PRECONDITIONERS AND FUNDAMENTAL SOLUTIONS

PER SUNDQVIST

1. Introduction. The topic of this Licentiate thesis is the iterative solution of large, sparse systems of equations originating from discretizations of partial differential equations (PDEs). Our approach is to construct a preconditioner that is a truncated discrete convolution operator. The kernel is a fundamental solution, or an approximation of one. Our method generalizes to linear systems with block-Toeplitz structure without an underlying PDE.

The problem of solving systems of equations is very important. It arises in many applications, both with and without the PDE connection. A large amount of time is spent in industrial codes to solve such systems.

The research behind the thesis is presented in three papers, Paper A–C. Both Paper A and Paper C deal with iterative solution methods and preconditioning, and they explore two branches of the same idea. The latter paper includes analysis showing that the method we propose has favorable properties for a relevant model problem. Furthermore, numerical experiments verify that it can compete with an alternative, efficient preconditioning method for a more realistic fluid flow problem. In Paper B, we construct an algorithm for computing fundamental solutions of difference operators. The algorithm is used when constructing the preconditioner in Paper C.

This summary is organized as follows: Section 2 reviews some elementary definitions and basic relations from the theory of PDE. Also, some relevant topics in numerical linear algebra (NLA) are reviewed. In section 3 we summarize the results from the three papers, and section 4 contains an overview of related preconditioners. Section 5 ends the summary with concluding remarks concerning future work and possible extensions of the method.

2. Preliminaries. In this section we review some topics from two distinct fields of mathematics. For ease of presentation, one-dimensional problems are considered in definitions and formulae. However, we emphasize that the preconditioning technique presented in section 3.3 is applied to two-dimensional problems and easily generalized to higher dimensions.

2.1. Differential Equations and Fundamental Solutions. A pth order linear partial differential operator in one dimension, denoted by $\mathbf{P}$, can be represented in terms of its action on a function,

$$\mathbf{P}\mathbf{u}(x) = \sum_{\alpha=0}^{p} A_\alpha \frac{\partial^\alpha}{\partial x^\alpha}\mathbf{u}(x),$$

where the independent variable x is real, A_α are $n_c \times n_c$ matrices, and $\mathbf{u}$ is an n_c-vector of functions of x. Here, the matrices are independent of x, so $\mathbf{P}$ is said to have constant coefficients.

Using the Fourier series, defined component-wise on $\mathbf{u}$,

$$\hat{\mathbf{u}}_k = \frac{1}{2}\int_{-1}^{1} \mathbf{u}(x)e^{-\pi ikx}dx, \quad k \in \mathbb{Z}, \tag{2.1}$$

$$\mathbf{u}(x) \sim \sum_{k\in\mathbb{Z}} \hat{\mathbf{u}}_k e^{\pi ikx}, \qquad x \in \mathbb{R}, \tag{2.2}$$

we can formally solve the fully periodic problem

$$\begin{cases} \mathbf{Pu}(x) = \mathbf{f}(x), \\ \mathbf{u}(x) = \mathbf{u}(x+2). \end{cases} \tag{2.3}$$

Transforming (2.3) using (2.1) gives

$$\hat{\mathbf{P}}_k\hat{\mathbf{u}}_k = \hat{\mathbf{f}}_k,$$

where

$$\hat{\mathbf{P}}_k = \sum_{\alpha=0}^{p} A_\alpha(i\pi k)^\alpha.$$

The solution is found by applying (2.2) to

$$\hat{\mathbf{u}}_k = \hat{\mathbf{P}}_k^{-1}\hat{\mathbf{f}}_k.$$

Using the notation $\hat{\mathbf{E}}_k = \hat{\mathbf{P}}_k^{-1}$, we obtain the expression

$$\mathbf{u}(x) = (\mathbf{E} * \mathbf{f})(x), \tag{2.4}$$

since the convolution corresponds to multiplication of Fourier coefficients.

However, $\mathbf{E}$ cannot always be expressed as a function, and to give a meaning to (2.4) for a general $\mathbf{P}$ (with constant coefficients), we need distributions.

A distribution in $\mathcal{D}'(\Omega)$ is a linear functional on the space of infinitely differentiable functions with compact support on Ω. This space is denoted $\mathcal{D}(\Omega)$ or $C_0^\infty(\Omega)$. The application of a distribution $\mathbf{v}$ to a test function $\phi \in \mathcal{D}(\Omega)$ is sometimes denoted

$$\mathbf{v}(\phi) = \int_\Omega \mathbf{w}(x)\phi(x)dx,$$

even though $\mathbf{w}(x)$ need not be defined for all x in Ω. Normally, no distinction is made between $\mathbf{v}$ and $\mathbf{w}$, and the distribution is used without explicit reference to the test function. Distributions generalize the concept of functions, in the sense that $\mathbf{w}$ can, but does not have to be a function. In either case, it is always possible to state that $\mathbf{w}$ should be interpreted in the distribution sense.

If a distribution can be identified with an infinitely differentiable function, the distribution is said to have empty singular support. Saying that the singular support of $\mathbf{v}$ is x_0, means that $\mathbf{v}$ can be identified with an infinitely differentiable function only when applied on test functions whose support does *not* contain x_0. The standard notation is

$$\text{sing supp}(\mathbf{v}) = \{x_0\}.$$

As an example we use the Dirac distribution, commonly just denoted δ_0, but defined as

$$\delta_0(\phi) \equiv \phi(0).$$

It can be thought of as "a function that is infinite at the origin and zero elsewhere", or more correctly as a distribution with both support and singular support only at the origin and whose integral is the identity.

Another example is the Heaviside step function, which can be defined as a function according to

$$H(x) = \begin{cases} 0, & x < 0, \\ 1, & x > 0. \end{cases} \tag{2.5}$$

It is often left undefined at $x = 0$. In the distribution sense, the corresponding definition is

$$H(\phi) = \int_0^\infty \phi(x)dx. \tag{2.6}$$

Considering either (2.5) or (2.6), it is easy to see that

$$\text{sing supp}(H) = \{0\}.$$

For more information on distributions we refer to e.g. [20].

Now, the problem of finding a fundamental solution of a given differential operator $\mathbf{P}$, i.e. $\mathbf{E}$ in equation (2.4), is equivalent to solving

$$\mathbf{PE} = \delta_0 I, \tag{2.7}$$

in the distribution sense. The identity matrix I is of size $n_c \times n_c$, which implies that $\mathbf{E}$ is matrix valued and has n_c^2 components. Actually, (2.7) is the definition of a fundamental solution.

In the theory of linear differential operators, fundamental solutions are used to make statements about existence and regularity of solutions. When considering such issues, Trèves states that "The knowledge of fundamental solutions (. . .) gives valuable information in practically all questions"[29].

2.2. Numerical Linear Algebra. The principles for solving linear systems,

$$\mathcal{B}u = f, \tag{2.8}$$

have been known for a very long time. However, if the number of unknowns is large, standard direct methods such as Gaussian elimination become very costly.

To solve a really large system, it is necessary to exploit the structure often present in the coefficient matrix $\mathcal{B}$. In the following we assume that $\mathcal{B}$ is an $n_c n \times n_c n$-matrix and that u is an unknown vector with $n_c n$ elements. It can be thought of as a grid function with n_c components, defined at n grid points.

Circulant and narrow banded matrices are examples of structured matrices that allows for a fast direct solution. For other matrices, such as general sparse matrices, i.e. matrices where the number of non-zero entries is of the order $n_c n$, or Toeplitz matrices[1], there are fast and robust methods for matrix-vector multiplication. A Toeplitz matrix is a matrix whose diagonals are constant,

$$\mathcal{T} = \begin{bmatrix} B_0 & B_{-1} & \cdots & B_{-n+2} & B_{-n+1} \\ B_1 & B_0 & B_{-1} & & B_{-n+2} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ B_{n-2} & & B_1 & B_0 & B_{-1} \\ B_{n-1} & B_{n-2} & \cdots & B_1 & B_0 \end{bmatrix}, \tag{2.9}$$

where we have assumed that $n_c = 1$. If $n_c > 1$, the entries B_j in (2.9) are $n_c \times n_c$-matrices, and $\mathcal{T}$ is a block Toeplitz matrix. A circulant matrix is a Toeplitz matrix where $B_j = B_{-n+j}$ for $j = 1, \ldots, n-1$. Toeplitz matrices are completely determined by the $2n-1$ entries on the first row and the first column, and circulant matrices are completely determined by only n entries, e.g. in the first column. Note that the circulant and the narrow banded matrices form subsets of the Toeplitz and the sparse matrix sets, respectively. Hence, it seems easier to find matrices suited for fast matrix-vector multiplication than for fast solution.

Discretizations of PDE problems normally result in matrices with some sparsity structure. As an example we consider the coefficient matrix for a one dimensional boundary value problem with constant coefficients, discretized using second order centered finite differences,

$$\begin{bmatrix} C_0 & C_{-1} & 0 & \cdots & 0 \\ B_1 & B_0 & B_{-1} & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & B_1 & B_0 & B_{-1} \\ 0 & \cdots & 0 & D_1 & D_0 \end{bmatrix}. \tag{2.10}$$

This matrix can be described as a banded (block) Toeplitz matrix with a low rank disturbance. Consider also the two-level structure of the matrix originating from the corresponding discretization of a scalar two dimensional system of n_c PDE with constant coefficients, assuming uniform grid with $n = m_1 m_2$ grid-points and lexicographical ordering of the unknowns. The size of the matrix (2.11) is

[1] There are fast direct solvers that exploit Toeplitz structure too, but for many important problems, those methods fail due to algorithmical problems such as instability [3].

$n_c m_1 m_2 \times n_c m_1 m_2$ and the structure is given by

$$\begin{bmatrix} \mathcal{C}_0 & \mathcal{C}_{-1} & 0 & \cdots & 0 \\ \mathcal{B}_1 & \mathcal{B}_0 & \mathcal{B}_{-1} & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \mathcal{B}_1 & \mathcal{B}_0 & \mathcal{B}_{-1} \\ 0 & \cdots & 0 & \mathcal{D}_1 & \mathcal{D}_0 \end{bmatrix}, \tag{2.11}$$

where

$$\mathcal{B}_0 = \begin{bmatrix} C_0 & C_{-1} & 0 & \cdots & 0 \\ B_1 & B_0 & B_{-1} & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & B_1 & B_0 & B_{-1} \\ 0 & \cdots & 0 & D_1 & D_0 \end{bmatrix}, \tag{2.12}$$

and $\mathcal{B}_{-1}$ and $\mathcal{B}_1$ are block diagonal matrices of size $n_c m_1 \times n_c m_1$. The matrices $\mathcal{C}_0$, $\mathcal{C}_{-1}$, $\mathcal{D}_0$ and $\mathcal{D}_1$, as well as C_0, C_{-1}, D_0 and D_1 represent boundary conditions. In other words, (2.11) is almost block Toeplitz with block Toeplitz blocks. The rank of the disturbance is approximately $2n_c(m_1 + m_2)$.

The matrix (2.10) is well suited for a direct band solver, but (2.11) is not, since its bandwidth increases with n. A viable approach is to solve the system iteratively. Many iterative methods can be written

$$u^{i+1} = \mathcal{Q}u^i + g, \qquad i = 0, 1, 2, \dots, \tag{2.13}$$

where $\mathcal{Q}$ is the iteration matrix and g is essentially the right hand side of (2.8). The iterations are terminated when some convergence criterion is fulfilled.

The properties of $\mathcal{Q}$ determines if (2.13) converges. A sufficient condition is

$$\|\mathcal{Q}\| < 1, \tag{2.14}$$

but such a bound is often hard to prove. It is sometimes easier to analyze the eigenvalues of $\mathcal{Q}$, and for normal matrices ($\mathcal{Q}^*\mathcal{Q} = \mathcal{Q}\mathcal{Q}^*$) a condition equivalent to (2.14) is

$$\rho(\mathcal{Q}) < 1, \tag{2.15}$$

where $\rho(\cdot)$ denotes the spectral radius. However, for non-normal matrices, (2.15) is in practice only a necessary condition for convergence of the actual computations, since the norm of the solution might initially grow very fast, causing overflow.

For the fixed-point method applied to (2.8), the iteration matrix is

$$\mathcal{Q} = I - \mathcal{B}. \tag{2.16}$$

If this matrix does not possess the desired properties, it is sometimes possible[2] to find a small scalar, Δt, that multiplies both sides of (2.8), such that the new iteration matrix $I - \Delta t\mathcal{B}$ fulfills (2.15). For matrices originating from discretizations of differential operators, a larger matrix $\mathcal{B}$ requires a smaller Δt, and a smaller Δt implies that a larger number of iterations is needed.

A remedy that might allow for using a basic iterative method without experiencing the increase in number of iterations described above, is preconditioning. Instead of multiplying both sides of (2.8) by a scalar, we use a preconditioning matrix $\mathcal{K}$, such that

$$\mathcal{K}\mathcal{B}u = \mathcal{K}f. \tag{2.17}$$

To be somewhat more specific, this is called left preconditioning, and other ways of using a preconditioner are possible. Another notation often used for left preconditioning is

$$\mathcal{M}^{-1}\mathcal{B}u = \mathcal{M}^{-1}f, \tag{2.18}$$

which is convenient when the preconditioning strategy is to approximate $\mathcal{B}$ by a matrix $\mathcal{M}$, for which the preconditioner solve $\mathcal{M}v = f$ is cheap. We prefer (2.17) since we construct an approximation of $\mathcal{B}^{-1}$ and consider the preconditioner as a matrix used for multiplication. The advantage of our strategy is that $\mathcal{K}$ can be chosen from a larger set of matrices since no preconditioner solve is needed. The main disadvantage is that it is hard to account for the boundary conditions present in $\mathcal{B}$.

Good preconditioning results in that it is possible to use less complicated iterative methods, but finding a preconditioner good enough is not a trivial task, especially if the preconditioned system is supposed to fulfill condition (2.14) or (2.15). It is sometimes advisable to use a more advanced iterative method. An example is the preconditioned conjugate gradient method, see e.g. [10], which is applicable if both $\mathcal{B}$ and $\mathcal{M}$ are Hermitian and positive definite. This method exhibits especially nice convergence properties if the spectrum of $\mathcal{M}^{-1}\mathcal{B}$ is clustered, i.e. the eigenvalues are concentrated in a few groups. For a definition of clustering and its implication on the convergence rate, see e.g. [1].

A very useful tool in numerical linear algebra is the discrete Fourier transform, since some discrete operators are more easily treated in Fourier domain. An example is discrete periodic convolution, which in physical domain is represented by a full circulant matrix. In Fourier domain, it is a diagonal matrix, and hence easy both to apply and, if possible, to invert. The reason for the widespread use of the discrete Fourier transform is the availability of the Fast Fourier Transform, FFT, which is an algorithm for computing the discrete Fourier transform of a vector with n entries in $\mathcal{O}(n\log(n))$ arithmetic operations.

[2] The real parts of the eigenvalues of B must all have the same sign and no eigenvalue is allowed to be purely imaginary.

We choose to define the discrete Fourier transform and its inverse by the relations

$$\begin{aligned} \hat{u}_k &= \sum_{j=-m}^{m-1} u_j e^{-\pi ijk/m}, \qquad k=-m,\dots,m-1, \\ u_j &= \frac{1}{2m}\sum_{k=-m}^{m-1} \hat{u}_k e^{\pi ijk/m}, \quad j=-m,\dots,m-1. \end{aligned} \tag{2.19}$$

Here, the sums are applied for each of the n_c components of u. The reasons for the choice (2.19) are the implication

$$\delta_j = \begin{cases} 1, & j=0, \\ 0, & j\neq 0 \end{cases} \Longrightarrow \hat{\delta}_k = 1, \tag{2.20}$$

and the similarity with the Fourier series as defined in (2.1).

A difference operator P in one dimension can be defined by

$$Pu_j = \sum_{j'=-q_R}^{q_L} B_{j'} u_{j-j'}, \quad j \in \mathbb{Z}. \tag{2.21}$$

Here, P is said to be of order q_L+q_R+1. A matrix representation of P would at least have bandwidth $n_c(q_L+q_R+1)$. If we assume $2m$-periodicity, i.e. $u_j = u_{j+2m}$ and use (2.19), we may define the symbol $\hat{B}$ of the difference operator P by

$$\widehat{Pu}_k = \hat{B}_k \hat{u}_k = \sum_{j'=-q_R}^{q_L} B_{j'} e^{-\pi ikj'/m} \hat{u}_k. \tag{2.22}$$

We also define the fundamental solution of a difference operator. In analogy to (2.7) we say that E_j is a fundamental solution of the operator P on the set of indices $j=-m,\dots,m-1$ if

$$PE_j = \delta_j I, \quad j=-m,\dots,m-1, \tag{2.23}$$

where δ_j is given by (2.20) and I is an $n_c \times n_c$ identity matrix.. Note that, if it exists, a $2m$-periodic fundamental solution is easy to compute, using (2.19) and (2.22),

$$\hat{E}_k = \hat{B}_k^{-1}, \qquad k=-m,\dots,m-1. \tag{2.24}$$

Both the definition and the computation of fundamental solutions are easily generalized to difference operators in d dimensions.

3. Preconditioners and Fundamental Solutions. This section summarizes the main ideas and the results of our research. Our approach is to construct a preconditioner that is efficient enough to be used together with the fixed-point iterative method. We are inspired by the theory for PDE, and mainly by equation (2.4), that expresses which integral operator that is the inverse of a given differential operator. For a problem in one dimension, the preconditioner is given on matrix form by

$$\mathcal{K} = \begin{bmatrix} E_0 & E_{-1} & \cdots & E_{-n+2} & E_{-n+1} \\ E_1 & E_0 & E_{-1} & \cdots & E_{-n+2} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ E_{n-2} & \cdots & E_1 & E_0 & E_{-1} \\ E_{n-1} & E_{n-2} & \cdots & E_1 & E_0 \end{bmatrix}, \tag{3.1}$$

since

$$\mathcal{K}u_j = \sum_{j'=0}^{n-1} E_{j-j'}u_{j'} \approx n \int_\Omega \mathbf{E}(x_j - y)\mathbf{u}(y)dy, \tag{3.2}$$

cf. (2.4). In (3.2), Ω is the domain in which the PDE is solved and x_j is the position of the jth grid point in Ω. An important question is how to choose the n_c-matrices $E_{-n+1}, \ldots, E_{n-1}$.

3.1. Paper A. Paper A describes a way to use fundamental solutions of differential operators to construct preconditioners for discrete problems. Here, the preconditioner is a discretization of an integral operator. A major difficulty is that fundamental solutions have non-empty singular support, and special care is needed when discretizing integral operators with singular kernels. We deal with that problem by replacing the fundamental solution in a neighborhood of its singular support by the solution of a linear system like (2.23), but in much smaller domain.

The analysis in Paper A 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 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 A is to introduce the idea of approximating the inverse of a difference operator by discretizing a truncated convolution operator. The most important experimental results are that grid independent convergence is achieved for a two-dimensional scalar first order equation. It is required that the artificial viscosity in the discretization is taken into account when choosing the formula for the fundamental solution.

One partial differential operator considered is

$$\mathbf{P} = \frac{\partial}{\partial x_1} + \frac{\partial}{\partial x_2},$$

and it is discretized in the interior of the unit square using second order centered differences and second order artificial viscosity,

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

The parameter ϵ_h is chosen proportional to $h = 1/\sqrt{n}$, 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.

Although P is a consistent approximation of $\mathbf{P}$, it also approximates

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

for every finite ϵ_h. A fundamental solution of $\mathbf{P}'$ is

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

where K_0 is the modified Bessel function of the second kind. This function is singular at the origin. Therefore, $E_{j_1,j_2} = \mathbf{E}(x_{j_1}, x_{j_2})$ is employed at all grid points, except for in a small neighborhood of the origin, where a linear system is solved. The physical size of the neighborhood is allowed to decrease as the number of grid points is increased, keeping the number of unknowns in the system constant. Table 3.1 shows average numbers of preconditioned fixed-point iterations required to reduce the maximum norm of the initial random error by a factor 10^7.

TABLE 3.1
Average numbers of iterations for the scalar convection problem. The preconditioner is based on a fundamental solution of the convection-diffusion operator.

h^{-1}	$\epsilon_h = h$	$\epsilon_h = h/2$	$\epsilon_h = h/3$	$\epsilon_h = h/4$
16	21	5	19	65
32	22	5	19	74
64	23	6	19	74
128	24	6	19	77
256	24	6	19	77
512	24	6	19	78
1024	24	6	19	76

3.2. Paper B. If $\hat{B}_k$ is non-singular for all k, it is easy to determine a fundamental solution of a difference operator in a box. The inverse discrete Fourier transform (2.19) is applied to $\hat{E}_k$ in (2.24). Paper B deals with the problem when $\hat{B}_k$ is singular for some k. We formulate an algorithm that makes it possible to compute a fundamental solution using fast Fourier transform also when $\hat{B}$ has singularities.

We show that existence of a fundamental solution that is periodic in all dimensions is equivalent to that the symbol is non-zero. However, de Boor, Höllig, and Riemenschneider have shown that a well behaved fundamental solution exists for all difference operators with constant coefficients [7], 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, 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. At this stage, the boundary conditions have to be provided by the user. The resulting linear systems are solved using a direct method. The bandwidth of the systems corresponds to the extent of the difference operator in the dimension that is not transformed. If the extent does not depend on the grid-size, it is possible to solve each system in a number of arithmetic operations proportional to the number of unknowns in that dimension. Finally, the fundamental solution is obtained by applying a $(d-1)$-dimensional FFT to the solution of the partly transformed problem.

3.3. Paper C. An obvious modification to the preconditioning method in Paper A is to use a fundamental solution of the difference operator instead. This approach is used in Paper C. The algorithm developed in Paper B is used to compute the fundamental solutions. Table 3.2 shows iteration counts for the same problem setting as studied in Table 3.1, but now we use the new preconditioning approach. Analysis for the case $\epsilon_h = h/2$ is presented in both Paper B and Paper C.

TABLE 3.2
Average numbers of iterations for the scalar convection problem. The preconditioner is based on a fundamental solution calculated by the algorithm in Paper B.

h^{-1}	$\epsilon_h = h$	$\epsilon_h = h/2$	$\epsilon_h = h/3$	$\epsilon_h = h/4$
16	19	3	13	26
32	19	2	13	26
64	19	2	13	26
128	19	2	13	26
256	19	2	13	26
512	19	2	13	26
1024	19	2	13	26

When comparing tables 3.1 and 3.2, an obvious conclusion is that the preconditioners behave similarly. The difference is that the iteration numbers are smaller when using the discrete fundamental solution. Another conclusion is that the analysis in Paper A of the differential operators is valid also for the discretization we use.

One advantage of the approach in Paper C, compared to that in Paper A, is that no explicit a-priori knowledge about the fundamental solution is required. In Paper A we use both an explicit formula for **E** and information on its singular support. In Paper C, 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 C, the Euler equations in two dimensions are solved. These PDE have variable coefficients, non-linearities and are solved on a non-uniform grid. The differential operator is given by

$$\mathbf{P}(\mathbf{u}) = A_1(\mathbf{u})\frac{\partial}{\partial x_1} + A_2(\mathbf{u})\frac{\partial}{\partial x_2}, \tag{3.3}$$

where $\mathbf{u} = (\rho, u_1, u_2)^T$ and

$$A_1(\mathbf{u}) = \begin{pmatrix} u_1 & \rho & 0 \\ c^2/\rho & u_1 & 0 \\ 0 & 0 & u_1 \end{pmatrix}, \qquad A_2(\mathbf{u}) = \begin{pmatrix} u_2 & 0 & \rho \\ 0 & u_2 & 0 \\ c^2/\rho & 0 & u_2 \end{pmatrix}.$$

Here, u_1 and u_2 are velocity components and ρ is the density. The local speed of sound is denoted by c. More information on the problem setting, such as boundary conditions and discretization schemes is found in Paper C.

The discretized equations can be written

$$\mathcal{B}(u)u = g,$$

which indicates that the coefficient matrix depends on the solution. Due to this non-linearity, the iterative method must be modified. We use

$$u^{i+1} = (I - \mathcal{K}(u^i)\mathcal{B}(u^i))u^i + \mathcal{K}(u^i)g,$$

which means that the preconditioner is updated every iteration, by recomputing the fundamental solution. To perform this computation, a constant coefficient difference operator P is first constructed by averaging $\mathcal{B}(u^i)$ over the grid. Then, the fundamental solution of P is computed using the algorithm in Paper B, and used for preconditioning.

Table 3.3 shows iteration counts for a setting where the Euler equations are solved in a straight channel using an upwind discretization. Three preconditioners are compared; the one using fundamental solutions (F), a semicirculant preconditioner (SC) and the inverse of $\mathcal{B}(u^i)$, which is denoted by B^{-1}. Semicirculant preconditioners are briefly discussed in section 4.5.

Table 3.3

Numbers of iterations for the non-linear Euler equations.

m_1	F	SC	B^{-1}
80	73	39	25
160	66	40	26
320	56	43	27
640	44	44	28

The problem is non-linear, and at least 25 iterations are required also when the inverse is used as a preconditioner. One interpretation of this observation is that there is no use in making expensive computations to approximate the inverse very well. In this spirit, we decreased the number of updates of the preconditioner by introducing an ad-hoc algorithm used to decide when to recompute the fundamental solution. Experiments show that six updates suffice to obtain the same iteration counts as in the F-column of Table 3.3.

Finally, we give an example of a fundamental solution of a difference operator. Figure 3.1 shows the components of E_{j_1,j_2} for an upwind approximation of $\mathbf{P}$ in (3.3), with $u_1 = \rho = 1$, $u_2 = 0$ and $c \approx 1.7$.

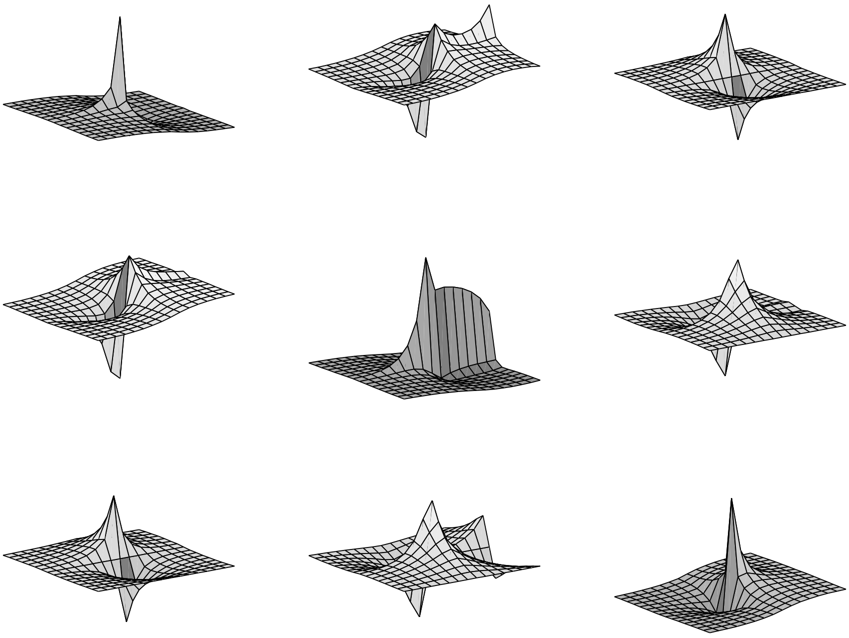

FIGURE 3.1. *The components of a fundamental solution.*

4. Related Preconditioners. A finite difference discretization of a one dimensional 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 Toeplitz structure, if it were not for the boundary conditions, cf. (2.10). 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 B_k will depend on the grid size. If there are derivatives of different order involved, this dependence can not be removed by a scaling. Yet another property that complicates the situation is that if there are derivatives of odd order in the PDE, the coefficient matrix will not be Hermitian. In spite of the 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 [28] and use the preconditioned conjugate gradient method, PCG. For a large number of references, see the expository paper by R. Chan and Ng [6]. 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. It is defined as a function f whose Fourier coefficients are the elements of $\mathcal{T}$, cf. (2.9),

$$B_j = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(\theta)e^{-ij\theta}d\theta, \quad j = -n+1, \dots n-1, \tag{4.1}$$

and it gives important information on e.g. the spectrum of $\mathcal{T}$, see [11] by Grenander and Szegő. The concept in (4.1) is easily generalized to several dimensions, corresponding to multi level block Toeplitz matrices, and some important theorems in [11] are generalized to a two-level case by Serra in [26].

The discrete fundamental solution, as defined in (2.23), is related to f in (4.1) since the symbol of P in (2.22) both generates the same Toeplitz matrix as f, and is used to compute E. In other words, $\hat{E}_k$ is equal to $1/f$ evaluated at discrete points. 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 (4.1) and using (2.22),

$$\begin{aligned} \frac{1}{2\pi}\int_{-\pi}^{\pi} \hat{B}(\theta)e^{-ij\theta}d\theta &= \frac{1}{2\pi}\int_{-\pi}^{\pi} \sum_{l=-q_R}^{q_L} B_l e^{-il\theta}e^{-ij\theta}d\theta \\ = \sum_{l=-q_R}^{q_L} \frac{1}{2\pi}B_l \int_{-\pi}^{\pi} e^{-i(l+j)\theta}d\theta &= \begin{cases} B_{-j}, & j = -q_R, \dots, q_L \\ 0, & \text{otherwise,} \end{cases} \end{aligned} \tag{4.2}$$

we find that $\hat{B}(-\theta)$ generates $\mathcal{T}$. The reason for reversing the direction is that we consider the sequence B_j of elements in $\mathcal{T}$ 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.

4.1. Circulant Preconditioners. In his proposal to use PCG for Toeplitz systems, Strang [28] also suggested using circulant matrices as preconditioners. A lot of research followed, and for a survey we again refer to [6]. The motivation for the suggestion is that all circulant matrices are diagonalized by the Fourier matrix $\mathcal{F}$, where

$$\mathcal{F}_{ij} = \frac{1}{\sqrt{m}}e^{2\pi ijk/m}. \tag{4.3}$$

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

$$\mathcal{C} = \mathcal{F}^* \Lambda \mathcal{F}, \tag{4.4}$$

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

$$\mathcal{C}^{-1} = \mathcal{F}^* \Lambda^{-1} \mathcal{F}, \tag{4.5}$$

provided Λ is invertible. It is easy to see 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, $\mathcal{F}$ and $\mathcal{F}^*$ are applied using FFT and Λ is stored as a vector, which is computed by applying FFT on the first column of $\mathcal{C}$. 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 $\mathcal{C}$.

To obtain a circulant preconditioner, one constructs a circulant approximation of $\mathcal{T}$ and uses its inverse. The approach is successful for various such approximations, but only under certain assumptions on the decay of the elements of $\mathcal{T}$ away from the main diagonal, or on the function generating $\mathcal{T}$.

4.2. Inverse Toeplitz Preconditioners. The preconditioners that, from an NLA point of view, are closest related to ours, are the inverse Toeplitz preconditioners described in [12] by Hanke and Nagy, and in [5] by R. Chan and Ng. The main similarity is that the approach used in both papers is to approximate $\mathcal{T}^{-1}$ by a Toeplitz matrix. Both also come across the problem to invert a function with zeros, cf. (2.24).

The main idea by Hanke and Nagy is to embed the matrix $\mathcal{T}$, which is assumed to be a banded Toeplitz matrix, in the smallest circulant matrix that contains $\mathcal{T}$ as the principal sub-matrix,

$$\tilde{\mathcal{T}} = \begin{bmatrix} \mathcal{T} & \mathcal{T}_{12} \\ \mathcal{T}_{21} & \mathcal{T}_{22} \end{bmatrix}. \tag{4.6}$$

The inverse of $\tilde{\mathcal{T}}$ is partitioned similarly,

$$\tilde{\mathcal{T}}^{-1} = \tilde{\mathcal{K}} = \begin{bmatrix} \mathcal{K} & \mathcal{K}_{12} \\ \mathcal{K}_{21} & \mathcal{K}_{22} \end{bmatrix}, \tag{4.7}$$

and the matrix $\mathcal{K}$ is used as a preconditioner for $\mathcal{T}$. The relation (4.5) is used to factorize $\tilde{\mathcal{T}}$. In the iterative method, it is the action of $\mathcal{K}$ that is needed, i.e. $\mathcal{K}u$, which is the first n components of

$$\tilde{\mathcal{K}} \begin{bmatrix} u \\ 0 \end{bmatrix} = \tilde{\mathcal{F}}^* \tilde{\Lambda}^{-1} \tilde{\mathcal{F}} \begin{bmatrix} u \\ 0 \end{bmatrix}. \tag{4.8}$$

If zeros occur on the diagonal of $\tilde{\Lambda}$, the inverse cannot be computed. The fix then is to replace $\tilde{\Lambda}^{-1}$ in (4.8) by another diagonal matrix $\tilde{\Lambda}^{-}$, given by

$$\tilde{\Lambda}^{-}_{jj} = \begin{cases} 1/\tilde{\Lambda}_{jj}, & \tilde{\Lambda}_{jj} > 0, \\ 0, & \tilde{\Lambda}_{jj} \leq 0. \end{cases} \tag{4.9}$$

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 $\mathcal{K}$ 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 $\tilde{\mathcal{T}}$. 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 [5], R. Chan and Ng propose using the $n \times n$ Toeplitz matrix generated by $1/f$ as a preconditioner for $\mathcal{T}$. They motivate the choice by a lemma that roughly says that if $\mathcal{T}$ is a (lower or upper) triangular non-singular Toeplitz matrix generated by f, then $\mathcal{T}^{-1}$ is generated by $1/f$.

The evaluation of (4.1) 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,

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} \frac{e^{-ij\theta}}{f(\theta)} d\theta \approx \frac{1}{sn}\sum_{k=0}^{sn-1} \frac{e^{-2\pi ijk/sn}}{(\mathfrak{K} * f)(2\pi k/sn)}, \quad j = -n+1, \ldots, n-1. \tag{4.10}$$

Kernels functions $\mathfrak{K}$ such as the Dirac kernel, the Dirichlet kernel or the Fejér kernel are used, see e.g. [30]. Its 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, provided $\mathfrak{K} * f$ is not zero for any $k = -n+1, \ldots, n-1$. In case a zero occurs, Chan and Ng replaces its inverse by zero, much like Hanke and Nagy. This is different from our approach, since we instead form the preconditioner so that zeros do not occur.

4.3. 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 $\mathcal{T}$ is full, or with an increasing number of non-zero diagonals, since if it were narrow banded, the direct solver would serve better by being used on $\mathcal{T}$ directly.

In [4], 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 $\mathcal{M}$ 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 [26]. In [27], 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 [9] by Favati, Lotti, and Menchi. It is applicable since the authors provide the Cholesky factors of the preconditioner.

The band Toeplitz preconditioners suffer from two major drawbacks. One is that the generating function, or at least its zeros, has to be known. The other is the generalization to Toeplitz matrices with block structure, such as the system (2.11) described in section 3. 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 [26] is using multigrid, but there are systems for which that method requires highly specialized implementations in order to obtain a solution within a reasonable amount of time. Another possibility is the use of fast transforms for the preconditioner solve, see e.g. [13], but that imposes limitations on the construction of the preconditioner.

The development of band Toeplitz preconditioners is ongoing, and recently, Noutsos and Vassalos [24] proposed using 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 [8] by Di Benedetto, Fiorentino, and Serra. The preconditioned matrix is $\mathcal{M}^{-1}\mathcal{K}\mathcal{T}$, where $\mathcal{T}$ is generated by some f and $\mathcal{M}$ by a low degree trigonometric polynomial $\tilde{f}$. $\mathcal{K}$ 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.

The matrices generated by $f(\theta) = \theta^4$ are troublesome also for band Toeplitz preconditioners. All of the ones mentioned above show increasing numbers of iterations, though the increase is mild for some of the approximations proposed by Serra in [27].

4.4. 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 [23], where further references are found.

Finite elements on unstructured meshes are used to discretize differential operators, resulting in unstructured matrices. The unstructured meshes makes 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. [21]. 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 realistic problems such as Navier–Stokes equations in two dimensions.

4.5. 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, 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 multi-level 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, [17] [18] [19], and the method shows favorable convergence properties for both the linearized Navier–Stokes equations for small Reynolds numbers [16] and the non-linear Euler equations [2]. In paper C, 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 other, as long as there are fast transforms available. In [13] and [15], Hemmingsson uses fast sine transforms to construct preconditioners for block Toeplitz matrices. Another example is [25] by Otto and Larsson, where the Helmholtz equation is solved.

5. Concluding Remarks. In this thesis, we summarize three reports considering preconditioners and fundamental solutions. The idea to construct preconditioners based on fundamental solutions in the way we do is new. So is the algorithm for computing fundamental solutions of difference operators that have singular symbols. There are of course many open questions and further research is necessary.

In the near future we plan to apply the method using discrete fundamental solutions to other realistic problems, e.g. flow problems with shocks. Another application field that might be considered is electromagnetism.

There are several possibilities to enlarge the set of problems for which our method is applicable. To deal with second order problems, one approach is to rewrite the PDE as a system of first order equations. Another idea is to construct a compensation for that the boundary conditions in the PDE are not captured by the preconditioner. This could be considered as a discrete version of the boundary element method, and our intention is to use it in a preconditioning context.

Combining our preconditioning technique with a domain decomposition framework for multi-block grids along the lines of [14] and [22], is one possible path to explore, in order to deal with more complicated geometries.

Further analysis of other relevant model problems is needed e.g. to understand for which problems the extremely favorable convergence results in Table 3.2 are possible. We will also try to provide automatic generation of boundary conditions for the partly transformed systems in Paper B such that the resulting fundamental solution is well behaved.

REFERENCES

[1] O. Axelsson and V. A. Barker, *Finite Element Solution of Boundary Value Problems, Theory and Computation*, Academic Press, 1984.

[2] H. Brandén and S.Holmgren, *Convergence acceleration for the steady state Euler equations.* Accepted for publication in Comput. Fluid., 2002.

[3] J. R. Bunch, *Stability of methods for solving Toeplitz systems of equations*, SIAM J. Sci. Statist. Comput., 6 (1985), pp. 349–364.

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

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

[6] R. H. Chan and M. K. Ng, *Conjugate gradient methods for Toeplitz systems*, SIAM Rev., 38 (1996), pp. 427–482.

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

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

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

[10] G. H. Golub and C. F. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, 3rd ed., 1996.

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

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

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

[14] ———, *A domain decomposition method for first-order PDEs*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 1241–1267.

[15] ———, *Toeplitz preconditioners with block structure for first-order PDEs*, Numer. Linear Algebra Appl., 3 (1996), pp. 21–44.

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

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

[18] ———, *A framework for polynomial preconditioners based on fast transforms I: Theory*, BIT, 38 (1998), pp. 544–559.

[19] ———, *A framework for polynomial preconditioners based on fast transforms II: PDE applications*, BIT, 38 (1998), pp. 721–736.

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

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

[22] E. Larsson, *A domain decomposition method for the Helmholtz equation in a multilayer domain*, SIAM J. Sci. Comput., 20 (1999), pp. 1713–1731.

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

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

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

[26] S. Serra, *Preconditioning strategies for asymptotically ill-conditioned block Toeplitz systems*, BIT, 34 (1994), pp. 579–594.

[27] ———, *Optimal, quasi-optimal and superlinear band-Toeplitz preconditioners for asymptotically ill-conditioned positive definite Toeplitz systems*, Math. Comp., 66 (1997), pp. 651–665.

[28] G. Strang, *A proposal for Toeplitz matrix calculations*, Stud. Appl. Math., 74 (1986), pp. 171–176.

[29] F. Trèves, *Linear Partial Differential Equations with Constant Coefficients*, Gordon and Breach, 1966.

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