^{\pi} \sum_{l=-n+1}^{n-1} B_l e^{-il\theta} e^{-ij\theta} d\theta \\ &= \sum_{l=-n+1}^{n-1} \frac{1}{2\pi} B_l \int_{-\pi}^{\pi} e^{-i(l+j)\theta} d\theta = B_{-j}, \end{aligned} \tag{3.3}$$

we find that $\hat{B}(-\theta)$ generates T. The reason for reversing the direction is that we consider the sequence B_j of elements in T to live in physical domain, whereas the common view when dealing with Toeplitz matrices is to consider it as Fourier coefficients of f, i.e. to exist in Fourier domain.

3.1 Matrix Algebra Preconditioners

In his proposal to use PCG for Toeplitz systems, Strang [59] also suggested using circulant matrices as preconditioners. The motivation for the suggestion is that all $n \times n$ circulant matrices are diagonalized by the Fourier matrix $\mathscr{F}$, where

$$\mathscr{F}_{ij} = \frac{1}{\sqrt{n}} e^{2\pi ijk/n}. \tag{3.4}$$

Since $\mathscr{F}$ is unitary, the diagonalization of the circulant matrix C takes the form

$$C = \mathscr{F}^* \Lambda \mathscr{F}, \tag{3.5}$$

where Λ is a diagonal matrix with the eigenvalues of C as diagonal entries. As a consequence of the diagonalization, C is easily inverted,

$$C^{-1} = \mathscr{F}^{*}\Lambda^{-1}\mathscr{F}, \tag{3.6}$$

if possible. It is easy to realize that the inverse of a circulant matrix is also circulant.

Note that neither the Fourier matrix nor the preconditioner need to be explicitly formed. Instead, $\mathscr{F}$ and $\mathscr{F}^{*}$ are applied using FFT and Λ is stored as a vector, which is computed by applying FFT on the first column of C. This column determines the circulant matrix completely, and can be considered as a representation of a periodic difference operator. Since the vector containing the diagonal of Λ is computed as a Fourier transform, or symbol, of this difference operator, it essentially represents the function generating C.

The circulant preconditioning approach can be generalized by considering other unitary matrices that can be applied by a fast transform, such as any of the eight discrete sine and cosine transforms. A set of matrices that are diagonalized by the same matrix forms an algebra, and the set of circulants is one example.

Although the matrix algebra preconditioners work well for many Toeplitz matrices, they are not well suited for the generalization to the multilevel case. The unitary matrices can still be applied quickly and the members of the algebra can hence be inverted efficiently, when possible, but their performance as preconditioners is degraded. A series of negative results are found in [45, 55, 57].

3.2 Inverse Toeplitz Preconditioners

The preconditioners that, from a linear algebra point of view, are closest related to ours, are the inverse Toeplitz preconditioners described in [24] by Hanke and Nagy, and in [8] by R. Chan and Ng. The main similarity is that the approach used in both papers is to approximate T^{-1} by a Toeplitz matrix. Both also come across the problem to invert a function with zeros. It is important to note that the inverse of a finite Toeplitz matrix is not Toeplitz in general, although the inverse of an infinite Toeplitz is. It is also important, at least when analyzing asymptotic properties of the preconditioned systems, that if the infinite Toeplitz matrix T is generated by a function f with zeros of order one or higher, there is no function generating T^{-1} in the usual sense.

The main idea by Hanke and Nagy is to embed the matrix T, which is assumed to be a banded Toeplitz matrix, in the smallest circulant matrix that contains T as the principal submatrix,

$$\tilde{T} = \begin{bmatrix} T & T_{12} \\ T_{21} & T_{22} \end{bmatrix}, \tag{3.7}$$

in the same fashion as when a Toeplitz matrix is applied by transform technique. The inverse of $\tilde{T}$ is partitioned similarly,

$$\tilde{T}^{-1} = \tilde{K} = \begin{bmatrix} K & K_{12} \\ K_{21} & K_{22} \end{bmatrix}, \tag{3.8}$$

and the matrix K is used as a preconditioner for T. The relation (3.6) is used to factorize $\tilde{T}$. In the iterative method, it is the action of K that is needed, i.e. Ku, which is the first n components of

$$\tilde{K} \begin{bmatrix} u \\ 0 \end{bmatrix} = \tilde{\mathscr{F}}^* \tilde{\Lambda}^{-1} \tilde{\mathscr{F}} \begin{bmatrix} u \\ 0 \end{bmatrix}. \tag{3.9}$$

If zeros occur on the diagonal of $\tilde{\Lambda}$, the inverse cannot be computed. The fix then is to replace $\tilde{\Lambda}^{-1}$ in (3.9) by another diagonal matrix $\tilde{\Lambda}^{-}$, given by

$$\tilde{\Lambda}^{-}_{jj} = \begin{cases} 1/\tilde{\Lambda}_{jj}, & \tilde{\Lambda}_{jj} > 0, \\ 0, & \tilde{\Lambda}_{jj} \leq 0. \end{cases} \tag{3.10}$$

The reason for replacing also inverses of negative eigenvalues is that Hanke and Nagy uses PCG as iterative method and hence need a positive definite preconditioner, which they prove K to be.

Both theory and numerical experiments show that PCG converges in a small and constant number of iterations when the method is applied to banded Toeplitz matrices, both when $\Lambda^{-1} \neq \Lambda^{-}$ and when there is equality. The former case corresponds to that there are zeros in the function that generates $\tilde{T}$. Two-level banded block Toeplitz matrices with banded Toeplitz blocks are also considered. Numerical experiments where the coefficient matrix is the two-dimensional discrete Laplacian show a number of iterations that increases with the number of unknowns.

In [8], R. Chan and Ng propose using the $n \times n$ Toeplitz matrix generated by $1/f$ as a preconditioner for T. They motivate the choice by a lemma that roughly says that if T is a (lower or upper) triangular nonsingular Toeplitz matrix generated by f, then T^{-1} is generated by $1/f$.

The evaluation of (3.2) with f replaced by $1/f$, is a problem in itself, since f might be unknown or given on a form which makes the integration complicated, or it might have zeros. The remedy used by R. Chan and Ng is to approximate the integral by a sum and f by a convolution product of a kernel function and f,

$$\frac{1}{2\pi}\int_{-\pi}^{\pi}\frac{e^{-ij\theta}}{f(\theta)}d\theta \approx \frac{1}{sn}\sum_{k=0}^{sn-1}\frac{e^{-2\pi ijk/sn}}{(\mathscr{K}*f)(2\pi k/sn)}, \quad j=-n+1,\dots,n-1. \tag{3.11}$$

Kernel functions $\mathscr{K}$ such as the Dirac kernel, the Dirichlet kernel or the Fejér kernel are used, see e.g. [61]. Their role is essentially to provide an approximation of f without explicit knowledge of f itself. The positive integer s is used to parameterize the family of preconditioners.

Experiments show fixed numbers of PCG-iterations for various Toeplitz matrices, but increasing numbers for a full matrix generated by $f(\theta)=\theta^4$. Two-level systems are not considered.

In the case where the Dirichlet kernel is used and $s=2$, the preconditioner obtained by R. Chan and Ng is similar to the one we obtain in Paper III, provided $\mathscr{K}*f$ is not zero for any $k=-n+1,\dots,n-1$. In case a zero occurs, Chan and Ng replaces its inverse by zero, much like Hanke and Nagy. This is different from our approach in Paper III, since we instead form the preconditioner so that zeros do not occur.

3.3 Band Toeplitz Preconditioners

One way to precondition a Toeplitz matrix, generated by a function f that has zeros, is to use the inverse of another Toeplitz matrix, generated by a function $\tilde{f}$. If $\tilde{f}$ is a trigonometric polynomial of low degree, and whose zeros coincide with the zeros of f, the preconditioner is the inverse of a narrow banded matrix, and fast direct methods can be used for the preconditioner solve. Band Toeplitz preconditioners are mainly applicable if the matrix T is dense, or with an increasing number of nonzero diagonals, since if it were narrow banded, the direct solver would serve better by being used on T directly.

In [7], R. Chan constructs a preconditioner for one-level matrices generated by a function f with a zero of order 2ν at $\theta=\theta_0$. The preconditioner M is generated by $\tilde{f}(\theta)=(2-2\cos(\theta-\theta_0))^\nu$, i.e. by a low degree trigonometric polynomial whose zero match the zero of f, with respect to both position and order. Serra generalizes to two-level systems in [53]. In [54], he suggests various strategies to approximate f by $\tilde{f}$, and applies the ideas to one-level systems.

A band Toeplitz preconditioner that is an exception in the sense that it can be successfully used for narrow banded systems is the one introduced in [13] by Favati, Lotti, and Menchi. It is applicable since the authors provide the Cholesky factors of the preconditioner.

The concept of band Toeplitz preconditioning can be modified, and in [46], Noutsos and Vassalos propose a more involved approximation of f, resulting in a preconditioner that is a product of three band matrices. A combination of Toeplitz and inverse Toeplitz preconditioning is developed in [4] by Di Benedetto, Fiorentino, and Serra. The preconditioned matrix is $M^{-1}KT$, where T is generated by some f and M by a low degree trigonometric polynomial $\tilde{f}$. K is generated by an approximation of $\tilde{f}/f$ which is bounded since $\tilde{f}$ is chosen so that its zeros match the zeros of f, as described above.

One drawback of band Toeplitz preconditioners is that difficulties occur in the generalization to the multilevel case. The definition of the preconditioner might be straight forward, but for the preconditioner solve, the direct band solvers are no longer fast enough. One possibility proposed in [53] is using multigrid, and in [14], Fiorentino and Serra introduce a multigrid method with an arithmetic complexity that is linear in the number of unknowns, if the Toeplitz matrix is banded and generated by a nonnegative function. Another possibility is the use of fast transforms for the preconditioner solve, see e.g. [25], but that imposes limitations on the construction of the preconditioner.

3.4 Green's function preconditioners

Another related work is that on Green's function preconditioners, since theory on PDE, and especially on fundamental solutions and Green's functions is used to devise efficient preconditioners for linear systems, arising from discretizations of partial differential operators. The method is developed by Kay, Wathen, and Loghin, and can be studied in Loghin's thesis [40], where further references are found.

Finite elements on unstructured meshes are used to discretize differential operators, resulting in unstructured matrices. The unstructured meshes make it difficult to compute discrete fundamental solutions. Instead, knowledge about properties of the fundamental solution is used to motivate a different approach. Assuming that the numbering of the unknowns follow the flow, and using information on the decay properties of the fundamental solution (and the Green's function), inverses of lower or block-lower triangular matrices are proposed as preconditioners.

The approach that yields the best behavior for a scalar two dimensional advection-diffusion problem is one where the preconditioner is a matrix arising from a discretization of a differential operator. The difference between the stiffness matrix and the inverse of the preconditioner lies in the discretization process, both in the choice of the bilinear form and in the choice of the test functions (which are chosen to possess decay properties similar to those of the Green's function). Also the boundary conditions are by necessity modified.

For a discretized *system* of PDE, it is in general impossible to find a numbering of the unknowns that follow the flow, even if the system is convection dominated. The reason is that different components of the PDE might have opposite flow directions. For the Green's function preconditioners this implies that there is no block-lower triangular matrix that approximates the stiffness matrix well enough. To deal with systems of PDE, Schur complement preconditioning is used in order to obtain scalar problems, for which either the Green's function preconditioner or other, standard preconditioners are applicable, see e.g. [36]. The knowledge about Green's functions is also used when proposing an explicit form of an approximation of the inverse of the Schur complement.

One drawback is that for each new differential equation, a considerable amount of work might be needed to formulate the preconditioner. However, when that work is done, the preconditioners obtained are easily constructed as well as efficiently applied for a large class of domains and discretizations. Furthermore, numerical experiments show that the number of iterations required to obtain a fixed reduction of the residual norm actually decreases as the number of unknowns increases, even for quite realistic problems such as Navier–Stokes equations in two dimensions.

3.5 Semicirculant Preconditioners

One class of problems for which the circulant preconditioners fail is discretizations of differential equations which are ill-posed when periodic boundary conditions are imposed in all dimensions. However, imposing periodicity in all but one dimension might result in a well-posed problem. This is the same idea that we use when constructing the discrete fundamental solution in Paper II, but here it is applied to the difference approximation of the PDE directly. A discretization of a PDE such as the one described, results in a matrix with multilevel block structure. All levels corresponding to periodic dimensions will also have circulant structure, implying that the preconditioner solve can be efficiently executed using transform methods.

A framework has been developed by Holmgren and Otto, [28, 30, 31], and the method shows favorable convergence properties for both the linearized Navier–Stokes equations for small Reynolds numbers [27] and the nonlinear Euler equations [5]. In Paper III, we compare our method with a semicirculant preconditioner.

It is possible to modify the idea behind semicirculant preconditioners by replacing the periodic boundary conditions with others, as long as there are fast transforms available. In [25] and [26], Hemmingsson uses fast sine transforms to construct preconditioners for block Toeplitz matrices. Another example is [48] by Otto and Larsson, where the Helmholtz equation is solved. In [47] by Otto, a tensor framework is developed in which preconditioners for discretizations of multidimensional PDEs are constructed. The preconditioners are generalizations of semicirculants in the sense that any combination trigonometric transforms can be applied in different dimensions.

One way to motivate the use of semicirculant preconditioners is by the negative results on matrix algebra preconditioners mentioned in section 3.1. If the problem at hand is a constant coefficient partial difference equation, the semicirculant approach is a way of modifying the fully circulant approach. The semicirculants do not belong to a unitary algebra, and hence, by violating the conditions for the negative results, good results are again possible. Additional beneficial properties of the semicirculant preconditioners are that coefficients that vary in one space dimension only, and boundary conditions in that dimension, can be efficiently accounted for. When these properties are utilized, the statement that semicirculants are modifications of circulants is a little more farfetched.

4 Review of Boundary Methods

Approximate solutions of PDEs are usually determined by some numerical method that involves a discretization of the domain on which the PDE is posed. The unknowns of the algebraic, or discrete, equations are in one way or another related to points distributed all over the domain of interest. Such methods can be referred to as domain methods, and the most common are finite element (FEM), finite difference (FDM), and finite volume (FVM) methods.

This chapter is a survey of another class of discrete methods, where the unknowns of the algebraic equations are related to the boundary of the domain only. This class will be called *boundary methods*, a term used e.g. by Brebbia and Walker in [6] and in [12]. The purpose of the survey is to relate the boundary summation equations presented and used in Paper IV and V to the literature.

Several aspects are deliberately left out, such as parallelization issues and analytical details such as which Sobolev spaces to use. Not much is said about the complexity of the methods either. The PDEs are assumed to be linear and to have constant coefficients, unless stated otherwise. Also, only interior boundary value problems are considered, although exterior problems constitute an important class where boundary methods often have an advantage over domain methods.

4.1 Boundary Integral Equations

Boundary integral equations, BIEs, are not solution methods for PDEs, but make the ground on which boundary element methods and the method of fundamental solutions build. The PDEs considered here are defined on some domain, and they can be reformulated into an integral equation defined on the boundary of the same domain, i.e. a BIE. These integral equations can be used when proving e.g. uniqueness of PDE solutions, such as in [37] by Kress, but they can also be discretized and used as numerical methods for approximating solutions of PDEs.

One classification of BIEs, used e.g. by Golberg in [15], is as either *direct* or *indirect*. In the direct equations, the unknown is the function u, i.e. the solution of the PDE, or the normal derivative of u, at the boundary. If that is not the case, if the unknown is related to u in some other way, the equations are called indirect. A more general classification of integral equations is as Fredholm's first or second kind [62]. A Fredholm equation is an integral equation where the domain Γ of the integral is known and fixed, and it is said to be of the first kind if it is of the form

$$\int_\Gamma k(x,y)\phi(y)ds(y) = f(x),$$

and of the second kind if it is of the form

$$\phi(x) = f(x) + \lambda \int_\Gamma k(x,y)\phi(y)ds(y),$$

where ϕ is the unknown and k is the kernel.

Different BIEs can be derived from the same PDE and there are different ways to derive them. In textbooks, this is usually done for the Laplace equation in the plane with mixed boundary conditions,

$$\begin{cases} \Delta u(x) = 0, & x \in \Omega \subset \mathbb{R}^2, \\ u(x) = f(x), & x \in \Gamma_1, \\ \frac{\partial}{\partial n} u(x) = g(x), & x \in \Gamma_2, \end{cases} \tag{4.1}$$

where $\Gamma_1 \cup \Gamma_2 = \partial\Omega = \Gamma$, $\Gamma_1 \cap \Gamma_2 = \emptyset$, and n is the outward pointing unit normal. The domain Ω is open. It is an understatement to say that this is a well studied problem, since the broad field of potential theory is the study of solutions to the Laplace equation [62]. Two approaches to derive BIEs from (4.1) are described by Kress [37]. The first is to express the solution as a *single* or *double layer potential* using the fundamental solution E of the Laplace operator. The resulting BIEs are indirect.

The single layer potential,

$$u(x) = \int_\Gamma \psi(y)E(x-y)ds(y), \quad x \in \Omega, \tag{4.2}$$

is shown to be a solution of (4.1) with $\Gamma_1 = \emptyset$ (Neumann problem), provided that ψ satisfies

$$\kappa(x)\psi(x) + \int_\Gamma \psi(y)\frac{\partial E(x-y)}{\partial n(x)}ds(y) = g(x), \quad x \in \Gamma. \tag{4.3}$$

The function κ is related to the so-called jump conditions and will be treated briefly later on. The double layer potential,

$$u(x) = \int_\Gamma \phi(y)\frac{\partial E(x-y)}{\partial n(y)}ds(y), \quad x \in \Omega, \tag{4.4}$$

is shown to be a solution of (4.1) with $\Gamma_2 = \emptyset$ (Dirichlet problem) if ϕ satisfies

$$\kappa(x)\phi(x) - \int_\Gamma \phi(y)\frac{\partial E(x-y)}{\partial n(y)} ds(y) = -f(x), \quad x \in \Gamma. \tag{4.5}$$

The other approach in [37] is to use Green's formula for (4.1),

$$u(x) = \int_\Gamma \left(\frac{\partial u}{\partial n}(y)E(x-y) - u(y)\frac{\partial E(x-y)}{\partial n(y)} \right) ds(y), \quad x \in \Omega, \tag{4.6}$$

implying that u can be determined in all of Ω once u and its normal derivative are determined on all of Γ. Letting x tend to the boundary yields

$$\kappa(x)u(x) = \int_\Gamma \left(\frac{\partial u}{\partial n}(y)E(x-y) - u(y)\frac{\partial E(x-y)}{\partial n(y)} \right) ds(y), \quad x \in \Gamma. \tag{4.7}$$

If the Neumann data are given on $\Gamma_2 = \Gamma$, the equation

$$\kappa(x)u(x) + \int_\Gamma u(y)\frac{\partial E(x-y)}{\partial n(y)} ds(y) = \int_\Gamma g(y)E(x-y)ds(y), \quad x \in \Gamma \tag{4.8}$$

can be solved to determine u on Γ. If instead Dirichlet data are given, one can take the normal derivative of (4.7) to obtain

$$\begin{aligned} \kappa(x)\frac{\partial u}{\partial n}(x) - &\int_\Gamma \frac{\partial u}{\partial n}(y)\frac{\partial E(x-y)}{\partial n(x)} ds(y) \\ = -\frac{\partial}{\partial n}&\int_\Gamma f(y)\frac{\partial E(x-y)}{\partial n(y)} ds(y), \quad x \in \Gamma, \end{aligned} \tag{4.9}$$

which is solved for the function $\partial u/\partial n$. In the differentiation of (4.7) some additional properties of the jump conditions are used to obtain (4.9). Both (4.8) and (4.9) are direct BIEs.

The function κ related to the jump conditions is

$$\kappa(x) = 1 - \frac{\alpha(x)}{2\pi}, \quad x \in \Gamma, \tag{4.10}$$

where $\alpha(x)$ is the interior angle of the boundary at x. It is supposed that $0 < \alpha < 2\pi$, i.e. boundaries with cusps are ruled out.

4.2 Boundary Element Methods

The most successful boundary method is the boundary element method, BEM. It has been efficiently applied to a wide variety of problems, ranging from the Laplace equation to elasticity, electromagnetic, and flow problems. It is a method for discretizing integral equations, not necessarily originating from

PDEs. If the underlying problem is a PDE, it is required that a fundamental solution is known explicitly. Fredholm equations of both kinds are included in the framework if the integral equation is formulated as

$$\lambda v(x) + Kv(x) = f(x), \quad x \in \Gamma, \tag{4.11}$$

where K is an integral operator

$$Kv(x) = \int_{\Gamma} k(x,y)v(y)ds(y).$$

The function $k(x,y)$ is called the kernel of K.

The two most popular versions of BEM are based on *collocation* and the *Galerkin formulation* respectively. Both can be applied to any of the BIEs presented in section 4.1 and both use finite dimensional subspaces to represent the approximate solution. The basis functions spanning these spaces are supposed to have local support. It is of course possible to use basis functions with global support, but then the resulting method is not called a BEM [6].

The collocation version requires that the integral equation is satisfied exactly at some collocation points $\{x_i\}, i = 1, \dots, n$, for all functions

$$v_h = \sum_{j=1}^{n} v_j \phi_j, \tag{4.12}$$

in the finite dimensional subspace Φ. Here, v_j is the scalar coefficient corresponding to ϕ_j, the j'th basis function of Φ. Hence,

$$\lambda \sum_{j=1}^{n} v_j \phi_j(x_i) + \left(K \sum_{j=1}^{n} v_j \phi_j \right)(x_i) = f(x_i), \quad i = 1, \dots, n. \tag{4.13}$$

This is a square linear system where the v_j's are unknown and whose coefficient matrix A has elements

$$A_{i,j} = \lambda \phi_j(x_i) + \int_{\Gamma} k(x_i,y)\phi_j(y)ds(y).$$

The general Galerkin formulation uses two finite dimensional subspaces, or two different bases for the same subspace. It is required that the projection of the integral equation onto one of the subspaces is satisfied for all functions in the other subspace. By assuming that the two spaces are one and the same, and that it is spanned by both $\{\phi_i\}$ and $\{\psi_i\}$, $i = 1, \dots, n$, this can be expressed

$$\lambda(v_h, \psi_i) + (Kv_h, \psi_i) = (f, \psi_i), \quad i = 1, \dots, n. \tag{4.14}$$

Using (4.12) and specifying the scalar product in the usual way, the result is a linear system for v_i with matrix elements

$$A_{i,j} = \lambda \int_\Gamma \phi_j(x)\psi_i(x)ds(x) + \int_\Gamma \int_\Gamma k(x,y)\phi_j(x)\psi_i(y)ds(x)ds(y).$$

Often, only one set of basis functions is used, i.e. $\phi_i = \psi_i$.

The Galerkin formulation is more expensive to use, since a double integral must be evaluated for each matrix element, as opposed to the collocation formulation, where only single integrals are needed. Galerkin is popular anyway, since its analytical properties are better understood.

Convergence theorems for (4.13) and (4.14) are available e.g. in [15] (and several references therein) expressed for more general integral equations. The theorems are specialized, yielding estimates of the type

$$||v - v_h||_\infty \leq ch^{\min(r,l+1)}, \tag{4.15}$$

where v is the exact solution, v_h is the solution of (4.14), c is a constant, h is the maximal size of the elements used to partition the boundary, r is the number of continuous derivatives of v, and l is the degree of the piecewise polynomials used for the basis functions.

Estimates such as (4.15) assume that the integrals determining matrix elements and right hand sides are evaluated exactly. This is almost never the case in applications. Instead they are evaluated using quadrature rules, and the resulting methods are referred to as *discrete Galerkin* or *discrete collocation*. Convergence theorems for discrete Galerkin methods can be found in [2] by Atkinson and Bogomolny. Error estimates for the interior of the domain seem quite rare, possibly due to that the evaluation of e.g. a potential in many interior points is costly. If the solution is required all over the domain, other methods than BEM might be preferable.

A more direct approach than collocation or Galerkin is the *Nyström method*, where the integral operator K is discretized using a quadrature rule. The linear system to be solved is then

$$\lambda v_i + \sum_{j=1}^{n} w_j k(x_i, y_j) v_j = f(x_i), \quad i = 1, \ldots, n, \tag{4.16}$$

where w_j are quadrature weights, x_i and y_j are quadrature points and v_i approximates $v(x_i)$. The Nyström method is cheaper than both collocation and Galerkin, but its error analysis is more involved [37].

Efforts have been made to construct quadrature rules that speed up the assembly of Galerkin matrices, which is often the most expensive part of the computations. In [17], Graham, Hackbush, and Sauter uses different quadrature depending on the local properties of the kernel, yielding what is called a *Hybrid Galerkin method*. The fact that most of the matrix entries are integrals of smooth parts of the kernel is used to motivate the use of a simpler quadrature, while more accurate methods are used for singular, or nearly singular parts of the kernel. This proposed hybrid method fits into the framework developed in [16] by the same authors.

The main advantage of BEM is the small number of unknowns, compared to FEM or FDM. Nevertheless, the linear systems that arise when BEM is applied to application problems are often too large to allow for direct solution methods. Then iterative procedures such as GMRES can be used, e.g. as in [3]. In [39] it is reported that GMRES can encounter numerical problems in its orthogonalization procedure, and that re-orthogonalization together with a higher precision in the arithmetic is needed. Multigrid is also used, see references in [37] and [15].

Iterative methods have the advantage that the coefficient matrix is not needed explicitly. Only the action of the matrix applied to a vector is used, and for BEMs, this can be obtained using methods for fast multiplication by dense matrices, such as the the ones described e.g. by Greengard and Rokhlin[18], by Hackbush and Novak [23], and by Tyrtyshnikov[60]. The use of algebraic multigrid preconditioners has been proposed by Langer, Pusch, and Reitzinger in [38]. There is also a fast direct solver by Martinsson and Rokhlin [41].

When searching for iterative methods for integral equations, it is useful to know that the term *iterative* is used also for a technique to enhance the accuracy of the BEM itself, e.g. in [15].

If the PDE is inhomogeneous, the BIE contains an integral over all of the domain Ω. There are no unknowns in the domain integral, but it must be evaluated or eliminated in order to construct a right hand side for the BEM. Four methods dealing with this problem are presented and compared by Ingber, Mammoli, and Brown in [34], where it is found that the most efficient method is to evaluate the integral using a multipole technique.

The fact that a fundamental solution is used explicitly in BEM for PDE problems is considered as one of its most serious drawbacks. Attempts to circumvent this is presented in [11] and [32], where Fourier transforms of differential operators are used. None of these attempts seem to have reached a broader audience.

4.3 Method of Fundamental Solutions

The method of fundamental solutions, MFS, is a method for finding approximate solutions to PDEs, preferably elliptic, homogeneous ones such as (4.1). As BEM, it relies on that a fundamental solution is explicitly known. The approximation of the solution is expanded in an ansatz as a linear combination of n translates of the fundamental solution,

$$u_h(x,c,Y) = \sum_{j=1}^{n} c_j E(x - y_j), \quad x \in \bar{\Omega} = \Omega \cup \Gamma, \tag{4.17}$$

where $c = \{c_j\}, j = 1,\ldots,n$ are (unknown) coefficients and $Y = \{y_j\}$ is a set of points *outside* $\bar{\Omega}$. The y_j are called the locations of the singularities of the fundamental solutions[1], or simply the singularities. One way to determine the coefficients c_j is by applying the boundary conditions to the ansatz, and minimizing the l_2 norm of the result,

$$F(c,Y) = \sum_{i=1}^{m} |Bu_h(x_i,c,Y) - f(x_i)|^2, \tag{4.18}$$

with respect to c and sometimes also with respect to Y. Here, $x_i \in \Gamma, i = 1,\ldots,m$, are the *observation points*, B is the operator representing boundary conditions, and f contains boundary data. The survey [12] by Fairweather and Karageorgis is focused on this approach. Alternative approaches are the Galerkin or the collocation method, where $n = m$ yields square systems. In the collocation method, the positions of the collocation points are important. A rule applicable to potential problems is given by Katsurada and Okamoto in [35].

The entries of the coefficient matrix for the collocation method applied to (4.1) is

$$A_{i,j} = \begin{cases} E(x_i - y_j), & x_i \in \Gamma_1, \\ \frac{\partial}{\partial n(x)} E(x_i - y_j), & x_i \in \Gamma_2, \end{cases} \tag{4.19}$$

and the right hand side contains samplings of the boundary data f.

The reason for placing y_j outside $\bar{\Omega}$ is to avoid $x_i = y_j$, i.e. the case when $E(x_i - y_j)$ is singular for elliptic equations. If the minimization of (4.18) is performed also with respect to Y, i.e. if the singularities are allowed to move, the minimization problem is nonlinear. This approach was proposed by Mathon and Johnston in [42]. It is reported that the increased complexity due to that a nonlinear optimization routine is needed is compensated for by the higher quality of the approximation. Also, problems with nonlinear boundary conditions are easily treated in the framework with moving singularities.

Inhomogeneous PDEs are not easily treated with MFS. The remedy is to reformulate the problem by subtracting a particular solution, so that the forcing function is eliminated. This is done by Poullikkas, Karageorghis, and Geor-

[1] This seems inadequate in our notation, since fundamental solutions of elliptic operators are singular only at the origin. However, in the MFS-literature it is common to denote the fundamental solution as a function of two variables, $\Phi(x,y) = E(x-y)$, and then it is more natural to call y the singularity, since Φ, as a function of x is singular at y.

giou in [49], where also the special case with a harmonic right hand side is examined.

The convergence of MFS can be very rapid, at least for some special problems. If the domain Ω is a circle of radius r, and the singularities are placed on a concentric circle with radius $R > r$, the collocation version can yield

$$\sup_{x\in\Omega} |u(x) - u_h(x)| = \mathscr{O}((r/R)^n), \tag{4.20}$$

i.e. an exponential convergence in the number of unknowns, see [58] by Smyrlis and Karageorghis, and the references therein.

The coefficient matrix obtained in the collocation method can be very ill-conditioned, hence the sensitivity with respect to the location of the singularities, see e.g. [1] by Ahmed, Lavers, and Burke. In [50] by Ramachandran, a solution method based on the singular value decomposition is proposed, that partly eliminates this sensitivity. The SVD technique used in a reservoir simulation is described in [20] by Guevara-Jordan and Rojas.

Iterative methods are used also for MFS, see [52] by Saavedra and Power, where the fast multipole method is used.

4.4 Method of Difference Potentials

This and the next section will deal with partial difference equations, and some additional notation is needed. The difference equations we consider can be written

$$\begin{cases} \sum_{j\in S} P_j u_{i-j} = f_i, & i \in \Omega \subset \mathbb{Z}^d, \\ \sum_{j\in S_i} B_{i,j} u_{i-j} = f_i, & i \in \Gamma, \end{cases} \tag{4.21}$$

where P_j are $n_c \times n_c$ matrices and can be seen as stencil weights, S is the set of displacements for which P_j is nonzero, u_i is the unknown vector valued function with n_c components for each $i \in \bar{\Omega} = \Omega \cup \Gamma \subset \mathbb{Z}^d$. The indices j will occasionally be referred to as grid points. Both n_c and d are small positive integers. The set Γ is defined to contain all grid points where the equations are different from the first row of (4.21) and the S_i are such that $j \in S_i$ implies $i - j \in \bar{\Omega}$. The n_c-matrices $B_{i,j}$ represent boundary conditions and are allowed to vary both with i and j.

The method of difference potentials, MDP, is described in the book [51] by Ryaben'kii in terms of operators operating on grid functions. It is developed as a discrete counterpart to potential theory, but covering more than just discretizations of the Laplacian. The basics are reviewed here, translated to a notation making the comparison to other methods easier.

The idea in MDP is to transform a given difference equation with constant coefficients and a set of boundary conditions into an auxiliary problem with less complicated boundary conditions and with a modified right hand side. The auxiliary problem is posed on a larger domain that contains the original one. Let the original problem be (4.21) with $f_i = 0, i \in \Omega$ and $d = 2$. Then an admissible auxiliary problem would be

$$\begin{cases} \sum\limits_{j \in S} P_j \tilde{u}_{i-j} = \tilde{f}_i, & i \in \Omega^+, \\ \sum\limits_{j \in \tilde{S}_i} \tilde{B}_j \tilde{u}_{i-j} = \tilde{f}_i, & i \in \Gamma^+, \end{cases} \tag{4.22}$$

where Ω^+ is a square that contains Ω and is large enough so that $\Gamma^+ \cap \Gamma = \emptyset$. The boundary conditions are chosen such that the auxiliary problem is nonsingular and easy to solve, e.g. with some fast transform technique. The modified right hand side is defined to be

$$\tilde{f}_i = \begin{cases} \sum\limits_{j \in S} P_j v_{i-j}, & i \in \Omega^- \equiv \bar{\Omega}^+ \setminus \Omega, \\ 0, & i \in \Omega, \end{cases} \tag{4.23}$$

where $\bar{\Omega}^+ \equiv \Omega^+ \cup \Gamma^+$ and

$$v_i = \begin{cases} v_{\gamma,i}, & i \in \gamma, \\ 0, & i \notin \gamma. \end{cases} \tag{4.24}$$

The set γ contains (i):the grid points in Γ, (ii):the grid points in Ω for which the stencil touches Γ, and (iii):the grid points touched by the stencils in the boundary conditions on Γ. Figure 4.1 might be helpful when identifying the sets. In the figure, it is assumed that the boundary Γ of the original problem is a square that follows the lines on the grid. It is also assumed that the ordinary five-point stencil is used in the interior and that the stencils S_i on the boundary do not reach further into the domain than the interior stencil would.

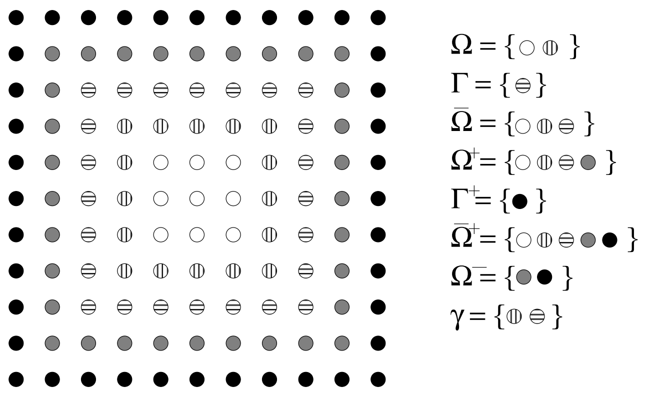


Figure 4.1: The sets used in MDP.

The solution $\tilde{u}$ to the auxiliary problem is called *difference potential* and it is shown to coincide with the original solution on $\bar{\Omega}$. The problem of finding u is reduced to the problem of finding v_γ, which is called the *density* of the potential.

Equations (4.22) to (4.24) define a mapping Q from functions defined on γ to functions defined on $\bar{\Omega}^+$. For this mapping, it is shown in [51] that any function $\tilde{u}$ satisfying the first line of (4.21), with $f = 0$, it holds that

$$Q_\gamma \tilde{u}_\gamma = \tilde{u}_\gamma,$$

where $\tilde{u}_\gamma$ is the restriction of $\tilde{u}$ to γ and Q_γ is the operator that first applies Q and then restricts to γ. Hence, the density v_γ of the potential that coincides with the solution u of (4.21) on $\bar{\Omega}$, is itself coinciding with u on γ. The equations used to determine v_γ are

$$\begin{cases} (Q_\gamma v_\gamma)_i - v_{\gamma,i} = 0, \\ \sum_{j \in S_i} B_{i,j} v_{\gamma,i-j} = f_i, \end{cases} \qquad i \in \gamma. \tag{4.25}$$

These equations are over-determined, but consistent. The number of unknowns is reduced greatly. Equation (4.25) can be solved iteratively, since the operator Q is applied mainly by solving (4.22), which is easy and fast enough. There is no error in the density if (4.25) is solved exactly, and the potential coincides exactly with the solution of the original difference equation. The only error is in the discretization of the PDE.

In order to obtain a matrix formulation of (4.25), some additional notation is needed. Let P be the coefficient matrix representing (4.22), let n_A be the number of elements in A, n_B the number of elements in B, and let $\theta_{A,B}(C)$ be the matrix with n_A rows and n_B columns that satisfies

$$\theta_{A,B}(C)v_i = \begin{cases} v_i, & i \in C, \\ 0, & i \in A \setminus C, \end{cases} \quad \forall v = \{v_j\}, j \in B, \tag{4.26}$$

where A, B, and C are sets of grid points. Then, $\theta_{A,A}(A)$ is the identity matrix of size n_A. Now, the matrix representing Q_γ in the first row of (4.25) can be written

$$\theta_{\gamma,\bar{\Omega}^+}(\gamma)P^{-1}\theta_{\bar{\Omega}^+,\bar{\Omega}^+}(\Omega^-)P\theta_{\bar{\Omega}^+,\gamma}(\gamma), \tag{4.27}$$

which is a square matrix of size n_γ. Note that the boundary conditions of the original problem (4.21) are not present in this expression. They are instead imposed by the second row of (4.25).

The fact that the density coincides with the solution on γ makes it reasonable to compare MDP with direct BIEs such as (4.6) and (4.8). A motivation for using MDP on a finite difference discretization of a PDE, instead of using BEM on the corresponding BIE is given in [21] by Gürlebeck and Hommel. It is stated that important potential theoretical properties of the BIE are lost in the BEM discretization. If on the other hand MDP is used, some properties from classical potential theory carries over to the discrete case.

Inhomogeneous problems can easily be solved by first using (4.22) to find a particular solution, that is used to eliminate the inhomogeneity and to modify the boundary data.

4.5 Boundary Summation Equations

Boundary summation equations, BSE, are related to MDP in the sense that it reduces partial difference equations into algebraic equations with less unknowns. Although inspired by theory for PDEs, it is developed in a linear algebra setting. It is presented in Paper IV, where it is also applied to a number of difference equations, representing discretizations of inhomogeneous, elliptic and non-elliptic PDEs, both systems and scalar equations.

Let the difference equation (4.21) be represented by the partitioned square linear system (2.10), let v be defined by (2.12), and let

$$\begin{pmatrix} K_\Gamma & K_{\Gamma\Omega} \\ K_{\Omega\Gamma} & K_\Omega \end{pmatrix}$$

be defined by (2.13). Then the reduced system, the BSE, is given by (2.15).

As for MDP, no approximation is performed in the reduction process, and the error in u is only due to the discretization of the PDE, provided that (2.15) is solved exactly. If it is not solved exactly, it is easy to see that the error satisfies

$$e \equiv u - u^* = \begin{pmatrix} K_\Gamma & K_{\Gamma\Omega} \\ K_{\Omega\Gamma} & K_\Omega \end{pmatrix} \begin{pmatrix} v - v^* \\ 0 \end{pmatrix},$$

where u^* is the approximation to the solution on $\bar{\Omega}$ and v^* is the approximate solution to (2.15).

To put BSE in perspective, let (4.21) represent a homogeneous PDE, i.e. let $\boldsymbol{f}_{\boldsymbol{\Omega}} = 0$ in (2.9), implying $f_i = 0$ for $i \in \Omega$. The solution is then expressed using (2.16) as

$$u_i = \begin{pmatrix} K_\Gamma & K_{\Gamma\Omega} \\ K_{\Omega\Gamma} & K_\Omega \end{pmatrix} \begin{pmatrix} v_\Gamma \\ 0 \end{pmatrix}_i = \sum_{j\in\Gamma} E_{i-j} v_{\Gamma,j}, \quad i \in \bar{\Omega}, \tag{4.28}$$

which when compared to (4.2) motivates the statement that BSE is related to the single layer potential. The most related BEM is the Nyström method in (4.16). One can also relate BSE to a collocation MFS, by comparing (4.19) and the expression for the coefficient matrix A in (2.15). The latter can be written

$$Av_i = \sum_{j\in S_i} B_{i,j} \sum_{k\in\Gamma} E_{i-j-k} v_k, \tag{4.29}$$

i.e. each element in A has a value that is computed by applying the boundary conditions to the fundamental solution. However, there is no guarantee that v_Γ in (4.28) even resembles ψ in (4.2), or that the matrix elements given by (4.29) are similar to the ones in (4.19). The reason is that the fundamental solutions of the differential operator and the difference operator can differ very much.

It is also possible to compare the coefficient matrices for BSE and MDP, by writing A in (2.15) using the θ-matrices defined in (4.26),

$$A = \theta_{\Gamma,\bar{\Omega}}(\Gamma) PK \theta_{\bar{\Omega},\Gamma}(\Gamma).$$

The sets $\bar{\Omega}$ and Γ are the same as in section 4.4, but here P represents the original difference equation (4.21). The matrix K is the one in (2.13).

Inhomogeneous equations are automatically treated in the BSE framework. For homogeneous problems, (2.15) simplifies to

$$Av_\Gamma = f_\Gamma.$$

5 Concluding remarks

The topic of the thesis is the solution of large, sparse, structured linear systems of algebraic equations. Two solution strategies are presented, the first being the construction of a preconditioner and the second an algorithm for reducing the number of unknowns of the systems. Both strategies use fundamental solutions, either of differential operators, or of difference operators.

Besides the presentation of the five papers, the thesis contains a review of preconditioners that are related to preconditioners based on fundamental solutions, and a review of boundary methods that have properties in common with the boundary summation equations.

All the linear systems of equations treated in the thesis are discretizations of PDEs, but both the fundamental solution preconditioners and the boundary summation equations can be applied to systems with other origins. All areas where banded Toeplitz or Toeplitz-like matrices are important could be considered.

In papers dealing with matrix algebra preconditioners for Toeplitz matrices, a common result for the preconditioned matrix $M^{-1}T$ in the one level case is

$$M^{-1}T = I + N + R, \tag{5.1}$$

where I is the identity, N is a matrix with small norm, and R is a matrix with low rank. In the multilevel case, the rank of R will not be independent of the size of T, and a preconditioned iterative method will degrade. If T is a banded Toeplitz matrix and M^{-1} is a preconditioner based on a fundamental solution of a difference operator, the same results seem to hold, but with $N = 0$. The boundary summation equations can roughly be described as a way to compensate for the R-matrix.

One suggestion for further research is to study approximate solvers for the boundary summation equations. By using such a solver as a preconditioner, one could possibly obtain estimates as (5.1), but where also the norm of R is small. Solving the boundary summation equations iteratively, as we do, is a step in this direction, but one could also consider making the reduction process approximate in itself.

Another suggestion is to use tools from non-Hermitian Toeplitz theory to analyze the fundamental solution preconditioners in more detail. Then, as always, there is a problem with boundary conditions, but treating the Toeplitz part, and hence one set of boundary conditions, would be a good start. However, the tools must be used with care.

It would also be interesting to explore the applicability of the fundamental solutions to other application areas, such as image deblurring. The challenge is different from the PDE case; it is easier in the sense that constant coefficients and rectangular domains are not considered as simplifications, and it is more difficult in the sense that the expectations on fast iterative solvers are higher and that there is a need to treat noise. The question of boundary conditions is another difference. For PDEs they are usually given, but for deblurring problems they can often be chosen so that the resulting problem is easy to solve. In e.g. [44, 56], boundary conditions are chosen so that the resulting matrices belong to unitary algebras for which fast inversion is available. However, the choice of boundary conditions affects the influence of noise [10], and if there is a set of boundary conditions that are good with respect to noise-treatment, but that does not put the matrix in a suitable algebra, the boundary summation equations might be useful.

6 Acknowledgement

Thank you Henrik Brandén and Sverker Holmgren for being my supervisors and for teaching me many important things and a lot of numerical analysis.

7 Summary in Swedish: Numeriska beräkningar med fundamentallösningar

Den här sammanfattningen innehåller en introduktion till den forskning som presenteras i avhandlingen, på ett språk som förhoppningsvis kan förstås utan fackkunskap i ämnet. Syftet är att relatera avhandlingens resultat, algoritmerna, till vardagsnära fenomen och förklara vad som är eftersträvansvärt när man konstruerar nya algoritmer, och varför.

Många fenomen av intresse för ingenjörer, fysiker, ekonomer, kemister och andra kan beskrivas med hjälp av *differentialekvationer*. Differentialekvationerna utgör en matematisk modell av ett fenomen och frågor om fenomenet kan ibland besvaras efter att ekvationerna lösts. Följande tabell exemplifierar fenomen som framgångsrikt har modellerats med differentialekvationer och några tänkbara frågor.

Fenomen	Frågor
Luftströmning	Hur mycket last kan ett flygplan bära? Hur mycket bränsle drar en bil?
Blixtnedslag	Hur ska man placera elektronik och kablar i ett flygplan för att minimera skador?
Ljudutbredning i vatten	Ubåt eller mink?
Ljudutbredning i luft	Hur nära bebyggelse kan man placera vindkraftverk?
Prissättning av aktieoptioner	När ska man sälja?
Hållfasthet	Kommer bron hålla? Hur många får gå in i hissen?
Förbränning	Behövs mer syre?
Väder	Regnar det i morgon också?

Listan kan givetvis göras mycket längre. Ett svar på en fråga är i allmänhet inte detsamma som en lösning till en differentialekvation. Ta hållfasthetsfenomenet som exempel: lösningen till differentialekvationen kan vara spänningen i varje del av materialet i bron. Svaret på frågan om bron håller får man genom att kontrollera om spänningen någonstans överskrider vad materialet klarar.

Gemensamt för fenomenen i fråga är att man vill kunna göra förutsägelser. Redan innan bron är byggd vill man veta om den kommer att hålla. När det gäller vädret inser man att det ibland kan vara bråttom att få fram ett svar. Tar det två dagar att lösa ekvationerna kan man lika gärna strunta i det, om frågan gäller morgondagens regn.

Gemensamt för fenomenen är också att de differentialekvationer som används för att modellera dem nästan aldrig kan lösas exakt. Med det menas inte att de är olösbara i princip, bara att ingen vet hur det ska gå till. Som ingenjör behöver man dock inte ge upp för det. Om man betraktar frågorna i tabellen inser man att det ofta räcker med ett ungefärligt svar. Det kan man få genom att lösa differentialekvationerna ungefärligt, eller med ett annat ord *approximativt*.

Det finns många metoder för att bestämma approximativa lösningar till differentialekvationer. Processen att modifiera och formulera om en differentialekvation till en så kallad *algebraisk ekvation* kallas *diskretisering*. Olika diskretiseringsmetoder ger i allmänhet upphov till olika algebraiska ekvationer, även om de appliceras på samma differentialekvation. Syftet är att åstadkomma ekvationer som är lättare att lösa och vars lösning säger något om lösningen till differentialekvationerna.

De algebraiska ekvationerna innehåller bara enkla operationer och är lättare att lösa i den meningen att man kan formulera algoritmer som tar fram lösningen. Med *algoritm* menas en uppsättning regler för vad som ska göras och i vilken ordning. Oftast måste väldigt många operationer utföras, det är inte ovanligt att det blir flera tusen miljarder, men eftersom de är enkla kan de utföras av en dator. Algoritmen formuleras som ett program som körs på datorn.

Lösningen till de algebraiska ekvationerna kan vara en uppsättning värden som ligger nära lösningen till differentialekvationen. Ta broexemplet igen: differentialekvationens lösning var spänningen i varje del av materialet. Lösningen till de algebraiska ekvationerna kan då vara approximationer till spänningen i ett antal punkter utspridda i materialet. Dessa punkter kan kallas *nätpunkter* och skillnaden mellan den approximativa och den exakta lösningen i nätpunkterna kallas *diskretiseringsfelet*.

Kravet att lösningen till de algebraiska ekvationerna ska likna differentialekvationens lösning innebär att ju fler nätpunkter som används, desto mindre ska diskretiseringsfelet bli. Ofta behövs väldigt många nätpunkter, miljontals, för att felet ska bli acceptabelt litet. Bara att lagra lösningen i alla nätpunkter kräver att datorn har ett stort minne. Dessutom kräver oftast algoritmen en hel del extra minne för att kunna beräkna lösningen.

Approximation är ett centralt begrepp. Om man ännu en gång tar bron som exempel så utförs den första, och kanske grövsta, approximationen när problemet modelleras med differentialekvationer. Ingenjören som modellerar gör förenklingar, som till exempel att betongen är homogen, eller att bågen under bron beskrivs av en cirkelbåge. Lösningen till differentialekvationen kommer inte att bli identisk med den verkliga spänningen i bron, men det gör inte så mycket eftersom man måste räkna med säkerhetsmarginaler. Nästa approximation utförs av numerikern som diskretiserar differentialekvationen. Felet som uppkommer då, diskretiseringsfelet, är lättare att kontrollera, och kan minskas vid behov genom att fler nätpunkter används. Ingenjören är ofta nöjd om diskretiseringsfelet är mindre än någon promille.

Eftersom både modelleringen och diskretiseringen introducerar fel så är det inte nödvändigt att lösa de algebraiska ekvationerna exakt heller. Om man nöjer sig med en approximativ lösning även av de algebraiska ekvationerna får man tillgång till många fler algoritmer än om man kräver en exakt lösning. De approximativa algoritmerna är utvecklade med tanke på minnes- och tidsåtgång, och felet de genererar är kontrollerbart. Felet kan minskas till priset av mer datorminne och tid, men det är finns ingen vinst i att tvinga ner felet så att det blir mycket mindre än diskretiseringsfelet.

Syftet med avhandlingen är att presentera två metoder för att lösa algebraiska ekvationer som uppkommer vid diskretisering av differentialekvationer. Vid första anblicken kan det framstå som en minimal detalj i arbetet att få fram ett svar på en bro- eller väderfråga, men det är en flaskhals. Om man inte kan lösa de algebraiska ekvationerna tillräckligt snabbt så kommer svaret på frågan om morgondagens väder först i övermorgon.

Hela kedjan från fenomen till algoritm kan sammanfattas med ett exempel. Den kan börja med en enkel fråga, exempelvis ”Hur mycket bensin kommer bilen dra?”, ställd av konstruktionschefen hos en biltillverkare. En ingenjör får reda ut vad som påverkar förbrukningen och bland mycket annat visar det sig att luftmotståndet är viktigt. Problemet att ta reda på luftmotståndet förenklas och modelleras med differentialekvationer, en uppgift för en fysiker med flödesproblem som specialitet. Differentialekvationerna diskretiseras av en numeriker som kan CFD, ”computational fluid dynamics”. Nästa steg är att lösa de algebraiska ekvationerna och det är här avhandlingens algoritmer kommer in. Förmodligen krävs ytterligare en länk i kedjan, där en programmerare och en datortekniker får göra sitt, för att ett riktigt storskaligt problem ska få sin lösning. Lösningen löper sedan tillbaka genom kedjan, och i varje steg gör respektive expert en tolkning och en bedömning av rimligheten hos lösningen. Ingen länk kan hoppas över och alla länkar påverkar varandra.

7.1 Iterativa metoder och förkonditionerare

Den första algoritmen i avhandlingen kan beskrivas som en modifiering av redan existerande approximativa lösare för algebraiska ekvationer. Modifieringen kallas *förkonditionerare*, på engelska ”preconditioner”, och den kan tillämpas på de flesta lösare i klassen *iterativa metoder*. Iterativa metoder utnyttjar det faktum att det är lättare att gissa en lösning till en ekvation och sedan testa den, än att lösa ekvationen exakt. Testet av den gissade lösningen ger ett resultat som kallas *residual*. Med hjälp av residualen kan man sedan modifiera sin gissning och testa igen. Proceduren upprepas, itereras, med allt fler modifieringar av gissningen, tills man är nöjd. Residualen är inte samma sak som felet, men med ledning av residualen kan man uppskatta hur stort felet är och därmed avgöra om man är nöjd med den modifierade gissningen.

Nackdelen med iterativa metoder är att det inte är säkert att felet någonsin blir så litet att man är nöjd. Det finns flera möjliga händelseförlopp varav följande tre är speciellt viktiga.

1. Felet minskar och blir tillräckligt litet tillräckligt snabbt.
2. Felet minskar, men så långsamt att det behövs väldigt många iterationer.
3. Felet växer ju mer man itererar.

Det första fallet är det gynnsamma. Man säger att metoden konvergerar snabbt och med det menar man att felet närmar sig noll snabbt nog. I det andra fallet konvergerar metoden också, men för långsamt, och i det tredje fallet säger man att metoden divergerar.

Det finns både fördelar och nackdelar med att använda iterativa metoder för att bestämma approximativa lösningar till de algebraiska ekvationer som är diskretiseringar av differentialekvationer. Den viktigaste fördelen är att det är lättare att testa en gissning, att beräkna en residual, för sådana ekvationer än för algebraiska ekvationer i allmänhet. Den viktigaste nackdelen är att det inte finns någon iterativ metod som konvergerar tillräckligt snabbt för alla diskretiseringar. Det är tvärt om så att nästan ingen iterativ metod är snabb nog för någon diskretisering alls.

Den vanligaste metoden att förbättra konvergensegenskaperna hos en iterativ metod är att använda en förkonditionerare. Det kan beskrivas som att man modifierar de algebraiska ekvationerna på ett kontrollerat sätt, så att den exakta lösningen inte påverkas. Modifieringen gör att varje iteration tar lite längre tid och att mer datorminne tas i anspråk. Om förkonditioneraren är bra blir effekten att metoden konvergerar i så mycket färre iterationer att tidsförlusten i varje iteration kompenseras. Det extra minneskravet får inte heller vara för stort. Det finns många förkonditionerare och de flesta konstrueras utifrån de algebraiska ekvationerna, utan hänsyn till varifrån ekvationerna kommer.

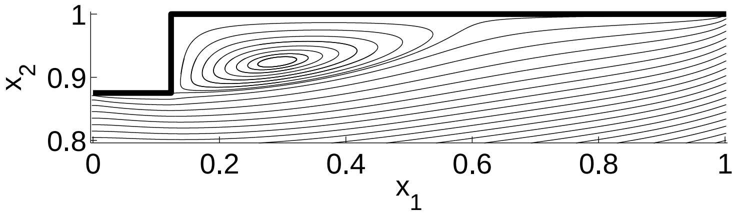
I artikel I presenteras en förkonditionerare som skiljer sig från mängden genom att den konstrueras utifrån differentialekvationen. Den diskretiseras sedan för att kunna användas tillsammans med de algebraiska ekvationerna. Den andra förkonditioneraren i avhandlingen presenteras i artikel III och konstrueras utan information om vilken differentialekvation som är diskretiserad. Den är dock inspirerad av teori för differentialekvationer och kan ses som en utveckling av den första. Artikel II beskriver ett viktigt delsteg i konstruktionen av den andra förkonditioneraren. Avhandlingens båda förkonditionerare fungerar bra för diskretiseringar av en klass av differentialekvationer där de flesta andra varianter fungerar dåligt.

7.2 Randmetoder

Vissa fenomen, till exempel blixtnedslaget och ibland luftens strömning, modelleras med differentialekvationer som kan formuleras om till *integralekvationer*. Dessa integralekvationer kan i sin tur diskretiseras till algebraiska ekvationer, vars egenskaper skiljer sig från diskretiseringar av motsvarande differentialekvationer på flera sätt. En viktig positiv egenskap är att man inte behöver använda lika många nätpunkter i diskretiseringen. En viktig negativ egenskap är att det inte är lika enkelt att beräkna en residual som det är för diskretiseringar av differentialekvationer. Det går att göra snabbt, men det kräver avancerade algoritmer.

I artikel IV beskrivs en ny metod för att modifiera vissa algebraiska ekvationer på ett sätt som påminner om hur differentialekvationer kan omformuleras till integralekvationer. Resultatet efter modifiering har jag valt att kalla *randsummationsekvationer*. Kraven som metoden ställer på de algebraiska ekvationerna är uppfyllda för en viss typ av diskretiseringar av en viss typ av differentialekvationer. Det är precis samma differentialekvationer som kan omformuleras till integralekvationer. Integralekvationernas positiva och negativa egenskaper som nämns ovan gäller också randsummationsekvationerna. Klassen av algebraiska ekvationer som kan modifieras till randsummationsekvationer innefattar också ekvationer som inte är diskretiseringar av differentialekvationer, exempelvis sådana som uppkommer i bildbehandlingsproblem när man vill rekonstruera en bild som har blivit suddig.

Randsummationsekvationerna används i artikel V för ett flödesproblem. Hastighet och tryck i en fluid (vätska eller gas) har beräknats i ett rör. Figur 7.1 visar en detalj där den tjocka svarta linjen är rörets vägg. Fluiden strömmar in från vänster och ut åt höger. De tunna linjerna är strömlinjer, och man ser att det uppkommer en virvel bakom den skarpa kanten där röret vidgas.



Figur 7.1: Strömlinjer för en fluid i ett rör.

Modellen som beskriver flödet är användbar för många olika fluider. Om man antar att fluiden är vatten så motsvarar lösningen i figuren flödet i en kanal som bara är en knapp millimeter lång. Vattnets hastighet är ungefär 1,5 meter per sekund till vänster i figuren. Om man i stället antar att fluiden är glycerol, så uppkommer samma flödeslinjer i en sju decimeter lång kanal då hastigheten är två meter per sekund.

Diskretiseringen av differentialekvationerna ger algebraiska ekvationer vars lösning består av tre värden i varje nätpunkt som man vill ha ett svar i. Lösningen i bilden har beräknats i ungefär 64 000 punkter, vilket innebär att ungefär 192 000 värden behövs. Lösningen till randsummationsekvationerna å andra sidan behöver bara beräknas i 1015 nätpunkter, vilket motsvarar 3045 separata värden. Antalet värden som måste beräknas har alltså minskats högst väsentligt med hjälp av algoritmen som presenteras i artikel IV. Trots det är båda lösningarna precis lika bra.

Avslutningsvis är det viktigt att komma ihåg att avhandlingens resultat inte är själva lösningarna med virvlar och strömlinjer, utan metoderna som använts för att beräkna lösningarna. Flödesproblemen används för att testa metoderna. Man kan jämföra med konstruktionen av ett verktyg, där resultatet exempelvis är en hammare, inte en islagen spik. Man testar hammarens egenskaper genom att slå i olika spikar i olika material. Andra uppgifter kräver andra verktyg och hammaren kan inte ersätta en skruvmejsel, även om man kan slå i skruvar också. På samma sätt behövs olika algoritmer för att lösa olika beräkningsproblem, även om det kan visa sig att en algoritm utvecklad för en viss typ av ekvationer kan användas i helt andra sammanhang också.

Bibliography

[1] M. T. Ahmed, J. D. Lavers, and P. E. Burke. An evaluation of the direct boundary element method and the method of fundamental solutions. *IEEE Trans. Mag.*, 25(4):3001–3006, 1989.

[2] K. Atkinson and A. Bogomolny. The discrete Galerkin method for integral equations. *Math. Comp*, 48(178):595–616, 1987.

[3] L. P. Barra, A. L. G. A. Coutinho, W. J. Mansur, and J. C. F. Telles. Iterative solution of BEM equations by GMRES algorithm. *Comp. Struct.*, 44(6):1249–1253, 1992.

[4] F. Di Benedetto, G. Fiorentino, and S. Serra. C. G. preconditioning for Toeplitz matrices. *Comput. Math. Appl.*, 25(6):35–45, 1993.

[5] H. Brandén and S. Holmgren. Convergence acceleration for the steady-state Euler equations. *Comput. & Fluids*, 32:1075–1092, 2003.

[6] C. A. Brebbia and S. Walker. *Boundary Element Techniques in Engineering*. Newnes–Butterworths, 1980.

[7] R. H. Chan. Toeplitz preconditioners for Toeplitz systems with nonnegative generating functions. *IMA J. Num. Anal.*, 11:333–345, 1991.

[8] R. H. Chan and K.-P. Ng. Toeplitz preconditioners for Hermitian Toeplitz systems. *Linear Algebra Appl.*, 190:181–208, 1993.

[9] C. de Boor, K. Höllig, and S. Riemenschneider. Fundamental solutions for multivariate difference equations. *Amer. J. Math.*, 111:403–415, 1989.

[10] M. Donatelli and S. Serra Capizzano. Anti-reflective boundary conditions and re-blurring. *Inverse Problems*, 21:169–182, 2005.

[11] F. M. E. Duddeck. Fourier-BEM: or What to do if no fundamental solution is available? *Meccanica*, 36:437–448, 2001.

[12] G. Fairweather and A. Karageorgis. The method of fundamental solutions for elliptic boundary value problems. *Adv. Comput. Math.*, 9:69–95, 1998.

[13] P. Favati, G. Lotti, and O. Menchi. A polynomial fit preconditioner for band Toeplitz matrices in image reconstruction. *Linear Algebra Appl.*, 346:177–197, 2001.

[14] G. Fiorentino and S. Serra. Multigrid methods for symmetric positive definite block Toeplitz matrices with nonnegative generating functions. *SIAM J. Sci. Comput.*, 17(5):1068–1081, 1996.

[15] M. A. Golberg. *Numerical Solution of Integral Equations*. Plenum Press, 1990.

[16] I. G. Graham, W. Hackbush, and S. A. Sauter. Discrete boundary element methods on general meshes in 3D. *Numer. Math.*, 86:103–137, 2000.

[17] I. G. Graham, W. Hackbush, and S. A. Sauter. Hybrid Galerkin boundary elements: theory and implementation. *Numer. Math.*, 86:139–172, 2000.

[18] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *J. Comput. Phys.*, 73:325–348, 1987.

[19] U. Grenander and G. Szegő. *Toeplitz Forms and Their Applications*. Chelsea, New York, 2nd edition, 1984.

[20] J. M. Guevara-Jordan and S. Rojas. A method of fundamental solutions for modeling porous media advective fluid flow. *Appl. Num. Math.*, 47:449–465, 2003.

[21] K. Gürlebeck and A. Hommel. On finite difference potentials and their applications in a discrete function theory. *Math. Meth. Appl. Sci.*, 25:1563–1576, 2002.

[22] B. Gustafsson and H. Stoor. Navier–Stokes equations for almost incompressible flow. *SIAM J. Numer. Anal.*, 28(6):1523–1547, 1991.

[23] W. Hackbusch and Z. P. Novak. On the fast matrix multiplication in the boundary element method by panel clustering. *Numer. Math.*, 54:463–491, 1989.

[24] M. Hanke and J. G. Nagy. Toeplitz approximate inverse preconditioners for banded Toeplitz matrices. *Numer. Algorithms*, 7:183–199, 1994.

[25] L. Hemmingsson. A fast modified sine transform for solving block-tridiagonal systems with Toeplitz blocks. *Numer. Algorithms*, 7:375–389, 1994.

[26] L. Hemmingsson. Toeplitz preconditioners with block structure for first-order PDEs. *Numer. Linear Algebra Appl.*, 3:21–44, 1996.

[27] S. Holmgren, H. Brandén, and E. Sterner. Convergence acceleration for the linearized Navier–Stokes equations using semicirculant approximations. *SIAM J. Sci. Comput.*, 21:1524–1550, 2000.

[28] S. Holmgren and K. Otto. Iterative solution methods and preconditioners for block-tridiagonal systems of equations. *SIAM J. Matrix Anal. Appl.*, 13:863–886, 1992.

[29] S. Holmgren and K. Otto. Semicirculant preconditioners for first-order partial differential equations. *SIAM J. Sci. Comput.*, 15:385–407, 1994.

[30] S. Holmgren and K. Otto. A framework for polynomial preconditioners based on fast transforms I: Theory. *BIT*, 38:544–559, 1998.

[31] S. Holmgren and K. Otto. A framework for polynomial preconditioners based on fast transforms II: PDE applications. *BIT*, 38:721–736, 1998.

[32] D. Homentcovschi and T. Singler. An introduction to BEM by integral transforms. *Engng. Anal. Boundary Elements*, 23:603–609, 1999.

[33] L. Hörmander. *The Analysis of Linear Partial Differential Operators I*. Springer-Verlag, 2nd edition, 1990.

[34] M. S. Ingber, A. A. Mammoli, and M. J. Brown. A comparison of domain integral evaluation techniques for boundary element methods. *Int. J. Numer. Meth. Engrg*, 52:417–432, 2001.

[35] M. Katsurada and H. Okamoto. The collocation points of the fundamental solution method for the potential problem. *Comp. Math. Appl.*, 31(1):123–137, 1996.

[36] D. Kay, D. Loghin, and A. Wathen. A preconditioner for the steady-state Navier–Stokes equations. *SIAM J. Sci. Comput.*, 24:237–256, 2002.

[37] R. Kress. *Linear Integral Equations*. Springer-Verlag, 1989.

[38] U. Langer, D. Pusch, and S. Reitzinger. Efficient preconditioners for boundary element matrices based on grey-box algebraic muligrid method. *Int. J. Numer. Meth. Engrg*, 58:1937–1953, 2003.

[39] C. Y. Leung and S. P. Walker. Iterative solution of large three-dimensional BEM elastostatic analyses using the GMRES technique. *Int. J. Numer. Meth. Engrg*, 40:2227–2236, 1997.

[40] D. Loghin. *Green's Functions for Preconditioning*. PhD thesis, University of Oxford, 1999.

[41] P. G. Martinsson and V. Rokhlin. A fast direct solver for boundary integral equations in two dimensions. Technical Report YALEU/DCS/RR-1264, Dept. of Math., Yale University, 2003.

[42] R. Mathon and R. L. Johnston. The approximate solution of elliptic boundary-value problems by fundamental solutions. *SIAM J. Numer. Anal*, 14(4):638–650, 1977.

[43] M. K. Ng. *Iterative Methods for Toeplitz Systems*. Oxford university Press, 2004.

[44] M. K. Ng, R. H. Chan, and W.-C. Tang. A fast algorithm for deblurring models with Neumann boundary conditions. *SIAM J. Sci. Comput*, 21(3):851–866, 1999.

[45] D. Noutsos, S. Serra Capizzano, and P. Vassalos. Matrix algebra preconditioners for multilevel Toeplitz systems do not insure optimal convergence rate. *Theoret. Comput. Sci*, 315:557–579, 2004.

[46] D. Noutsos and P. Vassalos. New band Toeplitz preconditioners for ill-conditioned symmetric positive definite Toeplitz systems. *SIAM J. Matrix Anal. Appl.*, 23(3):728–743, 2002.

[47] K. Otto. A tensor framework for preconditioners based on fast transforms. Submitted to BIT, 2004.

[48] K. Otto and E. Larsson. Iterative solution of the Helmholtz equation by a second-order method. *SIAM J. Matrix Anal. Appl.*, 21:209–229, 1999.

[49] A. Poullikkas, A. Karageorghis, and G. Georgiou. The method of fundamental solutions for inhomogeneous elliptic problems. *Comp. Mech*, 22:100–107, 1998.

[50] P. A. Ramachandran. Method of fundamental solutions: singular value decomposition analysis. *Comm. Num. Meth. Engrg*, 18:789–801, 2002.

[51] V. S. Ryaben'kii. *Method of Difference Potentials and Its Applications*. Springer Verlag, 2002.

[52] I. Saavedra and H. Power. Multipole fast algorithm for the least-squares approach of the method of fundamental solutions for three-dimensional harmonic problems. *Num. Meth. Partial Differential Equations*, 19(6):828–845, 2002.

[53] S. Serra. Preconditioning strategies for asymptotically ill-conditioned block Toeplitz systems. *BIT*, 34:579–594, 1994.

[54] S. Serra. Optimal, quasi-optimal and superlinear band-Toeplitz preconditioners for asymptotically ill-conditioned positive definite Toeplitz systems. *Math. Comp.*, 66(218):651–665, 1997.

[55] S. Serra Capizzano. Matrix algebra preconditioners for multilevel Toeplitz matrices are not superlinear. *Linear Algebra Appl.*, 343/344:303–319, 2002.

[56] S. Serra Capizzano. A note on antireflective boundary conditions and fast deblurring models. *SIAM J. Sci. Comput*, 25(4):1307–1325, 2003.

[57] S. Serra Capizzano and E. Tyrtyshnikov. Any circulant-like preconditioner for multilevel matrices is not superlinear. *SIAM J. Matrix. Anal. Appl*, 21(2):431–439, 1999.

[58] Y.-S. Smyrlis and A. Karageorghis. Some aspects of the method of fundamental solutions for certain harmonic problems. *J. Sci. Comp.*, 16(3):341–371, 2001.

[59] G. Strang. A proposal for Toeplitz matrix calculations. *Stud. Appl. Math.*, 74:171–176, 1986.

[60] E. Tyrtyshnikov. Mosaic-skeleton approximations. *Calcolo*, 33:47–57, 1996.

[61] J. S. Walker. *Fourier Analysis*. Oxford University Press, 1988.

[62] E. W. Weisstein. *CRC Concice Encyclopedia of Mathematics*. Chapman & Hall/CRC, 2nd edition, 2003.

Distribution: publications.uu.se
urn:nbn:se:uu:diva-5757

ACTA
UNIVERSITATIS
UPSALIENSIS
UPPSALA
2005