DISCRETE FUNDAMENTAL SOLUTION PRECONDITIONING FOR HYPERBOLIC SYSTEMS OF PDE

HENRIK BRANDÉN[*], SVERKER HOLMGREN[*] AND PER SUNDQVIST[*]

Abstract. We present a new preconditioner for the iterative solution of linear systems of equations arising from discretizations of systems of first order partial differential equations (PDEs) on structured grids. Such systems occur in many important applications, including compressible fluid flow and electormagnetic wave propagation. The preconditioner is a truncated convolution operator, with a kernel that is a fundamental solution of a difference operator closely related to the original discretization.

Analysis of a relevant scalar model problem in two spatial dimensions shows that grid independent convergence is obtained using a simple one-stage iterative method. As an example of a more involved problem, we consider the steady state solution of the non-linear Euler equations in a two dimensional, non-axisymmetric duct. We present results from numerical experiments, verifying that the preconditioning technique again achieves grid independent convergence, both for an upwind discretization and for a centered second order discretization with fourth order artificial viscosity.

Key words. Preconditioner, fundamental solution, Euler equations

AMS subject classifications. 65F10, 65N22

1. Introduction.

We are interested in solving systems of equations arising from systems of non-linear first-order d-dimensional PDE of the form,

$$\mathbf{P}(\mathbf{v}) \equiv \sum_{\nu=1}^{d} A_\nu(\mathbf{v}) \frac{\partial \mathbf{v}}{\partial x_\nu} = \mathbf{g}. \tag{1.1}$$

Here, $\mathbf{v}$ is an n_c-vector of unknown functions, $\mathbf{g}$ is an n_c-vector of forcing functions, and $A_\nu(\mathbf{v})$ are $n_c \times n_c$-matrices describing the coupling of the derivatives of components of $\mathbf{v}$ in direction ν. The aim of this paper is to introduce, analyze and perform numerical experiments for a new convergence acceleration, or preconditioning, technique based on convolution with a fundamental solution of a modified difference operator.

We assume that the domain and boundary conditions for (1.1) are given such that the problem is well-posed. We also assume that the matrices $A_\nu(\mathbf{v})$ have real eigenvalues and a complete set of eigenvectors, implying that (1.1) is the steady state version of a time-dependent, strongly hyperbolic system of PDE,

$$\frac{\partial \mathbf{v}}{\partial t} + \mathbf{P}(\mathbf{v}) = \mathbf{g}. \tag{1.2}$$

Problems of this type are encountered in many fields of industry and science. One important application area is computational fluid dynamics (CFD), where for example the Euler equations for compressible, isentropic flow, studied in Section 6,

[*] Department of Scientific Computing, Information Technology, Uppsala University, Box 337, S-751 05 Uppsala, Sweden (henrik@tdb.uu.se, sverker@tdb.uu.se and pers@tdb.uu.se).

are solved. The fundamental solution preconditioning described in this paper may also be useful for other problems of wave-propagation or convection type, where the steady state solution exists and is of interest. Such problems are found e.g. in settings where flow- and electromagnetic properties interact.

The construction and application of the fundamental solution preconditioner requires that the fundamental solution is defined on a structured grid. For a structured grid in d dimensions, the grid points are identified by a set of d indices. Such a grid is not restricted to have uniform space steps, and by introducing grid mappings, curvilinear grids are included in the framework. For reasons of simplicity, memory requirement, serial efficiency, and parallelization properties, discretization on structured grids are used in many industrial codes. Complex geometries may be treated using multi-block grids, where the component grids are structured. In this paper we analyze and perform numerical experiments for discretizations of PDEs of the form (1.1) in two space dimensions on a single, structured grid. Extending the fundamental solution preconditioning technique to problems in more than two space dimensions is straight-forward [6]. No additional complications arises when constructing or applying the preconditioner. For multi-block grids, the fundamental solution preconditioners could be applied within a domain decomposition framework, similar to the preconditioning techniques presented in [13] and [21].

For some applications, a conservative formulation corresponding to (1.1) is commonly used in computations. In Section 2, we describe how the fundamental solution preconditioning technique can be applied also in such settings.

The standard iterative methods used are presented in Section 2. In Section 3, we define the notation used for discretizations of systems of PDE on structured grids and review the standard discretization techniques employed for the problems in the following sections. In Section 4 we present the new preconditioning technique, which is based on an algorithm for computing fundamental solutions of constant coefficient difference operators presented in [6]. We also briefly review the semicirculant convergence acceleration technique, which is used for comparison. In Section 5, we analyze the convergence properties of the preconditioned forward Euler iteration employed for solving a relevant model problem in 2D. The main result of the analysis is that the norm of the iteration operator is asymptotically bounded by a small number, implying that the fundamental solution preconditioning results in fast and grid independent convergence. In Section 6, the convergence properties and arithmetic work for the new preconditioning technique is studied when applied to the fully non-linear Euler equations describing 2D channel flow. Also for this more realistic problem setting, the results show that the number of arithmetic operations per grid point does not grow significantly when the number of grid points is increased.

2. Iterative methods.

Discretizing the spatial derivatives in (1.1) results in a large system of non-linear equations for the vector v^∞, representing the steady state solution,

$$\mathcal{B}(v^\infty)v^\infty = g. \tag{2.1}$$

Here, $\mathcal{B}(v)$ is a large, sparse matrix corresponding to a difference operator in space, which will be denoted by B. Using the same spatial discretization for the time-dependent problem (1.2) results in a large system of ODE,

$$\frac{dv}{dt} + \mathcal{B}(v)v = g. \tag{2.2}$$

A standard approach for computing v^∞ is to integrate (2.2) in time until the time derivative of v vanishes. In CFD codes for compressible flow problems, preconditioned explicit Runge-Kutta iterations of the following form [20] are often used,

$$\begin{array}{rcl} v^{(0)} & = & v^i \\ v^{(1)} & = & v^{(0)} - \alpha_1 \Delta t \mathcal{K}(v^{(0)}) \left(\mathcal{B}(v^{(0)})v^{(0)} - g\right) \\ & \vdots & \\ v^{(s)} & = & v^{(0)} - \alpha_s \Delta t \mathcal{K}(v^{(s-1)}) \left(\mathcal{B}(v^{(s-1)})v^{(s-1)} - g\right) \\ v^{i+1} & = & v^{(s)}. \end{array} \tag{2.3}$$

Here, the preconditioning matrix $\mathcal{K}$ is of the same size as $\mathcal{B}$ and corresponds to a preconditioning operator K in the same way as $\mathcal{B}$ corresponds to B. In practice, $\mathcal{K}$ is not explicitly formed, but an efficient algorithm for applying the preconditioner, i.e., computing $w = \mathcal{K}v$, is supplied.

In (2.3), Δt denotes the step in (pseudo-)time. If $\mathcal{K} = I$, $\alpha_s = 1$ is normally chosen, resulting in that the order of accuracy in time is at least one. For time-accurate computations, the CFL stability criterion implies that Δt has to be of the same order as the smallest space step. This leads to that many evaluations of the residual $r(v) \equiv \mathcal{B}(v)v - g$ are required for convergence to steady state, resulting in time-consuming computations. To improve the convergence properties, the preconditioner $\mathcal{K}$ could be chosen such that the CFL time step criterion is less restrictive or possibly completely removed, resulting in grid independent convergence.

The probably most popular convergence acceleration technique for the type of problems of interest here is multigrid, see, e.g., [17], [22], [23], [9] and [27]. In a multigrid scheme, grid independent convergence is mainly achieved by retaining the wave propagation property in the PDE, but increasing the wave propagation speed [22]. Using standard multigrid schemes for flow problems may result in grid-independent convergence, but the number of iterations is often rather large, see, e.g., [22] and [3]. Recent results show that, by careful tuning of the discretization, the type of multigrid cycle, the prolongation and restriction operators, and the choice of smoother, the number of iterations can be significantly reduced, see, e.g. [9], [27] and [19]. In multigrid schemes, many parameters must be properly chosen, and the methods also rely on the possibility of constructing reasonable coarse grid hierarchies. Finally, if a preconditioning technique is found successful, it might possibly be used for improving the smoother in a multigrid method. Hence, alternative techniques for convergence acceleration are worth considering.

In the iterative method (2.3), the number of levels s and the corresponding constants $\alpha_1, \ldots, \alpha_{s-1}$ are chosen such that $\|r^{i+1}\|$ is effectively reduced by the iteration. The wave propagation property of the PDE implies that the spectrum of $\mathcal{B}(v)$ approaches the origin along the imaginary axis. If a scheme retaining the wave propagation property is employed, the stability region for the Runge-Kutta method should include a segment of the imaginary axis centered at the origin. A standard scheme of this type is the three-stage method where $\alpha_1 = \alpha_2 = 0.6$, $\alpha_3 = 1$, henceforth denoted by RK3. Many practical computations have been performed using multigrid schemes where RK3 is employed as the smoother. If the preconditioner is chosen such that the wave propagation property is removed, Runge-Kutta iterations with $s < 3$ could possibly be employed. The forward Euler method corresponds to $s = 1$, and is given by

$$v^{i+1} = v^i - \Delta t \mathcal{K}(v^i) \left(\mathcal{B}(v^i)v^i - g\right).$$

We will denote this scheme by RK1. The analysis for a model problem presented in Section 5 shows that employing the fundamental solution preconditioner results in grid independent convergence when the RK1 scheme is used with pseudo-time step $\Delta t = 1$.

In Section 4, we define the fundamental solution preconditioner $K(v^i)$ as an approximation of the inverse of a difference operator related to $B(v^i)$. For a nonlinear problem of the form (1.1), this choice is motivated by noting that (2.3) is related to a fixed point iteration for the steady state problem (2.1). Assume that we linearize (2.3) by setting

$$\begin{array}{rcl} \mathcal{K}(v^{(j)}) & \equiv & \mathcal{K}(v^i), \\ \mathcal{B}(v^{(j)})v^{(j)} - g & \equiv & \mathcal{B}(v^i)v^{(j)} - g, \end{array}$$

for $j = 0, \ldots, s-1$. Then (2.3) is equivalent to solving (2.1) using a preconditioned iteration of the form,

$$\mathcal{K}(v^i)\mathcal{B}(v^i)v^{i+1} = \mathcal{K}(v^i)g, \tag{2.4}$$

where the solution of the linear systems (2.4) are approximated using one step of the linearized version of (2.3) as an inner iteration.

The preconditioner $\mathcal{K}$ may also be chosen to approximate the Jacobian matrix $\mathcal{J}$ of the residual r. Assume that we again set $\mathcal{K}(v^{(j)}) \equiv \mathcal{K}(v^i)$, but now linearize (2.3) by employing $r(v^i + \delta v) = r(v^i) + \mathcal{J}(v^i) \cdot \delta v$. Then, (2.3) is equivalent to solving (2.1) using Newton's method, i.e.,

$$v^{i+1} = v^i + \Delta v, \tag{2.5}$$

where the update Δv is computed by solving

$$\mathcal{K}(v^i)\mathcal{J}(v^i)\Delta v = -\mathcal{K}(v^i)r(v^i), \tag{2.6}$$

again using one step of the linearized version of (2.3) as the inner iteration.

For a conservative formulation of (1.1), the residual evaluations in (2.3) are computed as differences of numerical fluxes, and the difference operator B is never used. In such cases, $\mathcal{K}$ could be chosen as an approximation of the inverse of $\mathcal{J}(v^i)$.

3. Discretizations on structured grids. We assume that the spatial discretization of (1.2) is performed on a structured grid, where the grid points are identified by the index pairs (j_1, j_2), $0 \leq j_1 \leq m_1$, $0 \leq j_2 \leq m_2$. Physical and numerical boundary conditions are imposed for $j_1 = 0$, $j_1 = m_1$, $j_2 = 0$, and $j_2 = m_2$. The boundary conditions are, if not otherwise stated, used to eliminate the equations for the boundary points. This implies that we compute the approximate solution, represented by the n_c-vectors v_{j_1,j_2}, for $1 \leq j_1 \leq m_1 - 1$ and $1 \leq j_2 \leq m_2 - 1$.

When a structured grid is employed in a finite difference, finite volume or finite element discretization, the matrix $\mathcal{B}$ in (2.2) is sparse and structured. It is conveniently represented using a computational stencil, or equivalently the difference operator B, defined by an (arbitrarily ordered) list ℓ of displacement pairs (j_1', j_2') and a set of $n_c \times n_c$ coefficient matrices B_{j_1',j_2',j_1,j_2}. Applying the difference operator B to v_{j_1,j_2} then amounts to accumulating the grid function values for the stencil points, using the corresponding coefficient matrices as multipliers,

$$(Bv)_{j_1,j_2} = \sum_{(j_1',j_2') \in \ell} B_{j_1',j_2',j_1,j_2} v_{j_1-j_1',j_2-j_2'} \,, \quad 1 \leq j_1 \leq m_1 - 1 \,, \quad 1 \leq j_2 \leq m_2 - 1. \tag{3.1}$$

For the non-linear approximation in (2.2), $B_{j_1',j_2',j_1,j_2} \equiv B_{j_1',j_2',j_1,j_2}(v_{j_1,j_2})$. Within a band of width p next to the grid boundaries, the coefficient matrices in (3.1) are modified by the boundary conditions. We refer to this band as the boundary grid. For a constant coefficient difference approximation, we have that $B_{j_1',j_2',j_1,j_2} \equiv B_{j_1',j_2'}$ in the interior of the grid.

We now review two standard spatial discretization schemes, similar to schemes commonly used in applications. For ease of presentation, we apply the discretizations to a system of n_c linear PDE with constant coefficients,

$$\frac{\partial \mathbf{v}}{\partial t} + A_1 \frac{\partial \mathbf{v}}{\partial x_1} + A_2 \frac{\partial \mathbf{v}}{\partial x_2} = \mathbf{g}, \tag{3.2}$$

on a uniform grid with space step $h_1 = h_2 = h$.

The first scheme is a second-order accurate centered difference approximation, where scalar fourth-order artificial viscosity is added. For the interior of the grid, the difference approximation is given by

$$A_d D_0^{x_\nu} + \frac{\gamma_4}{2} h^3 \left(D_+^{x_\nu} D_-^{x_\nu} \right)^2 \,, \quad \nu = 1, 2. \tag{3.3}$$

Here, $D_+^{x_\nu}$, $D_-^{x_\nu}$ and $D_0^{x_\nu}$ denotes the standard forward, backward and centered difference operators in the ν-direction, respectively. Also, $\gamma_4 = \alpha c_4$, where $\alpha = \varrho(A_1) + \varrho(A_2)$, and $\varrho(\cdot)$ denotes the spectral radius [20]. Using the notation (3.1), the coefficient matrices describing the stencil in the interior of the grid are given by

$$\begin{aligned} 2hB_{0,0} &= 12\gamma_4 I_{n_c} \\ 2hB_{\pm1,0} &= \mp A_1 - 4\gamma_4 I_{n_c} \\ 2hB_{0,\pm1} &= \mp A_2 - 4\gamma_4 I_{n_c} \\ 2hB_{\pm2,0} = 2hB_{0,\pm2} &= \gamma_4 I_{n_c}. \end{aligned}$$

For this discretization, the boundary grid has width $p = 2$. The physical boundary conditions are imposed by prescribing the ingoing characteristic variables as a linear combination of the other variables, resulting in a well-posed problem [11]. For the outgoing characteristic variables extrapolation is employed. Also, numerical boundary conditions are required for the artificial viscosity terms. Here, standard modifications of the difference stencils [24] are used.

In application problems, c_4 is often chosen as a small constant where the solution is smooth. In the numerical experiments presented in Sections 5 and 6, we use the standard value $c_4 = 1/32$. If the solution contains shocks, c_4 is normally reduced to zero close to the discontinuity, and second order artificial viscosity is added instead, see, e.g., [26].

The second scheme is a standard first-order accurate upwind scheme, corresponding to that a matrix model for second-order artificial viscosity is introduced. This ensures that upwind differencing is employed for the characteristic solution variables. Assume that the matrices A_ν are diagonalized by the transforms

$$\Lambda_\nu = T_\nu^{-1} A_\nu T_\nu,$$

and define

$$|A_\nu| \equiv T_\nu |\Lambda_\nu| T_\nu^{-1}.$$

Here, $|\Lambda_\nu| = \mathrm{diag}(\mu_{\nu,1} \cdots \mu_{\nu,n_c})$. Let $\lambda_{\nu,i}$, $1 \leq i \leq n_c$, denote the eigenvalues of A_ν. In a basic upwind scheme, $\mu_{\nu,i} \equiv |\lambda_{\nu,i}|$ is used, leading to that no artificial viscosity is added for characteristic variables corresponding to $\lambda_{\nu,i} = 0$. For the non-linear Euler equations, this could results in unphysical discontinuities in the solution [14]. A standard remedy is to introduce a lower limit for the viscosity. We employ $\mu_{\nu,i} = \max(|\lambda_{\nu,i}|, c_f)$, where $c_f = 1/32$.

For the interior grid points, the upwind difference operator is given by

$$A_\nu D_0^{x_\nu} - \frac{1}{2}|A_\nu| h D_+^{x_\nu} D_-^{x_\nu} \ , \quad \nu = 1, 2. \tag{3.4}$$

The stencil is defined by

$$\begin{aligned} hB_{0,0} &= |A_1| + |A_2| \\ 2hB_{\pm1,0} &= \mp A_1 - |A_1| \\ 2hB_{0,\pm1} &= \mp A_2 - |A_2|. \end{aligned}$$

Since fourth order artificial viscosity is not included, the stencil is smaller than for the centered difference scheme. The boundary grid has width $p = 1$. Again, the ingoing characteristic variables are prescribed as linear combinations of the other variables.

When an upwind discretization of this type is used for application problems, $|A|$ is often multiplied by a limiter function Φ [26]. To achieve second order accuracy, Φ is chosen to be of size $\mathcal{O}(h)$ in smooth regions in the flow field. Furthermore, a TVD criterion is often enforced, resulting in that close to extrema of v, $\Phi = \mathcal{O}(1)$.

4. Preconditioners. In general, the matrix $\mathcal{K}$ in (2.3) could be diagonal or block-diagonal, representing point preconditioners, or narrow block-banded, representing local preconditioners where unknowns at a few neighboring grid points are coupled. Examples of such preconditioners are local time-stepping, preconditioning of the system of PDE, see, e.g., [28], and residual smoothing, see, e.g., [18] [10]. Applying the preconditioner may also involve unknowns at many grid points at large relative distances in the grid, corresponding to that $\mathcal{K}$ is essentially a full matrix. Examples of preconditioners of this type are the fundamental solution preconditioners considered in this paper, semicirculant preconditioners [4], and multigrid preconditioners. All these classes of preconditioners are based on properties of the PDE. Using fundamental solution preconditioners, the inverse of the difference operator is approximated by a convolution with a fundamental solution to a modified difference operator. In the semicirculant preconditioner, the inverse is approximated by solving a modified difference equation problem using a fast direct solver. Finally, in a multigrid scheme, a solution of the discretized problem on coarser grids is recursively computed and used as an approximate inverse in combination with smoothing operations on the fine grid.

In general, the fundamental solution preconditioner is a multi-level block matrix with Toeplitz structure within and among the blocks. Other preconditioners with this structure are the Toeplitz approximate inverse preconditioner presented in [12], and the Toeplitz preconditioner in [7], although the latter is developed only for the single-level case. Toeplitz matrices are often defined in terms of a function f that is said to generate the matrix. There is a close relation between f and the fundamental solution of a difference operator. This relation is further examined in [6].

In [5], Brandén and Sundqvist perform a preliminary study on a closely related idea. There, fundamental solutions of differential operators are used to construct preconditioners. Analysis is performed for the differential and integral operators, and it indicates grid independent convergence for first order equations when a fixed-point method is used. Numerical experiments show that this favorable behavior is indeed achievable.

4.1. Fundamental solution preconditioners. A fundamental solution of a linear, constant coefficient differential operator $\mathbf{P}$ is a distribution $\mathbf{E}$ that solves

$$\mathbf{PE} = I_{n_c}\delta_0,$$

where δ_0 is the d-dimensional Dirac distribution centered at the origin. Similarly, the fundamental solution E of a linear, constant coefficient difference operator P is defined by [8],

$$\sum_{(j_1', {j_2}')\in\ell} P_{j_1',j_2'} E_{j_1-j_1',j_2-j_2'} = \begin{cases} I_{n_c}, & j_1 = j_2 = 0, \\ 0, & j_1, j_2 \neq 0, \end{cases} \tag{4.1}$$

where E_{j_1,j_2} is an $n_c \times n_c$ matrix.

A general difference approximation operator B has variable coefficients arising from non-linearity or variable coefficients in the PDE, discretization on a non-uniform grid, or boundary conditions. In the fundamental solution preconditioner, the original boundary conditions are modified, and other types of variable coefficients are accounted for by averaging. We first form a constant coefficient approximation P by averaging B over the interior of the grid,

$$P_{j_1',j_2'} = \frac{1}{(m_1-1-2p)(m_2-1-2p)} \sum_{j_1=1+p}^{m_1-1-p} \sum_{j_2=1+p}^{m_2-1-p} B_{j_1',j_2',j_1,j_2},$$

and compute a fundamental solution E of P. Then, we define the preconditioner K by

$$\begin{aligned} u_{j_1,j_2} = (Kv)_{j_1,j_2} \equiv \sum_{j_1'=1}^{m_1-1} \sum_{j_2'=1}^{m_2-1} E_{j_1-j_1',j_2-j_2'} v_{j_1',j_2'}, \\ 1 \le j_1 \le m_1-1,\ 1 \le j_2 \le m_2-1. \end{aligned} \tag{4.2}$$

Here, K corresponds to a three-level block matrix $\mathcal{K}$. The two outer levels have Toeplitz structure, while the $n_c \times n_c$-blocks at the inner level inherit their structure from B_{j_1',j_2',j_1,j_2}. In practice, the matrix $\mathcal{K}$ is never formed. Instead, $u = \mathcal{K}v$ is efficiently performed using a standard embedding algorithm, similar to the one used for a single-level Toeplitz matrix in, e.g., [25] and [2]. Using this algorithm, v_{j_1,j_2} is first extended with zeros to the domain $-m_1 \le j_1 \le m_1-1$, $-m_2 \le j_2 \le m_2-1$. Then, the discrete Fourier transform $\hat{v}_{k_1,k_2}$ is computed using a fast Fourier transform of size $2m_1 \times 2m_2$. If an efficient FFT algorithm is not available for transforms of size $2m_1$ or $2m_2$, the domain could be extended further [2]. Now, the convolution on the extended domain is performed in Fourier space,

$$\hat{u}_{k_1,k_2} = (\widehat{Kv})_{k_1,k_2} = \widehat{E}_{k_1,k_2}\hat{v}_{k_1,k_2}\ , \qquad -m_1 \le k_1 \le m_1-1, \quad -m_2 \le k_2 \le m_2-1. \tag{4.3}$$

Here, each component of $\widehat{E}$ is the discrete Fourier transform of the corresponding component of E, defined on the extended domain. Finally, after an inverse transform of $\hat{u}$, u is restricted to the original domain $1 \le j_1 \le m_1-1$, $1 \le j_2 \le m_2-1$ again.

For the type of preconditioning described above to be useful, an efficient algorithm for determining $\widehat{E}_{k_1,k_2}$ on the extended domain must be supplied. Here, we apply the results presented in [6]. We give a brief review of the technique below.

In a practical computation of a fundamental solution using (4.1), a finite size grid must be defined. From (4.3), it is natural to require (4.1) to be fulfilled in the extended domain $-m_1 \le k_1 \le m_1-1$, $-m_2 \le k_2 \le m_2-1$. A potentially attractive approach is given by imposing periodic boundary conditions in both spatial directions at the boundaries of this domain. For a difference approximation, an analytic formula for $\widehat{P}_{k_1,k_2}$ is normally easy to derive. If $\widehat{P}_{k_1,k_2}$ is invertible at all grid points in the extended domain, we have that $\widehat{E}_{k_1,k_2} = \widehat{P}^{-1}_{k_1,k_2}$.

However, for many important application problems, this technique fails since, for these problems, (4.1) is ill-posed when periodic boundary conditions are imposed in both directions. According to Theorem 4.1 in [6], this is equivalent to that $\det(\widehat{P}_{k_1,k_2}) = 0$ at (at least) some grid point.

Hence, we have to find an alternative set of boundary conditions for (4.1), resulting in a well-posed difference equation for E. In the scheme presented in [6], the periodic boundary conditions are retained in the x_2-direction while another set of boundary conditions are imposed in the x_1-direction. Let

$$\tilde{\delta}_{j_1,k_2} = \begin{cases} 1, & j_1 = 0, \\ 0, & j_1 \neq 0. \end{cases}$$

By introducing the new boundary conditions and transforming (4.1) in the x_2-direction, we arrive at the partly transformed problem

$$\widetilde{P}_{k_2} \widetilde{E}_{j_1,k_2} \quad = \quad I_{n_c} \tilde{\delta}_{j_1,k_2} \,, \quad -m_1 \leq j_1 \leq m_1 - 1 \,, \; -m_2 \leq k_2 \leq m_2 - 1. \qquad (4.4)$$

Here, (4.4) is a set of ordinary difference equations, parameterized by k_2. Theorem 4.2 in [6] proves that, if the ordinary difference equation has a unique solution for all $-m_2 \leq k_2 \leq m_2 - 1$, then a unique fundamental solution E which satisfies the given boundary conditions exists. To analyze (4.4), it is sufficient to apply standard theory for ordinary difference equations. Furthermore, $\widetilde{E}$ is efficiently computed by solving $2m_2$ narrow-banded systems with $2m_1 n_c$ unknowns each. Finally, the entries in $\hat{E}_{k_1,k_2}$ are computed by performing $2m_2 n_c^2$ FFTs of size $2m_1$.

In Sections 5 and 6, we employ the fundamental solution preconditioner described above in combination with the forward Euler iteration. We denote this scheme by F-RK1.

4.2. Semicirculant preconditioners. For comparison, we also consider a semicirculant preconditioner for solving the Euler equations in Section 6. The semicirculant preconditioner corresponds to the discretization of a modified linear PDE problem [15] [16]. Here, the preconditioning algorithm provides a fast direct solver for the modified problem. When constructing the semicirculant preconditioner, periodicity is assumed in all directions but one. This is the same approach as for the fundamental solution preconditioners, but applied to the difference approximation of the PDE directly. In [4], numerical experiments show that semicirculant preconditioning achieves grid-independent convergence for the non-linear Euler equations, discretized with the same type of centered difference approximation as presented in Section 3. Analysis is also presented for a model problem. It is proved that, using the forward Euler method with a constant Δt, the spectrum for the iteration operator remains in finite region bounded away from the origin as $m_{1,2} \rightarrow \infty$. We denote the forward Euler iteration, combined with semicirculant preconditioning, by SC-RK1.

4.3. Arithmetic complexities. For the fundamental solution and semicirculant preconditioners, the arithmetic work is divided between two phases; computing the representation of the preconditioning operator, and applying the preconditioner to a grid function. When solving a linear PDE, the first phase only has to be performed once. For a non-linear iteration of the type (2.3), the preconditioner should in principle be recomputed at each iteration stage. However, as is shown in Section 6, in a practical computation the preconditioners could be reused for several time steps.

In Table 4.1, the number of arithmetic operations (floating point operations for variables of real type) per grid point for computing and applying the preconditioners for a system of PDE are shown.

TABLE 4.1
Arithmetic complexities for the fundamental solution and semicirculant preconditioners

	Compute
F	$n_c \Big(16\left((p+1)n_c-1\right)^2+24(p+1)+20n_c\left(\log_2 m_1+1\right)\Big)$
SC	$4n_c\left((p+1)n_c-1\right)^2$

	Apply
F	$n_c\left(20\log_2(m_1+m_2)+8n_c+40\right)$
SC	$n_c\left(5\log_2 m_1+12(p+1)n_c-4\right)$

For storing the representations of the fundamental solution and semicirculant preconditioners, $4n_c^2$ and $3pn_c^2$ memory positions per grid point are required, respectively.

5. A model problem. In this section, we apply the fundamental solution preconditioning technique to a relevant scalar model problem. For the upwind discretization described in Section 3, we analyze the convergence properties of the forward Euler iteration. We prove that, using $\Delta t = 1$, the norm of the iteration operator is bounded by a constant significantly smaller than one for all grid sizes, implying fast and grid independent convergence. We also present numerical results showing the same type of favorable convergence properties when the centered difference approximation in Section 3 is employed.

We consider the scalar Kreiss equation,

$$\frac{\partial v}{\partial x_1}+\frac{\partial v}{\partial x_2}=0. \tag{5.1}$$

The PDE (5.1) is solved on the unit square, where we prescribe $v(x_1,0)$ and $v(0,x_2)$, yielding a well-posed problem. The discretizations are applied on a uniform grid where $m_1=m_2=m$, implying that $h_1=h_2=h=1/m$.

The upwind difference operator is given by

$$(Bv)_{j_1,j_2} = \begin{cases} (D_-^{x_1} + D_-^{x_2})\, v_{j_1,j_2}, & \begin{cases} j_1 = 2,\dots,m-1 \\ j_2 = 2,\dots,m-1 \end{cases} \\ (D_-^{x_1} + h^{-1})\, v_{j_1,j_2}, & \begin{cases} j_1 = 2,\dots,m-1 \\ j_2 = 1 \end{cases} \\ (h^{-1} + D_-^{x_2})\, v_{j_1,j_2}, & \begin{cases} j_1 = 1 \\ j_2 = 2,\dots,m-1 \end{cases} \\ (h^{-1} + h^{-1})\, v_{j_1,j_2}, & \begin{cases} j_1 = 1 \\ j_2 = 1 \end{cases} \end{cases} . \qquad (5.2)$$

For this problem, the difference operator B has constant coefficients, and we simply have that

$$(Pv)_{j_1,j_2} = (D_-^{x_1} + D_-^{x_2}) v_{j_1,j_2}. \qquad (5.3)$$

The fundamental solution preconditioner is constructed as described in Section 4.1, using a Dirichlet boundary condition in the x_1-direction. In Section 5.1 in [6] it is shown that the fundamental solution E, satisfying

$$\begin{array}{lcll} (PE)_{j_1,j_2} & = & \delta_{j_1,j_2}, & j_1, j_2 = -m,\dots,m, \\ E_{j_1,-m} & = & E_{j_1,m}, & j_1 = -m,\dots,m, \\ E_{-m,j_2} & = & 0, & j_2 = -m,\dots,m, \end{array} \qquad (5.4)$$

exists and is unique. Below, we study the problem one step further and derive an explicit formula for E. First, define

$$E^D_{j_1,j_2} = \begin{cases} \dfrac{h}{2^{j_1+j_2+1}} \begin{pmatrix} j_1 + j_2 \\ j_2 \end{pmatrix}, & \begin{cases} j_1 = 0,1,\dots \\ j_2 = 0,1,\dots \end{cases} \\ 0, & \text{elsewhere} \end{cases} \qquad (5.5)$$

We now prove the following lemma,

Lemma 5.1. *E^D satisfies*

$$E^D_{j_1,j_2+1} \le \frac{1}{1+h} E^D_{j_1,j_2}, \quad \begin{cases} j_1 = 0,\dots,m-1 \\ j_2 = m, m+1,\dots \end{cases}$$

Proof. Using basic properties of binomial coefficients we have that

$$E^D_{j_1,j_2+1} = \frac{h}{2^{j_1+j_2+2}} \frac{j_1+j_2+1}{j_2+1} \begin{pmatrix} j_1 + j_2 \\ j_2 \end{pmatrix} = \frac{1}{2} \frac{j_1+j_2+1}{j_2+1} E^D_{j_1,j_2}.$$

If $(j_1 + j_2 + 1)/2\,(j_2 + 1) \le (1 + h)^{-1}$ or equivalently $j_1 \le (j_2 + 1)(1 - h)/(1 + h)$, the lemma is true. For the largest value of j_1 and the smallest value of j_2 this states that $m - 1 \le (m + 1)\,(1 - h)/(1 + h) = m - 1$ which obviously is true. □

The fundamental solution is given by the following theorem,

Theorem 5.2. *Let P be given by (5.3), and*

$$E_{j_1,j_2} = \sum_{n=0}^{\infty} E^D_{j_1,j_2+2nm}. \tag{5.6}$$

Then, E is well defined and satisfies (5.4).

Proof. First, we show that the sum (5.6) is convergent. According to Lemma 5.1 there is a constant $r < 1$ such that

$$\sum_{n=1}^{\infty} E^D_{j_1,j_2+2nm} \le E^D_{j_1,j_2+2m} \sum_{n=1}^{\infty} r^{2(n-1)m}, \quad \begin{cases} j_1 = 0,\dots,m \\ j_2 = -m,\dots,m \end{cases},$$

and the result follows from convergence of the geometric series. Now, the formulas (5.4) are verified by straight-forward calculation. □

We now study the convergence of the F-RK1 iteration. Define the error by $e^n = v^n - v$. We have that

$$e^{n+1} = e^n - \Delta t K B e^n, \quad n = 0, 1, \dots,$$

Hence, the error satisfies the following bound,

$$||e^n|| \le ||(I - \Delta t K B)^n||\,||e^0|| \le ||I - \Delta t K B||^n ||e^0||. \tag{5.7}$$

Consequently, $||I - \Delta t K B|| < 1$ is a sufficient condition for convergence. According to the following theorem, this condition is actually fulfilled for all grid sizes for $\Delta t = 1$.

Theorem 5.3. *Let B and K be given by (5.2) and (5.4) respectively. Then,*

$$||I - KB||_\infty < \frac{1}{2}\frac{1}{1-(1+h)^{-2m}}.$$

Proof. From summation by parts and the fact that E is a fundamental solution of P follows that

$$(v - KBv)_{j_1,j_2} = -h^{-1} \sum_{j_1'=1}^{m-1} E_{j_1-j_1',j_2-m}\, v_{j_1',m-1},$$

where we have used that $E_{j_1,j_2} = 0$ for $j_1 < 0$. Using $E \ge 0$ and the monotone behavior of E as stated by Lemma 5.1, we have that

$$\begin{aligned}
||I - KB||_\infty &= \max_{j_1, j_2 = 1, \dots, m-1} h^{-1} \sum_{j_1'=1}^{m-1} E_{j_1 - j_1', j_2 - m} \\
&= h^{-1} \sum_{j_1=0}^{m-2} E_{j_1, 1-m} \\
&= h^{-1} \sum_{n=1}^{\infty} \sum_{j_1=0}^{m-2} E^D_{j_1, (2n-1)m+1} \\
&\le \sum_{n=1}^{\infty} \frac{1}{(1+h)^{(2n-2)m}} h^{-1} \sum_{j_1=0}^{m-2} E^D_{j_1, m+1} \\
&< \frac{1}{1-(1+h)^{-2m}} 2^{-m-2} \sum_{j_1=0}^{m+1} 2^{-j_1} \begin{pmatrix} m+1+j_1 \\ m+1 \end{pmatrix} \\
&= \frac{1}{1-(1+h)^{-2m}} \frac{1}{2}.
\end{aligned}$$

The last step is proved by induction, using Pascal's identity. □

In Table 5.1 the bound of the norm, as provided by Theorem 5.3, is compared to numerically computed values of $||I - KB||_\infty$. By comparing the rows in the table, it is clear that the bound is not very sharp.

Table 5.1
The bound provided by Theorem 5.3 and numerically computed values of the norm

$m_{1,2}$	8	16	32	64	∞
$\lVert I - KB\rVert_\infty$	0.228	0.298	0.354	0.396	
Bound in Theorem 5.3	0.590	0.584	0.581	0.580	0.580

Finally, we present numerical experiments for the model problem (5.1) using both the upwind discretization, covered by the analysis above, and the centered discretization. We use a random initial guess and iterate until the maximum norm of the error is decreased by a factor 10^5. Our experiments show that the optimal value of Δt is 1 for the upwind case, and 0.6 when using centered differences. Iteration counts are presented in Table 5.2. From the table, it is clear that the number of iterations is independent of the number of grid points for both discretizations. However, the number of iterations for the upwind discretization is

Table 5.2
Grid refinement study for the model problem. The table lists the number of iterations required for convergence.

$m_{1,2}$	8	16	32	64	128	256	512
upwind, $\Delta t = 1$	3	2	2	2	2	2	2
centered, $\Delta t = 0.6$	35	39	41	40	41	40	41

much smaller than indicated by the results in Table 5.1. A simple computation using Theorem 5.3 yields that at most 22 iterations are required for $m_{1,2} > 8$, which is clearly not a sharp estimate. Furthermore, a similar bound computed using the numerically computed norms of the iteration operator presented in Table 5.1 does not explain the fast convergence either. However, for the upwind discretization applied to the model problem, it is in fact also possible to prove that $\lim_{m_{1,2}\to\infty} ||(I-KB)^2||_\infty = 0$. This property, which is not shared by the centered difference discretization, explains the very favorable convergence properties for the upwind discretization shown in Table 5.2.

6. The Euler equations for channel flow. We now consider the non-conservative formulation of the steady-state Euler equations

$$A_1(\mathbf{v})\frac{\partial \mathbf{v}}{\partial x_1} + A_2(\mathbf{v})\frac{\partial \mathbf{v}}{\partial x_2} = 0, \tag{6.1}$$

where $\mathbf{v} = (\rho, v_1, v_2)^T$, ρ is the density and v_1 and v_2 are the velocity components. We assume isentropic flow, implying that the pressure p is given by $p = \beta\rho^\gamma$. Here, we employ $\beta = 2$ and $\gamma = 1.4$. The speed of sound is $c = \sqrt{\gamma\beta\rho^{\gamma-1}}$.

The coefficient matrices $A_{1,2}$ are given by

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

and

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

We study sub-sonic flow in a non-axisymmetric duct, see Figure 6.1. We consider four problem settings, where we solve the Euler equations in a straight and a narrowing channel setting, using the centered and upwind discretizations described in Section 3. For the narrowing channel, the width of the outflow boundary is 3/4 of width of the inflow boundary. The length of the channel is ten times the width at the inflow, and the Mach number at the inflow for the steady-state solution is approximately 0.5. The boundary conditions are given by prescribing the in-going characteristic variables at the in- and outflow boundaries, and setting the flow perpendicular to the walls equal to zero, i.e.,

$$\begin{array}{ll} \rho + v_1 = g_1 \text{ and } v_2 = g_2 & \text{at inflow boundary,} \\ \rho - v_1 = g_3 & \text{at outflow boundary,} \\ \hat{n}\cdot(v_1, v_2) = 0 & \text{at walls.} \end{array} \tag{6.2}$$

The problem is discretized utilizing an orthogonal grid. We introduce an orthogonal mapping T

$$T : \left\{ \begin{array}{lcl} x_1 & = & x_1(\xi_1, \xi_2) \\ x_2 & = & x_2(\xi_1, \xi_2) \end{array} \right.$$

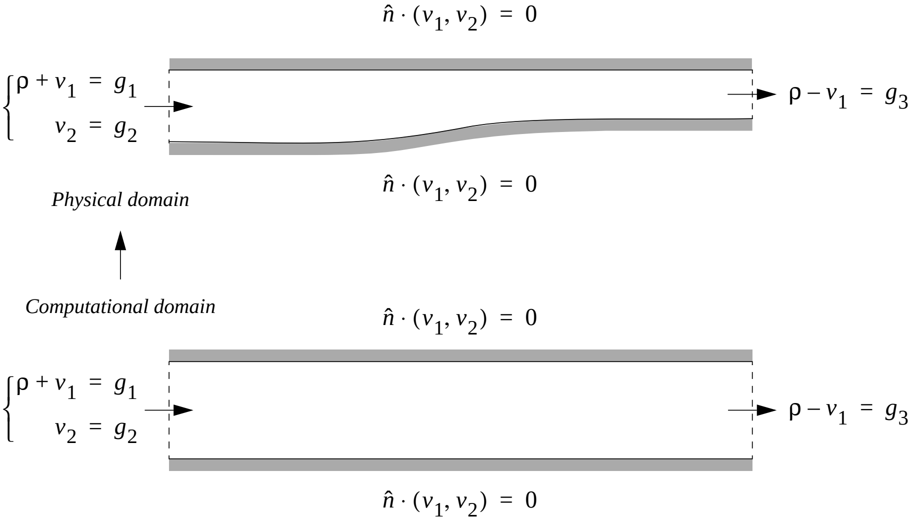

FIG. 6.1. *The physical and the computational domain.*

between the physical domain and a rectangular computational domain as shown in Figure 6.1. Using an orthogonal grid simplifies the implementation. However, the fundamental solution preconditioning technique is not limited to this situation. The discretization procedure is as follows:

1. A software package [1] is used to define the orthogonal mapping T.
2. The equations are transformed from the physical to the computational domain. The transformed equations are given in [3]. After solving the transformed problem, the solution can easily be transformed back to the physical domain.
3. The transformed equations are discretized on an $m_1 \times m_2$ uniform grid in the computational domain.
4. Physical and numerical boundary conditions are introduced according to the discussion in Section 3.

Once the problems are set up, they are solved using the F-RK1 and SC-RK1 schemes. For each problem setting, we perform a grid refinement study. For the two solution schemes we use a constant Δt, determined by numerical experiments, for each problem setting. In Tables 6.1 and 6.2, the values of Δt used are shown.

TABLE 6.1
The time-step Δt for the F-RK1 and SC-RK1 methods, centered difference approximation

Method	F-RK1	SC-RK1
Straight channel	1.5	0.7
Narrowing channel	0.8	1.1

TABLE 6.2
The time-step Δt for the F-RK1 and SC-RK1 methods, upwind discretization

Method	F-RK1	SC-RK1
Straight channel	0.9	0.6
Narrowing channel	0.9	0.9

Tables 6.3, 6.4, 6.5, and 6.6 show the number of iterations required for convergence for the four problem settings. Here, we use $m_2 = m_1/10$, resulting in approximately square grid cells in the physical domain. The fundamental solution and semicirculant preconditioners are recomputed at every time step. As initial guess we use a Gaussian pulse in variables ρ and v_1, and we iterate until the norm of the residual has been decreased by a factor of 10^5.

TABLE 6.3
Grid refinement study for the straight channel setting, centered difference approximation.

m_1	80	160	320	640
#F-RK1 iterations	73	53	47	48
#SC-RK1 iterations	51	59	63	66

TABLE 6.4
Grid refinement study for the narrowing channel setting, centered difference approximation.

m_1	80	160	320	640
#F-RK1 iterations	219	218	219	225
#SC-RK1 iterations	55	56	61	66

TABLE 6.5
Grid refinement study for the straight channel setting, upwind difference approximation.

m_1	80	160	320	640
#F-RK1 iterations	73	66	56	44
#SC-RK1 iterations	39	40	43	44

TABLE 6.6
Grid refinement study for the narrowing channel setting, upwind difference approximation.

m_1	80	160	320	640
#F-RK1 iterations	104	103	105	106
#SC-RK1 iterations	32	32	34	35

From the tables, we see that the number of iterations does not grow with the number of grid points for any of the schemes. In fact, for the straight channel settings, the number of iterations for the F-RK1 method decreases as the number of grid points is increased. For the narrowing channel settings, using the SC-RK1 scheme results in a significantly smaller number of iterations than the

F-RK1 scheme. However, for the straight channel settings the difference is in general smaller, and when employing the centered discretization the F-RK1 scheme requires less iterations than the SC-RK1 method for problems with many grid points. From the tables, we also see that when solving this non-linear system of PDE, the number of iterations required for the upwind and centered discretizations are similar, which was not the case for the linear, scalar model problem studied in Section 5, see Table 5.2.

As remarked in Section 4.3, it may not be necessary to change the preconditioner at each time step. We examine this by introducing an ad-hoc strategy for determining when to recompute the preconditioners. Let $\rho^i \equiv \|r(v^i)\|/\|r(v^0)\|$. If $\rho^i > 0.1$, the preconditioner is rebuilt in every iteration, and if $0.01 < \rho^i < 0.1$ it is rebuilt in every 10th iteration. Finally, if $\rho^i < 0.01$, the preconditioner is not rebuilt anymore. In Tables 6.7 and 6.8, the experiments for narrowing channel settings, shown in Tables 6.4 and 6.6 are repeated, but here the ad-hoc strategy described above is introduced.

Table 6.7

Grid refinement study for the narrowing channel setting, centered difference approximation, ad-hoc updating strategy for the preconditioner.

m_1	80	160	320	640
#F-RK1 iterations/preconditioner computations	218/6	217/8	218/20	224/28
#SC-RK1 iterations/preconditioner computations	54/7	58/10	70/12	67/13

Table 6.8

Grid refinement study for the narrowing channel setting, upwind difference approximation, ad-hoc updating strategy for the preconditioner

m_1	80	160	320	640
#F-RK1 iterations/preconditioner computations	97/7	100/8	103/11	105/12
#SC-RK1 iterations/preconditioner computations	28/5	33/7	34/7	37/8

Comparing the results to Tables 6.4 and 6.6, we find that the number of iterations is only marginally affected, while the number of preconditioner computations is significantly reduced. Note that, for the F-RK1 scheme, the number of iterations is in fact slightly decreased by introducing the updating strategy for the preconditioner.

Complementing the iteration numbers given above, we also compare the arithmetic complexities for the F-RK1 and SC-RK1 schemes. In Figure 6.2, we show the number of arithmetic operations per grid point for the straight and narrowing channel settings, discretized using the centered approximation, and employing the ad-hoc strategy for recomputing the preconditioners. Using the centered discretization, the residual evaluation is significantly less expensive than for the upwind discretization, and requires only about 120 arithmetic operations per grid point. The complexities for constructing and applying the preconditioners are computed by substituting $n_c = 3$ and $p = 2$ in the formulas in Table 4.1. From the figures, it is clear that the arithmetic complexities for the straight channel

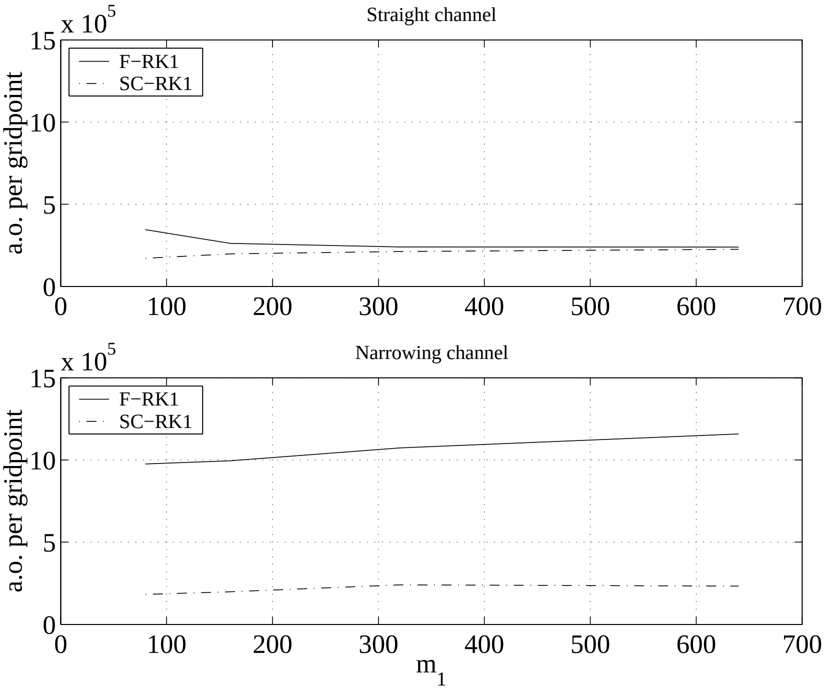

FIG. 6.2. *Arithmetic complexities per grid point for the centered difference approximation*

setting are almost the same, while the SC-RK1 scheme is more efficient for the narrowing channel setting. A probable reason for this is that variable coefficients in the x_2-direction are taken into account in the semicirculant preconditioner. In the fundamental solution preconditioner, variable coefficients are averaged in both spatial directions.

7. Conclusions. In this paper, the iterative solution of time-independent, hyperbolic systems of PDEs is considered. The PDEs are discretized by standard centered and upwind finite difference discretizations on structured grids, and the systems of equations are solved by integrating the corresponding time-dependent problem until steady state is reached. The integration is performed using explicit time-marching combined with a preconditioner.

We propose a new preconditioner that is a truncated convolution operator utilizing a discrete fundamental solution. Both the computation of the fundamental solution and the application of the preconditioning operator is performed using FFT. Considered as a multi-level block matrix, the preconditioner has Toeplitz structure on all levels except one.

Grid independent convergence is obtained, both for a scalar linear PDE with constant coefficients and for the non-linear Euler equations discretized on a non-uniform grid. For the scalar case, analysis is performed, proving that at most 22 iterations is enough to decrease the residual norm by a factor 10^5.

For comparison, a semicirculant preconditioner [4] is used. For a problem where the Euler equations are solved in a straight channel, the arithmetic work required for computing the solution is almost the same for the two preconditioners. For a narrowing channel setting, the semicirculant preconditioner is more efficient. We conclude that the fundamental solution preconditioner needs further development to become really competitive. However, the potential of the new preconditioner is indicated by the analysis, and to an even larger extent by the experiments, for the model problem in Section 5.

REFERENCES

[1] L. Abrahamsson, *Orthogonal grid generation for two-dimensional ducts*, J. Comput. Appl. Math., 34 (1991), pp. 305–314.

[2] D. H. Bailey and P. N. Swarztrauber, *The fractional Fourier transform and applications*, SIAM Rev., 33 (1991), pp. 389–404.

[3] H. Brandén and S. Holmgren, *Convergence acceleration for hyperbolic systems using semicirculant approximations*, J. Sci. Comput., 14 (1999), pp. 357–393.

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

[5] H. Brandén and P. Sundqvist, *Preconditioners based on fundamental solutions*, Tech. Rep. 2000-032, Dept. of Information Technology, Uppsala Univ., Uppsala, Sweden, 2000.

[6] H. Brandén and P. Sundqvist, *An algorithm for computing fundamental solutions of difference operators*, Tech. Rep. 2003-006, Dept. of Information Technology, Uppsala Univ., Uppsala, Sweden, 2003.

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

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

[9] B. Diskin, *Efficient multigrid methods for solving upwind-biased discretizations of the convection equation*, AMC, 123 (2001), pp. 343–379.

[10] R. Enander, *Implicit explicit residual smoothing for the multidimensional Euler and Navier–Stokes equations*, SIAM J. Sci. Comput., 18 (1997), pp. 1243–1254.

[11] B. Gustafsson, H. Kreiss, and J. Oliger, *Time dependent problems and difference methods*, John Wiley & Sons, 1995.

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

[14] C. Hirsch, *Numerical computation of internal and external flows, Volume 2: Computational methods for inviscid and viscous flows*, John Wiley & Sons, 1990.

[15] 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.

[16] ———, *Semicirculant preconditioners for first-order partial differential equations*, SIAM J. Sci. Comput., 15 (1994), pp. 385–407.

[17] A. Jameson, *Solution of the Euler equations for two-dimensional transonic flow by a multigrid method*, Appl. Math. Comput., 13 (1983), pp. 327–355.

[18] ———, *Computational transonics*, Comm. Pure Appl. Math., 41 (1988), pp. 507–549.

[19] A. Jameson and D. Caughey, *How many steps are required to solve the Euler equations of steady, compressible flow: In search for a fast solution algorithm*, in 15th Computational Fluid Dynamics Conference, 2001.

[20] A. Jameson, W. Schmidt, and E. Turkel, *Numerical solution of the Euler equations by the finite volume method using Runge-Kutta time-stepping schemes*, AIAA, 81-1259 (1981).

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

[22] P. Lötstedt, *Grid independent convergence of the multigrid method for first-order equations*, SIAM J. Numer. Anal., 29 (1992), pp. 1370–1394.

[23] ———, *Improved convergence to the steady state of the Euler equations by enhanced wave propagation*, J. Comput. Phys., 114 (1994), pp. 34–44.

[24] T. E. Pulliam, *Artificial dissipation models for the Euler equations*, AIAA, 12 (1986), pp. 1931–1940.

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

[26] R. C. Swanson and E. Turkel, *On central-difference and upwind schemes*, J. Comput. Phys., 101 (1992), pp. 292–306.

[27] J. Thomas, B. Diskin, A. Brandt, and J. South, *General framework for achieving textbook multigrid efficiency: one-dimensional Euler eample*, in Frontiers of computational fluid dynamics 2002, D. Caughey and M. Hafez, eds., World Scientific, 2002, pp. 61–80.

[28] E. Turkel, *Preconditioning techniques in computational fluid dynamics*, Annu. Rev. Fluid Mech., 31 (1999), pp. 385–416.