CONVERGENCE ACCELERATION FOR HYPERBOLIC SYSTEMS USING SEMICIRCULANT APPROXIMATIONS

HENRIK BRANDÉN* AND SVERKER HOLMGREN*

Abstract. The iterative solution of systems of equations arising from systems of hyperbolic, time-independent partial differential equations (PDEs) is studied. The PDE is discretized using a finite volume or finite difference approximation on a structured grid. A convergence acceleration technique where a semicirculant approximation of the spatial difference operator is employed as preconditioner is considered. The spectrum of the preconditioned coefficient matrix is analyzed for a model problem. It is shown that, asymptotically, the time step for the forward Euler method could be chosen as a constant, which is independent of the number of grid points and the artificial viscosity parameter. By linearizing the Euler equations around an approximate solution, a system of linear PDE with variable coefficients is formed. When utilizing the semicirculant (SC) preconditioner for this problem, which has properties very similar to the full nonlinear equations, numerical experiments show that the favorable convergence properties hold also here. We compare the results for the SC method to those of a multigrid (MG) scheme. The number of iterations and the arithmetic complexities are considered, and it is clear that the SC method is more efficient for the problems studied. Also, the MG scheme is sensitive to the amount of artificial dissipation added, while the SC method is not.

Key words. Convergence acceleration, Preconditioning, Semicirculant approximations, Euler equations

AMS subject classifications. 65N22, 65F10, 65M12

1. Introduction. We are interested in the solution of time-independent first-order systems of PDE,

$$\mathcal{B}\mathbf{v} - \mathbf{g} = 0. \tag{1.1}$$

Here, $\mathbf{v}$ is the vector of unknown functions and $\mathbf{g}$ is the forcing function vector. The differential operator $\mathcal{B}$ is given by

$$\mathcal{B} = \sum_{\nu=1}^{d} A_\nu \frac{\partial}{\partial x_\nu}, \tag{1.2}$$

where d is the number of spatial dimensions. We assume that a domain and boundary conditions are supplied, resulting in a well-posed problem. If the matrices A_ν have real eigenvalues, (1.1) could be interpreted as the steady-state limit of a time-dependent, hyperbolic system of PDE,

$$\frac{\partial \mathbf{v}}{\partial t} + \mathcal{B}\mathbf{v} - \mathbf{g} = 0. \tag{1.3}$$

Examples of application problems of this type include inviscid gas and fluid flow, plasma physics, and other problems where flow and electromagnetic properties interact.

A standard approach for solving problems of the type (1.1) is by the method of lines:

*Department of Scientific Computing, Information Technology, Uppsala University, Box 120, S-751 04 Uppsala, Sweden (henrik@tdb.uu.se and sverker@tdb.uu.se)

1. The derivatives in (1.2) are discretized, yielding a large system of equations for the vector v, representing the discrete solution,

$$Bv - g \equiv r(v) = 0. \tag{1.4}$$

Here, B is a difference approximation of (1.2), including boundary conditions. We choose the sign such that, for a stable discretization, the eigenvalues of B lie in the right half plane. The vector g contains forcing function values and boundary data, and $r(v)$ denotes the residual.
2. The semidiscrete time-dependent problem corresponding to (1.3),

$$\frac{dv}{dt} + r(v) = 0, \tag{1.5}$$

is integrated in time until steady-state is reached. Here, some explicit time-marching method is normally employed.

In this paper, we study a convergence acceleration, or *preconditioning*, technique based on semicirculant approximations of the spatial difference operator B. We will consider linear problems in two space dimensions. In general, the PDE (1.1) may be quasi-linear, i.e., the matrices A_ν depend on the solution $\mathbf{v}$. The framework is extensible to 3D and quasi-linear problems, and these generalizations are briefly discussed in §7. In this paper, we use second-order accurate central differences on structured grids for the discretization of the spatial derivatives in (1.2). For reasons of efficiency, memory requirement and simplicity, this type of discretization is often employed in application codes. The resulting systems of equations (1.4) are highly non-diagonally dominant, and standard preconditioning techniques may fail.

The treatment of multi-block structured grids, which are utilized to solve problems in complex geometries, is the subject of current research. It is not clear if the semicirculant approximation framework could be generalized to unstructured grids.

In §2, the iterative methods used for solving (1.5) are described. In §3, we give a review of the semicirculant approximation framework, and in §4, we briefly describe a standard multigrid acceleration technique which is used for comparison. The convergence for a scalar model problem is analyzed in §5, and in §6, we consider the solution of the linearized Euler equations governing channel flow.

For time-dependent and time-independent scalar model problems, preconditioners based on semicirculant approximations have previously been considered and analyzed in [9] [10] [11] [12]. Semicirculant preconditioners form a subset of the polynomial normal block (PNB) framework [13], which is employed for high-order accurate centered discretizations in [14]. In all of these papers, the semicirculant preconditioners are used together with non-stationary iterative methods like GMRES and TFQMR. The analysis of the model problem presented in §5 is related to the analysis in [11]. However, for the problem studied in [11], the artificial viscosity is of non-standard form.

In [8], the technique presented in this paper is successfully applied to the linearized Navier-Stokes equations describing viscous flow over a flat plate. Also, a scalar model problem of convection-diffusion type is analyzed, and it is shown that the time step for the forward Euler method could be chosen as a constant, independent of the number of grid points and the Reynolds number. In [6], another scalar convection-diffusion equation is considered, and analysis shows that the same type of favorable convergence properties holds also for this problem.

2. Iterative methods. For application problems, memory requirements are critical and an s-stage Runge-Kutta iteration of the following memory-lean form is often used for integrating (1.5),

$$\begin{aligned} v^{(0)} &= v^i \\ v^{(1)} &= v^{(0)} - \alpha_1 k r(v^{(0)}) \\ &\vdots \\ v^{(s)} &= v^{(0)} - \alpha_s k r(v^{(s-1)}) \\ v^{i+1} &= v^{(s)} \end{aligned} \tag{2.1}$$

Here, k denotes the time step. Note that s evaluations of the residual are required per iteration.

For (1.5), Fourier analysis could be used to show that convergence to steady state is essentially achieved by a combination of two processes: High frequency error modes are damped, and low frequency error modes are transported out of the domain through the outflow boundaries [4] [20]. Often, the transport process is the most important.

In many cases, the iteration (2.1) converges very slowly. There are two viable approaches for improving the convergence properties:

1. Improve the iterative method. For a stationary iteration of the type (2.1), $\alpha_s = 1$ is chosen for consistency. Since time accuracy is not important, the remaining parameters $\alpha_1, \dots, \alpha_{s-1}$ could be adjusted to achieve as large convergence rate as possible. Also, it may be possible to solve (1.4) directly, using a non-stationary iteration like GMRES.
2. Introduce some form of preconditioning. The effect of such a technique can in many cases be interpreted as improving the damping properties, increasing the error propagation speed, or both.

It is the belief of the authors that the choice of preconditioner is critical. As exemplified in this paper, even a very basic iteration scheme will be capable of computing a converged solution in a small number of arithmetic operations, if an efficient preconditioner is employed.

A preconditioned iteration for (1.4) could be defined by applying (2.1) to the problem

$$\frac{dv}{dt} + M^{-1} r(v) = 0. \tag{2.2}$$

Here, the matrix M specifies the preconditioner. The preconditioner solve $w = M^{-1} r(v)$ may be a *global* operation, where unknowns at many grid points at large relative distances in the grid are coupled. This corresponds to that M^{-1} is essentially a full matrix. In practice, M^{-1} is not explicitly formed, but its action is computed using an efficient solver. Examples of preconditioners of this type are the semicirculant approximations considered in this paper, and multigrid, see, e.g., [16] [4]. The matrix M^{-1} could also be diagonal or block-diagonal, representing *point* preconditioners coupling unknowns at the same grid point only, 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, see, e.g., [17], preconditioning of the system of PDE, see, e.g., [24], and residual smoothing, see, e.g., [16] [2].

For a given approximation v^0, the error $e = v - v^0$ satisfies the PDE

$$\frac{de}{dt} + M^{-1} B e = 0. \tag{2.3}$$

Applying (2.1) to this problem, we have that

$$e^{i+1} = Qe^i = Q^{i+1}e^0. \tag{2.4}$$

Here, $Q = p(kM^{-1}B)$, where $p(z)$ is the characteristic polynomial of the Runge-Kutta method.

From (2.4), it is clear that the error decreases if $\|Q^{i+1}\| < 1$. Let μ_ℓ, $\ell = 1, \ldots, m$, denote the eigenvalues of $M^{-1}B$. Under the assumption that $M^{-1}B$ is diagonalizable, we have $Q^{i+1} = W\Lambda^{i+1}W^{-1}$, where W is the eigenvector matrix of $M^{-1}B$, and $\Lambda = \mathrm{diag}\,(p(k\mu_\ell))$. This yields the bound

$$\left(\max_\ell |p(k\mu_\ell)|\right)^{i+1} \leq \|Q^{i+1}\| \leq \mathrm{cond}(W)\left(\max_\ell |p(k\mu_\ell)|\right)^{i+1}. \tag{2.5}$$

Hence, the iteration converges* if and only if all eigenvalues μ_ℓ lie inside the stability region for the Runge-Kutta method, i.e.,

$$\max_\ell |p(k\mu_\ell)| < 1. \tag{2.6}$$

If a family of matrices where $m \to \infty$ is studied, it is possible that *cond*(W) is not asymptotically bounded. In this case, (2.6) is only a necessary condition for convergence.

For an unpreconditioned problem, the wave propagation property of the PDE implies that the smallest eigenvalues of B approach the origin along the imaginary axis. The stability region of (2.1) should therefore include a significant segment of this axis, centered around the origin. Hence, $s \geq 3$ should be employed. In Figure 2.1, the stability region for the three-stage method defined by $\alpha_1 = \alpha_2 = 0.6$, $\alpha_3 = 1$ is shown. This method, which is often used in applications, will be denoted by RK3. When utilizing the semicirculant (SC) preconditioner, we will also employ the one-stage forward Euler method,

$$v^{i+1} = v^i - kM^{-1}r(v^i). \tag{2.7}$$

This method will be denoted by SC-RK1. The stability region is the unit circle centered around one, as shown in Figure 2.1.

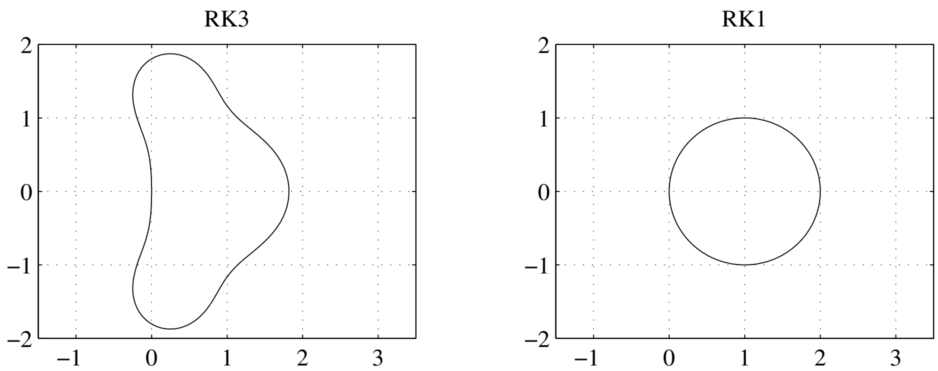

FIG. 2.1. *The stability regions for RK3 and RK1.*

*If cond(W) is very large, (2.5) indicates that the residual may initially grow so much that an overflow is generated in a practical computation. However, the right inequality in (2.5) is often found to be very pessimistic.

If no preconditioner is employed, the wave propagation property for the differential equation in combination with the CFL criterion for the explicit time-marching method implies that $\mathcal{O}(m)$ iterations are required for convergence, where m is the maximal number of grid points in one space dimension. For a preconditioned problem, the time accuracy is generally lost, and the time variable has no physical interpretation. A good preconditioner achieves *grid independent convergence* , i.e., the number of iterations required is independent of m, without introducing too many extra arithmetic operations in the preconditioner solve.

3. The semicirculant preconditioner. In this section, we will briefly review the construction of semicirculant approximations and the algorithms involved in the preconditioner solve for systems of n_c PDE in two space dimensions, discretized on an $m_1 \times m_2$ grid using a finite difference or finite volume method. For a detailed presentation, see, e.g, [13] and [14].

The discretization of (1.2) may be described by a computational stencil, whose extent is determined by the integers q_W, q_E, q_S, and q_N, see Fig. 3.1. In the formulas describing the arithmetic complexity and memory requirement, we will for simplicity assume that $q_W = q_E = q_S = q_N \equiv q$.

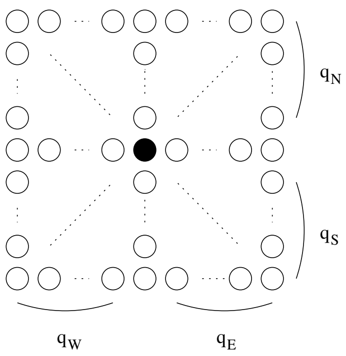

FIG. 3.1. *The maximal extent of the discretization stencil.*

Assume that the entries in the solution vector v in (1.4) are ordered first by equation in the PDE, then in the x_1-direction, and finally in the x_2-direction. Then, the matrix B has a two-level band structure. At the outer block level, the blocks are of size $n_c m_1 \times n_c m_1$, and the interior upper and lower block bandwidths are given by q_N and q_S. At the inner block level, the blocks are of size $n_c \times n_c$ and the interior upper and lower block bandwidths are given by q_E and q_W. The discretization of the boundary conditions may disturb the band structure slightly. In Figure 3.2, the matrix B resulting from the discretization of the linearized Euler equations presented in §6 is shown as an example.

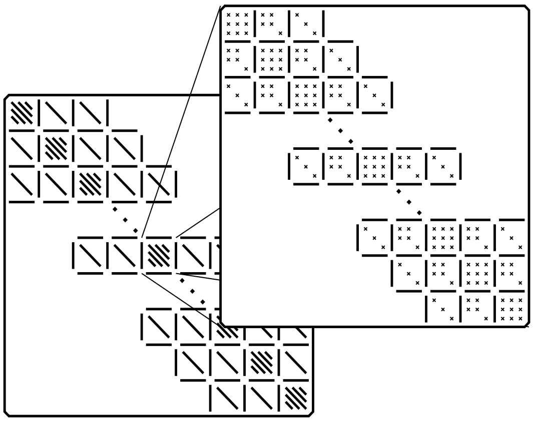

FIG. 3.2. *The matrix B arising from the discretization of the Euler equations in §6. Here, $n_c = 3$ and $q = 2$.*

The semicirculant approximation M is constructed using the original discretization applied to a modified PDE problem, where the boundary conditions in the x_1-direction are altered to periodic. Also, the coefficients in the difference approximation are averaged over this direction, implying that the matrix M has a circulant-block structure. The structure of M corresponding to Fig. 3.2 is shown in Fig. 3.3. Here, the outer level blocks, e.g., the expanded block, are circulant-block matrices.

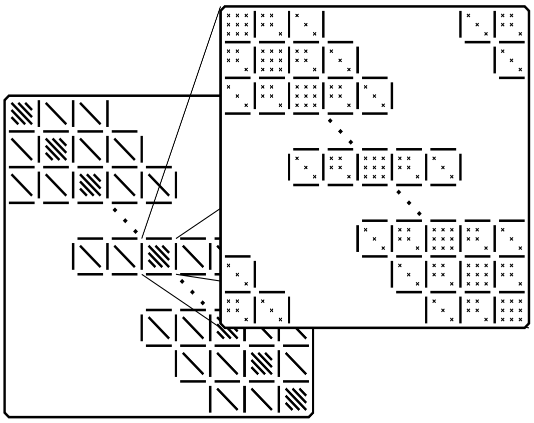

FIG. 3.3. *The structure of M corresponding to Fig. 3.2.*

As a result of the circulant structure, the non-zero entries in M may be stored in only $\mathcal{O}(m_2)$ memory positions. The arithmetic complexity per grid point required for the averaging is at most $\sim (2q+1)^2 n_c^2$. It is possible to take full advantage of the structure of the computational stencil and sparsity in the coefficients matrices in the PDE to reduce the operation count and the memory requirement, since the corresponding entries in M will be zero by construction.

The circulant-block structure of M also leads to the possibility of deriving an efficient algorithm for the preconditioner solve, $y = M^{-1}x$ [7]. The computations are divided into three phases:

1. Perform $n_c m_2$ independent FFTs of length m_1.
2. Solve $m_1/2+1$ independent block-banded systems of $n_c m_2$ equations (block bandwidth $2q+1$ and block size $n_c \times n_c$).
3. Perform $n_c m_2$ independent inverse FFTs of length m_1.

For factorizing the block-banded systems occurring in phase 2 of the preconditioner solve, $\sim 4n_c\left((q+1)n_c-1\right)^2$ a.o. per grid point are required. When the semicirculant preconditioner is employed for solving a linear PDE problem, the computation of the non-zero entries in M and the factorization of the block-banded system only have to be performed once, while for nonlinear problems, these computations possibly have to be performed every other iteration. The arithmetic complexity per grid point for the preconditioner solve, including back substitution for the block-banded systems in phase 2, is $\sim (5\log_2 m_1 + 12(q+1)n_c - 4)n_c$ if m_1 is a power of two, and slightly larger if this is not the case. The total arithmetic complexity is computed by adding contributions from the preconditioner solve and the residual evaluations. In the present implementation, the total memory requirement for the semicirculant solver is $\sim 3qn_c^2 m_1 m_2$. If the factorizations of the block-banded systems are not stored, an implementation requiring only $\sim 3qn_c^2 m_2$ memory positions is possible. Note that the potential parallelism in the solver is large. Parallel implementations are studied in [7].

4. The multigrid preconditioner. For comparison, we employ a standard multigrid method, given by the scheme in Equation (4.1.1) in [5]. This scheme is also often used in applications. For the model problem studied in §5, we use V-cycle iterations, while for the linearized Euler equations studied in §6, W-cycles yield better convergence properties and lower arithmetic complexity. At all grid levels, the same discretization procedure and CFL constant is employed. The prolongation and restriction operators are based on bilinear interpolation. For smoothing, we employ the Runge-Kutta scheme RK3, described in §2. One pre- and one post-smoothing iteration is applied, and on the coarsest grid one smoothing operation is performed instead of solving the problem exactly [21] [3]. We denote the resulting method by MG-RK3. The arithmetic complexities presented in §§5 and 6 are computed by summing up contributions from the smoother, the restriction and prolongation operators, and the residual evaluations.

5. A model problem. In this section, we study the semicirculant preconditioning technique applied to a scalar model problem. We consider the time-independent Kreiss equation on the unit square,

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

The PDE (5.1) is well posed if $u(x_1, 0)$ and $u(0, x_2)$ are prescribed.

The discretization is performed on a uniform grid with $(m_1+1)\times(m_2+1)$ grid points, implying that the space step in both directions is given by $h_d \equiv 1/m_d$, $d = 1, 2$. The approximate solution at grid point (jh_1, kh_2) is denoted by $u_{j,k}$. In the interior of the domain, the derivatives in (5.1) are approximated using second-order accurate centered differences, and artificial viscosity of standard type [17] is added. The difference approximation is given by

$$D_{0,x_d} - \frac{\gamma_2}{2} h_d D_{+,x_d} D_{-,x_d} + \frac{\gamma_4}{2} h_d^3 (D_{+,x_d} D_{-,x_d})^2\,, \quad d = 1, 2. \tag{5.2}$$

Here, $\gamma_{2,4} \geq 0$. The difference operators in the x_1-direction are defined by

$$
\begin{aligned}
D_{+,x_1} u_{j,k} &= \frac{1}{h_1}(u_{j+1,k} - u_{j,k}), \\
D_{-,x_1} u_{j,k} &= D_{+,x_1} u_{j-1,k}, \\
D_{0,x_1} &= \tfrac{1}{2}(D_{+,x_1} + D_{-,x_1}).
\end{aligned}
$$

The operators in the x_2-direction are defined correspondingly. Note that, if $\gamma_2 = 0$, (5.2) is a second-order accurate approximation of the first derivative, while if $\gamma_2 > 0$, the order of accuracy is reduced to one. For application problems, γ_4 is often chosen as a small positive constant, while γ_2 is non-zero only at grid points where the approximate solution has a large gradient, corresponding to, e.g., a shock.

For the numerical boundary conditions required at the outflow boundaries, we use one-sided differences. This does not destroy second-order accuracy in the interior of the domain. At the boundaries, no artificial viscosity is added in the direction normal to the boundary. Finally, at grid points next to the boundary, the fourth-order artificial viscosity operator must be modified.

By introducing the discretizations described above in (5.1), we arrive at a system of equations of the type (1.4), with $m_1 m_2$ unknowns. Since the difference approximation is separable, the coefficient matrix B may be written in Kronecker product form,

$$
B = \frac{1}{2h_1}\left(I_{m_2} \otimes B_{m_1} + \beta B_{m_2} \otimes I_{m_1}\right). \tag{5.3}
$$

Here, the grid quotient β is defined by

$$
0 < \beta \equiv \frac{h_1}{h_2} = \frac{m_2}{m_1} < \infty,
$$

and the $m_d \times m_d$-matrix B_{m_d}, $d = 1, 2$, is given by

$$
\begin{aligned}
B_{m_d} = &\begin{bmatrix}
0 & 1 & & & & & \\
-1 & 0 & 1 & & & & \\
 & -1 & 0 & 1 & & & \\
 & & \ddots & \ddots & \ddots & & \\
 & & & -1 & 0 & 1 & \\
 & & & & -1 & 0 & 1 \\
 & & & & & -2 & 2
\end{bmatrix} \\
+ \gamma_2 &\begin{bmatrix}
2 & -1 & & & & & \\
-1 & 2 & -1 & & & & \\
 & -1 & 2 & -1 & & & \\
 & & \ddots & \ddots & \ddots & & \\
 & & & -1 & 2 & -1 & \\
 & & & & -1 & 2 & -1 \\
 & & & & 0 & 0 & 0
\end{bmatrix}
\end{aligned}
$$

$$+\gamma_4 \begin{bmatrix} 5 & -4 & 1 & & & & \\ -4 & 6 & -4 & 1 & & & \\ 1 & -4 & 6 & -4 & 1 & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & 1 & -4 & 6 & -4 & 1 \\ & & & 1 & -4 & 5 & -2 \\ & & & & 0 & 0 & 0 \end{bmatrix}.$$

We will derive theoretic results for the case $\gamma_2 > 0$, $\gamma_4 = 0$, and give numerical results which shows that employing $\gamma_4 > 0$ yields the same type of favorable convergence properties as for the case covered by the theory.

The following lemma is a generalization of Lemma 6.1 in [22], where the result is established for the case $\gamma_2 = \gamma_4 = 0$.

LEMMA 5.1. *Assume that $\gamma_4 = 0$, and let the eigenvalues of B_{m_d} be denoted by λ_j, $j = 1, \dots, m_d$. Then,*

(i) $\lambda_j \neq \lambda_k$, $j \neq k$.
(ii) $\Re e(\lambda_j) \geq 2\gamma_2$, $\quad j = 1, \dots, m_d$.
(iii) *In the limit $m_d \to \infty$, λ_j approach the line*

$$2\gamma_2 + 2i\sqrt{1-\gamma_2^2}\sin\theta\ , \quad -\pi/2 \leq \theta \leq \pi/2.$$

Proof. The proof is found in Appendix A. □ Note that Lemma 5.1 implies that the matrix B_{m_d} is nonsingular and diagonalizable.

The semicirculant approximation corresponding to (5.3) is formed by altering the boundary conditions in the x_1-direction to periodic. For the case $\gamma_4 = 0$, this yields

$$M = \frac{1}{2h_1}\left(I_{m_2} \otimes \tilde{B}_{m_1} + B_{m_2} \otimes I_{m_1}\right),$$

where $\tilde{B}_{m_1}$ is given by

$$\tilde{B}_{m_1} = \begin{bmatrix} 0 & 1 & & & -1 \\ -1 & 0 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 0 & 1 \\ 1 & & & -1 & 0 \end{bmatrix} + \gamma_2 \begin{bmatrix} 2 & -1 & & & -1 \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ -1 & & & -1 & 2 \end{bmatrix}.$$

Since $\tilde{B}_{m_1}$ is circulant, we have

$$F_{m_1}^* \tilde{B}_{m_1} F_{m_1} = \tilde{\Lambda}_{m_1} = \mathrm{diag}_{j,m_1}(\tilde{\lambda}_j), \quad \tilde{\lambda}_j = (1-\gamma_2)\omega^{j-1} + 2\gamma_2 - (1+\gamma_2)\omega^{-(j-1)}.$$

Here, F_{m_1} is the m_1:th-order Fourier matrix and $\omega = exp(2\pi i/m_1)$.

THEOREM 5.2. *The matrix M is nonsingular and diagonalizable.*

Proof. From Lemma 5.1 we conclude that there exists a nonsingular matrix V_{m_2} diagonalizing B_{m_2}, that is

$$V_{m_2}^{-1} B_{m_2} V_{m_2} = \Lambda_{m_2} = \mathrm{diag}_{k,m_2}(\lambda_k).$$

This implies that

$$\begin{aligned}(V_{m_2}^{-1}\otimes F_{m_1}^*)M(V_{m_2}\otimes F_{m_1}) &= \frac{1}{2h_1}(V_{m_2}^{-1}\otimes F_{m_1}^*)(I_{m_2}\otimes\tilde{B}_{m_1}+\beta B_{m_2}\otimes I_{m_1})(V_{m_2}\otimes F_{m_1})\\ &= \frac{1}{2h_1}\left(I_{m_2}\otimes F_{m_1}^*\tilde{B}_{m_1}F_{m_1}+\beta V_{m_2}^{-1}B_{m_2}V_{m_2}\otimes I_{m_1}\right)\\ &= \frac{1}{2h_1}\left(I_{m_2}\otimes\tilde{\Lambda}_{m_1}+\Lambda_{m_2}\otimes I_{m_1}\right).\end{aligned}$$

Thus, M is diagonalizable, and the eigenvalues of M are $(2h_1)^{-1}(\tilde{\lambda}_j+\beta\lambda_k)$, $j=1,\ldots,m_1$, $k=1,\ldots,m_2$. From the formula for $\tilde{\lambda}_j$ and Lemma 5.1, it follows that $\Re e(\tilde{\lambda}_j+\beta\lambda_k)>2\gamma_2$, hence all eigenvalues are non-zero, which implies that M is nonsingular. ☐

It is possible to derive analytic formulas for the eigenvalues of $M^{-1}B$. Define the error matrix $E\equiv B-M$. Since $M^{-1}B=M^{-1}(M+E)=I+M^{-1}E$, examining the eigendecomposition of $M^{-1}E$ will suffice. Let δ be given by

$$\delta\equiv\frac{(1+\gamma_2)}{(1-\gamma_2)}.$$

THEOREM 5.3. *Assume that $\gamma_4=0$ and $0<\gamma_2<1$. Define ζ_k by*

$$\zeta_k\equiv\frac{\gamma_2}{1-\gamma_2}+\frac{\beta\lambda_k}{2(1-\gamma_2)}.$$

Then the matrix $M^{-1}E$ has the eigenvalue zero with multiplicity at least $m_2(m_1-2)$, and $2m_2$ eigenvalues given by

$$\mu_{(1,2),k}=$$
$$\frac{2-(2-\delta^{-1})z_k+R_1(z_k)\pm\sqrt{-(4\delta^{-1}-\delta^{-2})z_k^2-4(2-\delta^{-1})z_k-4(\delta-1)+R_2(z_k)}}{2(z_k+\delta z_k^{-1})},$$
$$k=1,\ldots,m_2,$$

where

$$R_1(z_k)=\big((\delta-1)z_k^{-1}+2-z_k\big)\frac{z_k^m}{1-z_k^m}+\big((\delta^{-1}-1)z_k+2+\delta z_k^{-1}\big)\frac{(-\delta^{-1}z_k)^m}{1-(-\delta^{-1}z_k)^m},$$

$$\begin{aligned}R_2(z_k)&=\big(-6\delta^{-1}z_k^2-(12-4\delta^{-1})z_k-8\delta+6-2\delta^{-1}+(4\delta-4)z_k^{-1}-4\delta z_k^{-2}\big)\frac{z_k^m}{1-z_k^m}\\ &+\big(-(6\delta^{-1}-2\delta^{-2})z_k^2-(12-8\delta^{-1})z_k-8\delta+10+4\delta z_k^{-1}\big)\frac{(-\delta^{-1}z_k)^m}{1-(-\delta^{-1}z_k)^m}\\ &+\big(z_k^2-4z_k-2\delta+6+(4\delta-4)z_k^{-1}+(\delta-1)^2z_k^{-2}\big)\Big(\frac{z_k^m}{1-z_k^m}\Big)^2\\ &+\big((\delta^{-1}-1)^2z_k^2+(4\delta^{-1}-4)z_k-2\delta+6+4\delta z_k^{-1}+\delta^2z_k^{-2}\big)\Big(\frac{(-\delta^{-1}z_k)^m}{1-(-\delta^{-1}z_k)^m}\Big)^2\\ &+\big(-(2+6\delta^{-1})z_k^2-(8-4\delta^{-1})z_k-12\delta+4-2\delta^{-1}+(8\delta-4)z_k^{-1}-(2\delta^2+6\delta)z_k^{-2}\big)\\ &\cdot\frac{z_k^m}{1-z_k^m}\cdot\frac{(-\delta^{-1}z_k)^m}{1-(-\delta^{-1}z_k)^m}\end{aligned}$$

and finally

$$z_k = -\zeta_k + \sqrt{\delta + \zeta_k^2}.$$

Proof. The proof is a minor modification of the proof of Theorem 2 in [11], which employs Lemma 2 in the same paper. Below, we give a brief review, and refer to [11] for further details.

First, a sparse, block-diagonal matrix $T = \mathrm{diag}_{k,m}(T_k)$, which is similar to $M^{-1}B$, is constructed. The non-zero entries in T_k are computed in terms of the quantities σ_p, $p = 1, \ldots, m_1$, where

$$\sigma_p = m_1^{-1} \sum_{j=1}^{m_1} \frac{\omega_{m_1}^{p(j-1)}}{2\zeta_k + \omega_{m_1}^{(j-1)} - \delta\omega_{m_1}^{-(j-1)}}.$$

From Lemma 5.1 it is clear that $\Re e(\zeta_k) > (\delta - 1)/2$, implying that the assumptions in Lemma 2 in [11] are fulfilled. By employing this lemma, the sums σ_p are computed utilizing the Poisson summation formula and the residue theorem. Solving the characteristic equations of T_k then yields the eigenvalues of $M^{-1}E$. □

We now utilize Theorem 5.3 for deriving an asymptotic expression for the eigenvalues of $M^{-1}E$.

THEOREM 5.4. *Assume that $\gamma_4 = 0$ and $0 < \gamma_2 < 1$. Then, in the limit $m_{1,2} \to \infty$, the non-zero eigenvalues of $M^{-1}E$ approach two curve segments given by*

$$\begin{aligned} \mu_{(1,2)}(z) &= \frac{2 - (2 - \delta^{-1})z \pm \sqrt{-(4\delta^{-1} - \delta^{-2})z^2 - 4(2 - \delta^{-1})z - 4(\delta - 1)}}{2(z + \delta z^{-1})}, \\ z(\zeta) &= -\zeta + \sqrt{\delta + \zeta^2}, \\ \zeta(\theta) &= \frac{(1+\beta)\gamma_2 + i\beta\sqrt{1 - \gamma_2^2}\sin\theta}{1 - \gamma_2}, \quad -\frac{\pi}{2} \le \theta \le \frac{\pi}{2}. \end{aligned}$$

Proof. From Lemma 5.1 it is clear that, as $m_2 \to \infty$, the points ζ_k, $k = 1, \ldots, m_1$, approach the line segment $\zeta(\theta)$. It is easily verified that the mapping $z(\zeta) = -\zeta + \sqrt{\delta + \zeta^2}$ transforms the semi-ellipses

$$\frac{1}{2}\left(\frac{\delta}{r}\exp(-i\phi) - r\exp(i\phi)\right), \quad 0 < r \le 1, \quad -\frac{\pi}{2} \le \phi \le \frac{\pi}{2},$$

to the semi-circles $r\exp(i\phi)$. Since the line segment $\zeta(\theta)$ is parallel to the imaginary axis, and $\Re e(\zeta(\theta)) > \gamma_2/(1 - \gamma_2) > 0$, this implies that

$$|z(\zeta(\theta))| \le |z(\zeta(0))| \equiv z_{max}(\beta, \gamma_2), \tag{5.4}$$

where

$$z_{max}(\beta, \gamma_2) = -\frac{(1+\beta)\gamma_2}{1 - \gamma_2} + \sqrt{\frac{1 + \gamma_2}{1 - \gamma_2} + \frac{(1+\beta)^2\gamma_2^2}{(1 - \gamma_2)^2}}.$$

A straightforward computation yields that $z_{max}(0,\gamma_2) \equiv 1$. For $0 < \beta < \infty$ and $0 < \gamma_2 < 1$, we have that

$$\frac{dz_{max}}{d\beta} = -\left(1 - \frac{1}{\sqrt{1 + \dfrac{(1+\gamma_2)(1-\gamma_2)}{(1+\beta)^2\gamma_2^2}}}\right)\frac{\gamma_2}{1-\gamma_2} < 0$$

Consequently $|z(\zeta(\theta))| < 1$, implying that

$$\lim_{m_1\to\infty}(z(\theta))^{m_1} = \lim_{m_1\to\infty}\left(-\delta^{-1}z(\theta)\right)^{m_1} = 0\,, \quad -\frac{\pi}{2} \le \theta \le \frac{\pi}{2},$$

which implies that the oscillatory terms $R_{1,2}$ in Theorem 5.3 vanishes as $m_1 \to \infty$, and the result follows. ☐

Figure 5.1 describes the convergence of the eigenvalues $1+\mu_{(1,2),k}$, computed using Theorem 5.3, to the curves $1+\mu_{1,2}(\theta)$ for the case $\beta = 1$, $\gamma_2 = 1/32$. The convergence rate depends on the choice of γ_2 and β. Experiments show that the convergence is slower for small values of γ_2, and for $\beta \ll 1$ or $\beta \gg 1$. In the figure, the eigenvalues for four different problem sizes $m_{1,2}$ are marked '+', while the solid lines represent the curves $1+\mu_{1,2}(\theta)$.

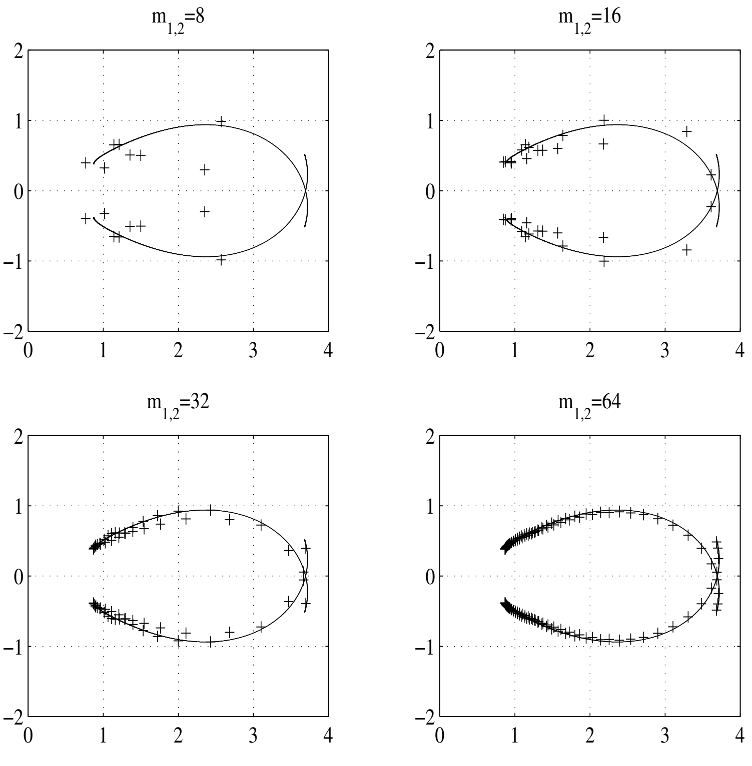

FIG. 5.1. *The eigenvalues of $M^{-1}B$ for different values of $m_{1,2}$. Here, $\gamma_2 = 1/32$ and $\gamma_4 = 0$.*

For $m_{1,2}$ sufficiently large, the convergence criterion (2.6) for the SC-RK1 iteration is effectively given by

$$|1 - k(1 + \mu_{(1,2)}(\theta))| < 1 \ , \quad -\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2}. \tag{5.5}$$

By employing a discretization of the parameter θ, a numerical solution of the parameterized one-variable equation

$$\max_{-\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2}} |1 - k(1 + \mu_{(1,2)}(\theta))| = 1 \tag{5.6}$$

yields a sharp stability bound $k_{max}(\beta, \gamma_2)$ for the time step k.

It is probably not possible to bound the condition number of the eigenvector matrix for $M^{-1}B$ by a constant, independent of $m_{1,2}$. Following the discussion in §2, this implies that (5.5) is possibly only a necessary condition for convergence. We now present numerical results showing that the convergence properties of the preconditioned forward Euler method is indeed described by the analysis above. We solve (2.3) for the model problem (5.1), using the SC-RK1 scheme. The iteration is stopped when the norm

$$||e|| = \left(\sum_{i,j} e_{i,j}^2 h_1 h_2 \right)^{1/2}$$

has been decreased by a factor 10^6. The initial error is given by a random function.

In Tables 5.1 and 5.2, results from a grid refinement study are presented. Here, $\beta = 1$ is chosen, and (5.6) yields $k_{max} = 0.5332$. In the tables, the optimal choice of the time step k determined from numerical experiments, the number of SC-RK1 iterations, and the corresponding number of arithmetic operations per grid point are shown. In Table 5.1, the case $\gamma_2 = 1/32$, $\gamma_4 = 0$, which is covered by the analysis above, is considered. In Table 5.2, the corresponding results for $\gamma_2 = 0$, $\gamma_4 = 1/32$ are shown.

TABLE 5.1
Grid refinement study for SC-RK1, $\gamma_2 = 1/32$ and $\gamma_4 = 0$.

$m_{1,2}$	8	16	32	64	128	256	512	1024
k_{opt}	0.53	0.47	0.43	0.45	0.45	0.45	0.45	0.45
# iterations	20	22	23	24	24	24	24	24
# a.o. per grid point $[10^3]$	1.18	1.41	1.59	1.78	1.90	2.02	2.14	2.26

TABLE 5.2
Grid refinement study for SC-RK1, $\gamma_2 = 0$ and $\gamma_4 = 1/32$.

$m_{1,2}$	8	16	32	64	128	256	512	1024
k_{opt}	0.63	0.58	0.56	0.55	0.54	0.54	0.54	0.54
# iterations	16	17	18	19	19	19	19	19
# a.o. per grid point $[10^3]$	1.07	1.22	1.39	1.56	1.65	1.75	1.84	1.94

Table 5.1 shows that the number of iterations required when employing the SC-RK1 method is bounded as $m_{1,2}$ grows. Grid independent convergence is achieved, as predicted by the analysis. The number of arithmetic operations per grid point increases

slightly, mainly because of the $\log_2(m_1)$-term in the formula for the arithmetic complexity. However, note that the increase is by less then a factor of two when the number of grid points is increased by a factor of 2^{14}. From Table 5.2, it is clear that the same type of grid independent convergence property also holds for the case when fourth-order artificial viscosity is used, which is not covered by the analysis.

When employing the SC preconditioner, it is not advantageous to use the three-stage Runge-Kutta iteration RK3. In Table 5.3, the results corresponding to Table 5.2 are shown for the SC-RK3 iteration. From the table, it is clear that the number of iterations required actually increases slightly, and since each iteration is three times more expensive, the number of arithmetic operations is significantly increased.

TABLE 5.3
Grid refinement study for SC-RK3, $\gamma_2 = 0$ and $\gamma_4 = 1/32$.

$m_{1,2}$	8	16	32	64	128	256	512	1024
k_{opt}	0.67	0.59	0.60	0.58	0.55	0.55	0.54	0.54
# iterations	21	23	23	24	25	25	26	26
# a.o. per grid point [10^3]	4.22	4.97	5.31	5.90	6.52	6.90	7.57	7.96

We now study how the error vanishes in the iteration process, comparing the RK3, MG-RK3 and SC-RK1 iterations. In the experiments, $\beta = 1$, $\gamma_2 = 0$, and $\gamma_4 = 1/32$ are chosen. We start with an initial error given by a Gaussian pulse,

$$e(x_1, x_2) = e^{-50((x_1-0.5)^2+(x_2-0.5)^2)}. \tag{5.7}$$

For the unpreconditioned RK3 iteration, a time step $k = \mathrm{CFL}/(m_1 + m_2)$ should be used. Experiments show that the optimal value for the CFL constant is 1.8. In Table 5.4, the number of iterations and the corresponding number of arithmetic operations per grid point are shown for the RK3 method.

TABLE 5.4
Grid refinement study for RK3. Here, CFL=1.8.

$m_{1,2}$	8	16	32	64	128	256	512	1024
# iterations	82	74	52	73	119	242	481	958
# a.o. per grid point [10^3]	4.92	4.44	3.12	4.38	7.14	14.5	28.9	57.6

From Table 5.4, it is clear that for $m_{1,2}$ sufficiently large, the number of iterations and the arithmetic work grows linearly with $m_{1,2}$, as predicted in §2. In Figure 5.2, the error in iterations 0, 15, 30 and 45 is shown for the case $m_{1,2} = 64$.

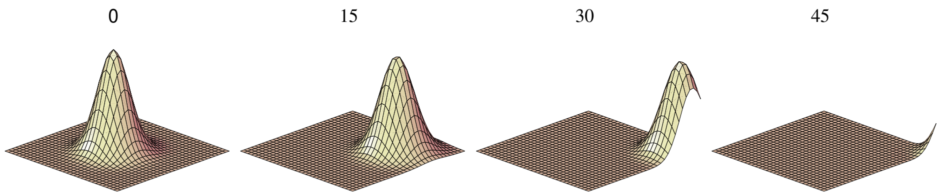

FIG. 5.2. *The solution after 0, 15, 30 and 45 iterations using the RK3 method. Here, $m_{1,2} = 64$ and CFL $= 1.8$.*

In the time-accurate computation shown in Figure 5.2, there is almost no damping and the pulse travels without changing shape. Convergence is achieved when the pulse propagates out of the domain. This is also clear from the error history in Figure 5.5, where the error starts to decrease after ~ 35 iterations, corresponding to time $T \approx 0.5$. The pulse is traveling with the velocity $\sqrt{2}$, implying that the top of the pulse has reached the boundary of the domain at this point in time.

In Table 5.5, a grid refinement study for the MG-RK3 iteration is presented. The number of grid levels are chosen to give convergence in a minimal number of iterations. It is not advantageous to employ all available grid levels. The solution on the coarsest grid must be reasonably similar to the fine grid solution to contribute to the convergence [19].

TABLE 5.5
Grid refinement study for MG-RK3.

$m_{1,2}$	8	16	32	64	128	256	512	1024
optimal # grid levels	1	2	2	2	3	3	4	4
# iterations	82	32	20	24	28	27	38	44
# a.o. per grid point [10^3]	4.92	5.06	3.16	3.79	4.38	5.11	6.95	8.05

From Table 5.5, it is clear that employing MG-RK3 results in a convergence rate which is roughly independent of the number of grid points. This is also consistent with the analysis and experiments in [19]. As for the SC-RK1 iteration, the number of arithmetic operations per grid point is approximately doubled when $m_{1,2}$ grows from 8 to 1024. For the MG-RK3 method, the growth is mainly due to the work performed on the coarser grids.

In Figure 5.3, the error at iterations 0, 3, 6 and 9 for the two-grid MG-RK3-iteration is shown.

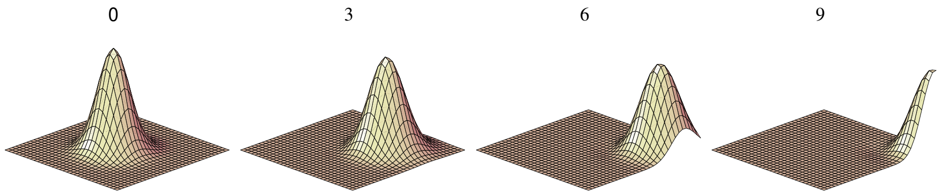

FIG. 5.3. *The solution after 0, 3, 6 and 9 iterations using the two-grid MG-RK3 method. Here, $m_{1,2} = 64$ and CFL=1.8.*

In Figure 5.3, we see that the error propagation speed is increased by a factor of four compared to the unpreconditioned RK3 iteration, which is also the result obtained from the analysis in [19]. In Figure 5.5, this property is further illustrated by that the norm of the error for the MG-RK3 method starts to decrease already after ~ 9 iterations.

The effect of the SC-RK1 iteration on the evolution of the solution is completely different from that of the RK3 and MG-RK3 methods. A grid refinement study is shown in Table 5.2. In Figure 5.4, the error at iterations 0, 1, 2 and 4 for the SC-RK1 iteration is shown.

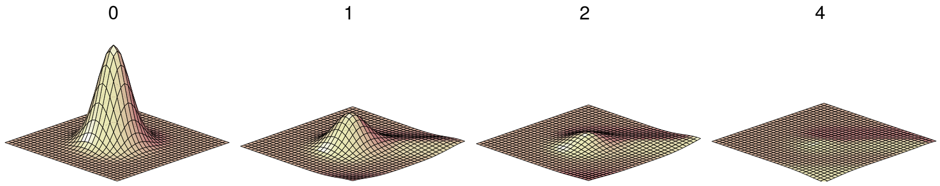

FIG. 5.4. *The solution after 0, 1, 2 and 4 iterations using the SC-RK1 method. Here,* $m_{1,2} = 64$ *and* $k = 0.55$.

From Figure 5.4, it is clear that the error is effectively damped, and no wave propagation can be seen. In Figure 5.5, the error history for the RK3, MG-RK3 and SC-RK1 methods are shown for the solution of a problem where $m_{1,2} = 64$.

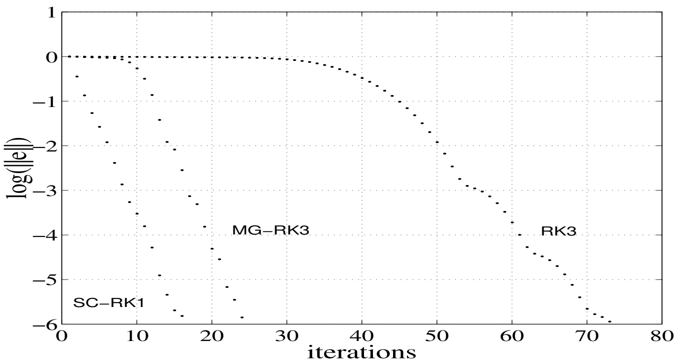

FIG. 5.5. $||e||$ *for the RK3, two-grid MG-RK3 and SC-RK1 method. Here,* $m_{1,2} = 64$.

As explained above, the plateaus at the beginning of the RK3 and MG-RK3 curves in Figure 5.5 originate from that the error is propagated out of the domain. For the SC-RK1 iteration, convergence is essentially achieved by damping, which is active already from the first iteration.

In Figure 5.6, the arithmetic complexities for the RK3, MG-RK3 and SC-RK1 iterations are shown for the grid refinement studies presented above.

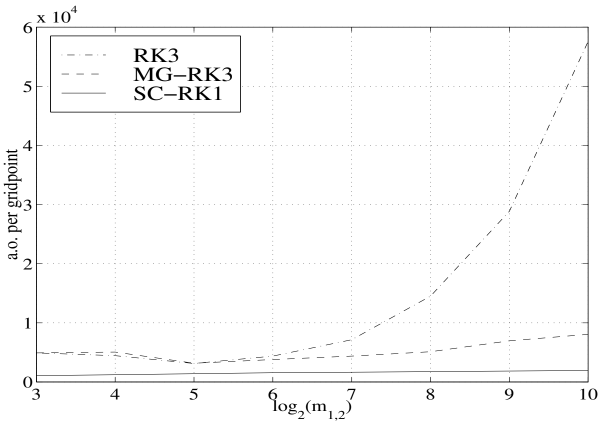

FIG. 5.6. *The number of a.o. required for the RK3, MG-RK3 and SC-RK1 iterations.*

From Figure 5.6, it is clear that the SC-RK1 method requires the smallest number of arithmetic operations for all problem sizes. For $m_{1,2} = 1024$, the SC-RK1 method is more than four times less demanding than the MG-RK3 iteration.

Finally, we study the convergence properties for the SC-RK1 iterations when different amounts of artificial viscosity is added in the difference approximation. In Figure 5.7, (5.6) is used for determining k_{max} as a function of γ_2, for three different values of the grid quotient β.

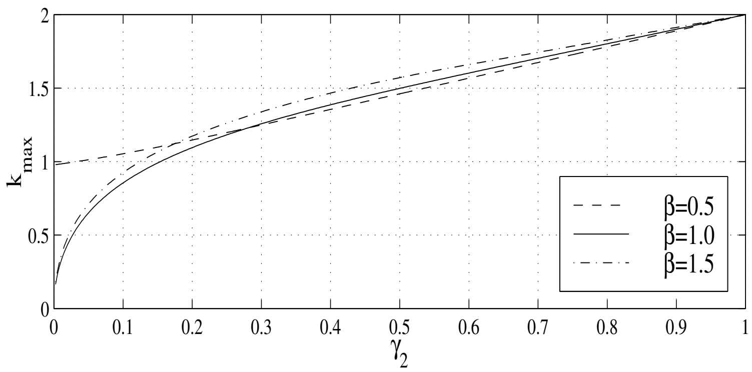

FIG. 5.7. *k_{max} as a function of γ_2 for different values of β.*

Further study of the formulas in Theorem 5.4 and the convergence criterion (5.5) verifies that, for $\beta \geq 1$, the time step bound k_{max} approaches zero when the artificial viscosity parameter is decreased. As γ_2 approaches zero, the curve $z(\theta)$ becomes a segment of the boundary of the semi-disc $\{z : |z| \leq 1, \Re e(z) \geq 0\}$. If $\beta \geq 1$, this segment includes the points $\pm i$, which are the poles for $\mu_{1,2}(\theta)$. Hence, $\mu_{1,2}(\theta)$ will grow, implying that k_{max} will approach zero. If $\beta < 1$, the segment of the semi-disc boundary does not include the points $\pm i$, and the curves $\mu_{(1,2)}$ will remain bounded. For example, the dashed curve in Figure 5.7 shows that, if $\beta = 0.5$ is used, $k_{max} > 0.97$ for all values of $\gamma_2 > 0$. This result is consistent with the result presented in [11] where, under the assumption $\beta < 1$, the spectrum is proved to be bounded for a case where the artificial viscosity vanishes as $m_1 \to \infty$.

Tables 5.6 and 5.7 show the time step bound k_{max} derived from (5.6), the optimal choice of k determined from numerical experiments, the number of iterations, and the corresponding number of arithmetic operations per grid point. The results are presented for different values of γ_2. In the experiments, grids where $m_1 = 1024$ and $m_2 = \beta m_1$ are employed. In Table 5.6, $\beta = 1$ is used, while in Table 5.7, $\beta = 1/2$ is utilized. In Table 5.8, the results corresponding to Table 5.7, but for fourth order artificial viscosity, are shown.

TABLE 5.6
Convergence study for SC-RK1. Here, $\gamma_4 = 0$ and $m_1 = m_2 = 1024$, i.e., $\beta = 1$.

γ_2	1/8	1/16	1/32	1/64	1/128	1/256	1/512	1/1024
k_{max}	0.93	0.71	0.53	0.39	0.28	0.20	0.15	0.10
k_{opt}	0.68	0.57	0.45	0.35	0.26	0.20	0.14	0.10
# iterations	14	18	24	34	47	62	92	132
# a.o. per grid point [10^3]	1.32	1.69	2.26	3.20	4.42	5.83	8.65	12.4

TABLE 5.7
Convergence study for SC-RK1. Here, $\gamma_4 = 0$, $m_1 = 1024$ and $m_2 = 512$, i.e., $\beta = 1/2$.

γ_2	1/8	1/16	1/32	1/64	1/128	1/256	1/512	1/1024
k_{max}	1.08	1.02	1.00	0.99	0.98	0.98	0.98	0.98
k_{opt}	0.72	0.68	0.68	0.68	0.68	0.62	0.55	0.55
# iterations	13	13	14	15	17	18	19	19
# a.o. per grid point $[10^3]$	1.22	1.22	1.32	1.41	1.60	1.69	1.77	1.77

TABLE 5.8
Convergence study for SC-RK1. Here, $\gamma_2 = 0$, $m_1 = 1024$ and $m_2 = 512$, i.e., $\beta = 1/2$.

γ_4	1/8	1/16	1/32	1/64	1/128	1/256	1/512	1/1024
k_{opt}	0.82	0.71	0.65	0.64	0.62	0.62	0.61	0.52
# iterations	14	14	14	14	15	16	17	22
# a.o. per grid point $[10^3]$	1.43	1.43	1.43	1.43	1.53	1.63	1.73	2.24

From Table 5.6, it is clear that if $\beta = 1$, the number of iterations increases when the viscosity parameter γ_2 is decreased. For $\beta = 0.5$, Table 5.7 shows that the convergence rate is almost unchanged by decreasing γ_2. These results are consistent with the analytic results presented in Figure 5.7. As shown in Table 5.8, the convergence properties are again almost unchanged by using fourth- instead of second-order artificial viscosity.

The results for the MG-RK3 iteration are shown in Table 5.9.

TABLE 5.9
Convergence study for MG-RK3. Here, $\gamma_2 = 0$, $m_1 = 1024$ and $m_2 = 512$, i.e., $\beta = 1/2$.

γ_4	1/8	1/16	1/32	1/64	1/128	1/256	1/512	1/1024
Optimal # grid levels	4	4	4	3	3	3	2	2
CFL	1.5	1.9	1.8	1.8	1.8	1.8	1.8	1.8
# iterations	39	32	40	72	85	120	200	252
# a.o. per grid point $[10^3]$	7.36	6.04	7.54	13.1	15.5	21.9	31.6	39.8

As shown in Table 5.9, both the number of grid levels and the CFL constant has to be varied to get acceptable convergence properties for the MG-RK3 iteration. For small values of γ_4, the convergence is even so quite slow. In Figure 5.8, the arithmetic complexities for the MG-RK3 and SC-RK1 iterations are shown for the convergence study presented above.

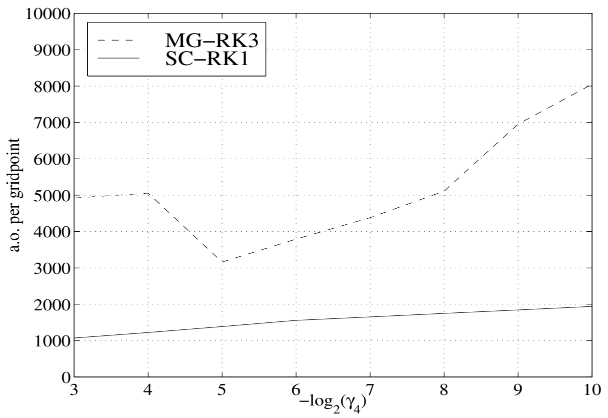

FIG. 5.8. *The number of a.o. required for the MG-RK3 and SC-RK1 iterations.*

From Figure 5.8, it is clear that the SC-RK1 method requires the smallest number of arithmetic operations for all values of γ_4. Also, the SC-RK1 method is less sensitive to changes in the artificial viscosity.

6. The linearized Euler equations. In this section, we consider the steady-state Euler equations

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

where $w = (\rho, v_1, v_2)$, ρ is the density and v_1 and v_2 are the velocity components. Furthermore,

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

and

$$A_2(w) = \begin{pmatrix} v_2 & 0 & \rho \\ 0 & v_2 & 0 \\ c^2/\rho & 0 & v_2 \end{pmatrix},$$

where c is the speed of sound. We have assumed isentropic flow, that is, the pressure p is given by $p = \beta\rho^\gamma$ for some constants β and γ. Using this assumption, the speed of sound is given by $c = \sqrt{\gamma p/\rho}$.

We linearize the Euler equations around an approximate solution $W = (R, V_1, V_2)$, which may vary in space. The linearized equations may be written

$$\tilde{A}_1\frac{\partial \tilde{w}}{\partial x_1} + \tilde{A}_2\frac{\partial \tilde{w}}{\partial x_2} + \tilde{A}_3\tilde{w} + \tilde{F} = 0, \tag{6.2}$$

where $\tilde{w} = (\tilde{\rho}, \tilde{v}_1, \tilde{v}_2)$ is the departure from the approximate solution,

$$\begin{aligned}
\tilde{A}_1 &= \begin{pmatrix} V_1 & c & 0 \\ c & V_1 & 0 \\ 0 & 0 & V_1 \end{pmatrix}, \\
\tilde{A}_2 &= \begin{pmatrix} V_2 & 0 & c \\ 0 & V_2 & 0 \\ c & 0 & V_2 \end{pmatrix}, \\
\tilde{A}_3 &= \begin{pmatrix} \dfrac{\partial V_1}{\partial x_1} + \dfrac{\partial V_2}{\partial x_2} & \dfrac{c}{R}\dfrac{\partial R}{\partial x_1} & \dfrac{c}{R}\dfrac{\partial R}{\partial x_2} \\ \dfrac{V_1}{c}\dfrac{\partial V_1}{\partial x_1} + \dfrac{V_2}{c}\dfrac{\partial V_1}{\partial x_2} & \dfrac{\partial V_1}{\partial x_1} & \dfrac{\partial V_1}{\partial x_2} \\ \dfrac{V_1}{c}\dfrac{\partial V_2}{\partial x_1} + \dfrac{V_2}{c}\dfrac{\partial V_2}{\partial x_2} & \dfrac{\partial V_2}{\partial x_1} & \dfrac{\partial V_2}{\partial x_2} \end{pmatrix}
\end{aligned}$$

and

$$\tilde{F} = \begin{pmatrix} c\dfrac{\partial V_1}{\partial x_1} + c\dfrac{\partial V_2}{\partial x_2} \\ V_1\dfrac{\partial V_1}{\partial x_1} + V_2\dfrac{\partial V_1}{\partial x_2} \\ V_1\dfrac{\partial V_2}{\partial x_1} + V_2\dfrac{\partial V_2}{\partial x_2} \end{pmatrix}.$$

If the approximation W varies slowly, $\tilde{A}_3$ and $\tilde{F}$ are close to zero. If also V_1 and V_2 are large, the matrices $\tilde{A}_1$ and $\tilde{A}_2$ are close to diagonal, and the system (6.2) degenerates into three uncoupled equations of the type (5.1).

We consider subsonic flow in a channel, that is, the Mach number is less than one at all points in the flow. At the left boundary there are two ingoing characteristics and at the right boundary there is one. Prescribing these, and setting the velocity perpendicular to the walls to be zero, results in the boundary conditions

$$\begin{array}{ll} \tilde{\rho}+\tilde{v}_1=\tilde{v}_2=0 & \text{at left boundary,} \\ \tilde{\rho}-\tilde{v}_1=0 & \text{at right boundary,} \\ \hat{n}\cdot(\tilde{v}_1,\tilde{v}_2)=0 & \text{at walls,} \end{array} \tag{6.3}$$

yielding a well posed problem.

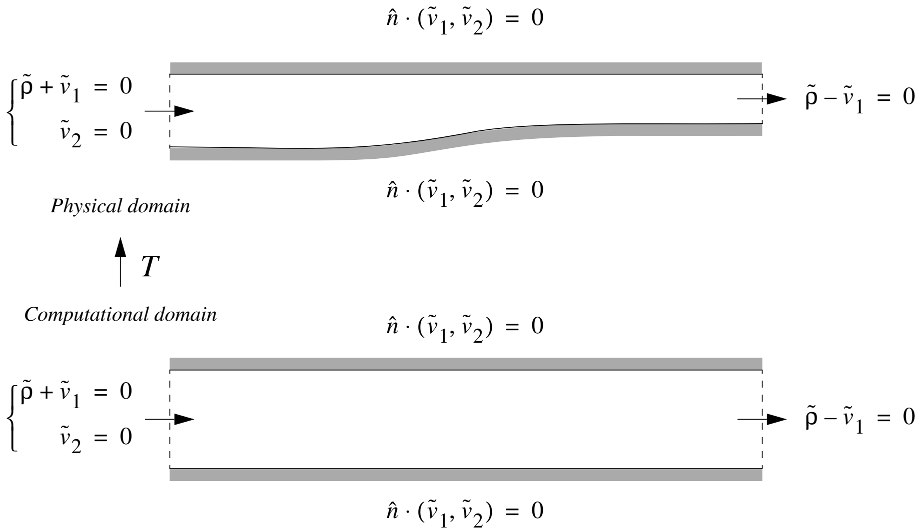

FIG. 6.1. *The physical and the computational domain.*

We consider both a straight and a narrowing channel setting. 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 is 0.5.

In order to simplify the discretization procedure, we introduce an orthogonal mapping T

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

between the physical domain and a rectangular computational domain as shown in Figure 6.1.

The transformed equations are

$$\bar{A}_1\frac{\partial\tilde{w}}{\partial\xi_1}+\bar{A}_2\frac{\partial\tilde{w}}{\partial\xi_2}+\bar{A}_3\tilde{w}+\bar{F}=0,$$

where

$$\begin{aligned}
\bar{A}_1 &= \frac{1}{s}\left(\frac{\partial x_2}{\partial \xi_2}\tilde{A}_1 - \frac{\partial x_1}{\partial \xi_2}\tilde{A}_2\right), \\
\bar{A}_2 &= \frac{1}{s}\left(\frac{\partial x_1}{\partial \xi_1}\tilde{A}_2 - \frac{\partial x_2}{\partial \xi_1}\tilde{A}_1\right),
\end{aligned}$$

and the components in $\bar{A}_3$ are given by

$$\begin{aligned}
[\bar{A}_3]_{11} &= \frac{1}{s}\left(\frac{\partial x_2}{\partial \xi_2}\frac{\partial V_1}{\partial \xi_1} - \frac{\partial x_2}{\partial \xi_1}\frac{\partial V_1}{\partial \xi_2} - \frac{\partial x_1}{\partial \xi_2}\frac{\partial V_2}{\partial \xi_1} + \frac{\partial x_1}{\partial \xi_1}\frac{\partial V_2}{\partial \xi_2}\right), \\
[\bar{A}_3]_{12} &= \frac{1}{s}\frac{c}{R}\left(\frac{\partial x_2}{\partial \xi_2}\frac{\partial R}{\partial \xi_1} - \frac{\partial x_2}{\partial \xi_1}\frac{\partial R}{\partial \xi_2}\right), \\
[\bar{A}_3]_{13} &= \frac{1}{s}\frac{c}{R}\left(-\frac{\partial x_1}{\partial \xi_2}\frac{\partial R}{\partial \xi_1} + \frac{\partial x_1}{\partial \xi_1}\frac{\partial R}{\partial \xi_2}\right), \\
[\bar{A}_3]_{21} &= \frac{1}{s}\frac{V_1}{c}\left(\frac{\partial x_2}{\partial \xi_2}\frac{\partial V_1}{\partial \xi_1} - \frac{\partial x_2}{\partial \xi_1}\frac{\partial V_1}{\partial \xi_2}\right) + \frac{1}{s}\frac{V_2}{c}\left(-\frac{\partial x_1}{\partial \xi_2}\frac{\partial V_1}{\partial \xi_1} + \frac{\partial x_1}{\partial \xi_1}\frac{\partial V_1}{\partial \xi_2}\right), \\
[\bar{A}_3]_{22} &= \frac{1}{s}\left(\frac{\partial x_2}{\partial \xi_2}\frac{\partial V_1}{\partial \xi_1} - \frac{\partial x_2}{\partial \xi_1}\frac{\partial V_1}{\partial \xi_2}\right), \\
[\bar{A}_3]_{23} &= \frac{1}{s}\left(-\frac{\partial x_1}{\partial \xi_2}\frac{\partial V_1}{\partial \xi_1} + \frac{\partial x_1}{\partial \xi_1}\frac{\partial V_1}{\partial \xi_2}\right), \\
[\bar{A}_3]_{31} &= \frac{1}{s}\frac{V_1}{c}\left(\frac{\partial x_2}{\partial \xi_2}\frac{\partial V_2}{\partial \xi_1} - \frac{\partial x_2}{\partial \xi_1}\frac{\partial V_2}{\partial \xi_2}\right) + \frac{1}{s}\frac{V_2}{c}\left(-\frac{\partial x_1}{\partial \xi_2}\frac{\partial V_2}{\partial \xi_1} + \frac{\partial x_1}{\partial \xi_1}\frac{\partial V_2}{\partial \xi_2}\right), \\
[\bar{A}_3]_{32} &= \frac{1}{s}\left(\frac{\partial x_2}{\partial \xi_2}\frac{\partial V_2}{\partial \xi_1} - \frac{\partial x_2}{\partial \xi_1}\frac{\partial V_2}{\partial \xi_2}\right), \\
[\bar{A}_3]_{33} &= \frac{1}{s}\left(-\frac{\partial x_1}{\partial \xi_2}\frac{\partial V_2}{\partial \xi_1} + \frac{\partial x_1}{\partial \xi_1}\frac{\partial V_2}{\partial \xi_2}\right).
\end{aligned}$$

Finally,

$$\bar{F} = \frac{1}{s}\begin{pmatrix}
c\left(\frac{\partial x_2}{\partial \xi_2}\frac{\partial V_1}{\partial \xi_1} - \frac{\partial x_2}{\partial \xi_1}\frac{\partial V_1}{\partial \xi_2} - \frac{\partial x_1}{\partial \xi_2}\frac{\partial V_2}{\partial \xi_1} + \frac{\partial x_1}{\partial \xi_1}\frac{\partial V_2}{\partial \xi_2}\right) \\
V_1\left(\frac{\partial x_2}{\partial \xi_2}\frac{\partial V_1}{\partial \xi_1} - \frac{\partial x_2}{\partial \xi_1}\frac{\partial V_1}{\partial \xi_2}\right) + V_2\left(-\frac{\partial x_1}{\partial \xi_2}\frac{\partial V_1}{\partial \xi_1} + \frac{\partial x_1}{\partial \xi_1}\frac{\partial V_1}{\partial \xi_2}\right) \\
V_1\left(\frac{\partial x_2}{\partial \xi_2}\frac{\partial V_2}{\partial \xi_1} - \frac{\partial x_2}{\partial \xi_1}\frac{\partial V_2}{\partial \xi_2}\right) + V_2\left(-\frac{\partial x_1}{\partial \xi_2}\frac{\partial V_2}{\partial \xi_1} + \frac{\partial x_1}{\partial \xi_1}\frac{\partial V_2}{\partial \xi_2}\right)
\end{pmatrix}.$$

Here,

$$s = \frac{\partial x_1}{\partial \xi_1}\frac{\partial x_2}{\partial \xi_2} - \frac{\partial x_1}{\partial \xi_2}\frac{\partial x_2}{\partial \xi_1}.$$

Note that the transformed equations have variable coefficients even if the approximate solution W is constant.

The approximation W is determined by solving a simplified flow problem. By assuming constant density and rotational free flow, the stationary Euler equations transform into a potential flow problem

$$\begin{aligned}
&\nabla^2\phi = 0, \\
&v_1^2 + v_2^2 \text{ constant},
\end{aligned}$$

where

$$v_1 = -\frac{\partial \phi}{\partial x_1}$$

and

$$v_2 = -\frac{\partial \phi}{\partial x_2}.$$

The boundary conditions are chosen to conserve momentum. For low Mach numbers, the solution to the potential flow problem is a good approximation of the solution to the Euler equations.

In summary, the discretization procedure for the linearized Euler equations is given by:

1. A software package [1] is used to define an orthogonal mapping from the physical to the computational domain, see Figure 6.1.
2. The mapping is used to transform the linearized Euler equations.
3. The transformed equations are discretized using an $m_1 \times m_2$ uniform grid in the computational domain. The derivatives are approximated using second order centered finite differences. We also add a fourth order dissipation of the same type as for the model problem (5.1).
4. The boundary conditions (6.3) are utilized. Outgoing characteristic variables at left and right boundaries are extrapolated, using linear extrapolation. At walls, the velocity parallel to the wall as well as the density is extrapolated. These boundary conditions are used to eliminate the equations corresponding to boundary points.

Thanks to the orthogonal mapping, the discretization procedure becomes rather simple. The result is a system of equations of the type (1.4) with $3m_1m_2$ unknowns. The structure of the coefficient matrix can be seen in Figure 3.2. After solving the transformed problem, the solution can easily be transformed back to the physical domain.

The potential flow problem is treated in the same way as above, that is, the problem is transformed to the computational domain and discretized using a centered finite difference approximation. After solving the problem, the solution is transformed back to the physical domain and utilized in building the matrix B and the right hand side g.

Once the problem has been set up, it is solved using the SC-RK1 and MG-RK3 methods. For the SC-RK1 iteration, we use a constant step $k = 0.5$ in pseudo-time for the straight channel setting. For the narrowing channel setting, the time step has to be decreased slightly, and $k = 0.4$ is used. Tor the MG-RK3 method, the time step is determined by the CFL-condition

$$k = \frac{\text{CFL}}{||V_1||_\infty/\tilde{h}_1 + ||V_2||_\infty/\tilde{h}_2 + c\sqrt{\tilde{h}_1^{-2} + \tilde{h}_2^{-2}}},$$

where CFL $= 1.8$, and $\tilde{h}_1$ and $\tilde{h}_2$ are the minimal space steps in each direction. As initial guess we use a Gaussian pulse in variables $\tilde{\rho}$ and $\tilde{v_1}$ and iterate until the norm of the residual has been decreased by a factor of 10^5.

A grid refinement study shows grid independent convergence for both the SC-RK1 and the MG-RK3 methods. In Tables 6.1 and 6.2, the number of iterations required for convergence are shown. The corresponding arithmetic complexities are displayed in Figures 6.2 and 6.3. We see that the SC-RK1 method requires less computational work for all problem sizes. The coarsest grid used in the multigrid method has $m_{1,2} = 8$ for all problems.

TABLE 6.1
Grid refinement study for the straight channel setting.

$m_{1,2}$	8	16	32	64	128
#MG-RK3 iterations	411	181	240	240	178
#SC-RK1 iterations	34	35	38	38	38

TABLE 6.2
Grid refinement study for the narrowing channel setting.

$m_{1,2}$	8	16	32	64	128
#MG-RK3 iterations	390	169	183	155	116
#SC-RK1 iterations	40	42	46	48	49

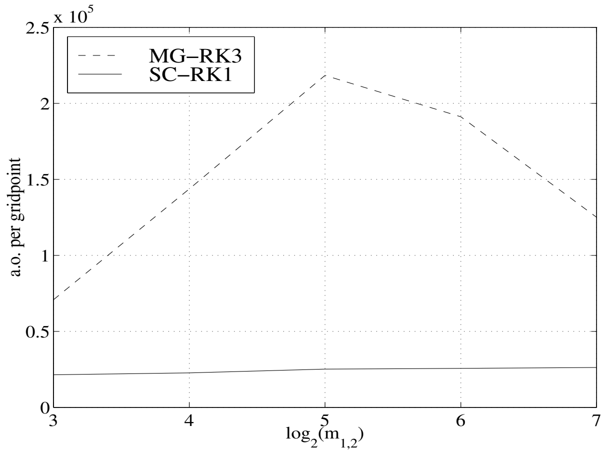

FIG. 6.2. *The number of arithmetic operations required for the MG-RK3 and SC-RK1 methods in the straight channel setting.*

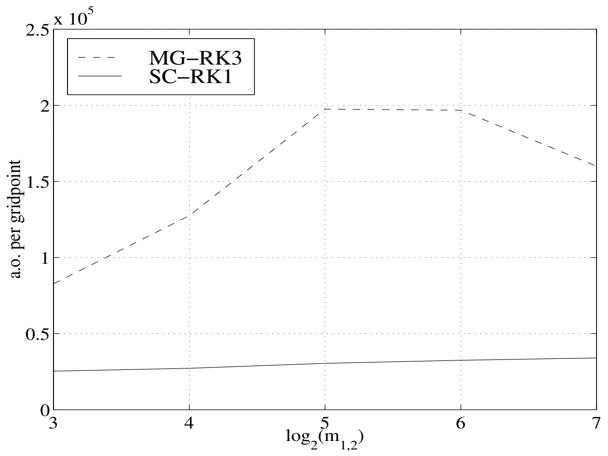

FIG. 6.3. *The number of arithmetic operations required for the MG-RK3 and SC-RK1 methods in the narrowing channel setting.*

In Figure 6.4 we see how the MG-RK3 method propagates the initial guess from left to right in the computational domain, while waves are reflected back and forth against the walls. As for the model problem, the SC-RK1 method has a completely different effect on the evolution of the solution. Figure 6.5 shows that the solution is effectively damped.

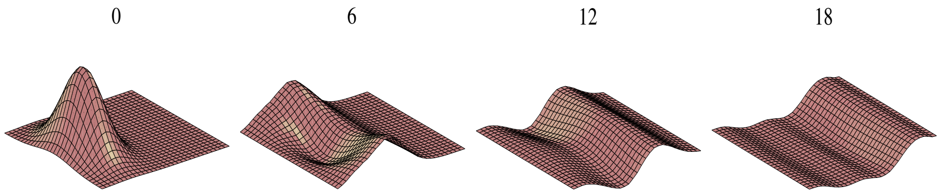

FIG. 6.4. *The density $\tilde{\rho}$ computed using the MG-RK3 method. The pictures show the state after 0, 6, 12 and 18 two-grid iterations respectively. Here, $m_1 = m_2 = 32$.*

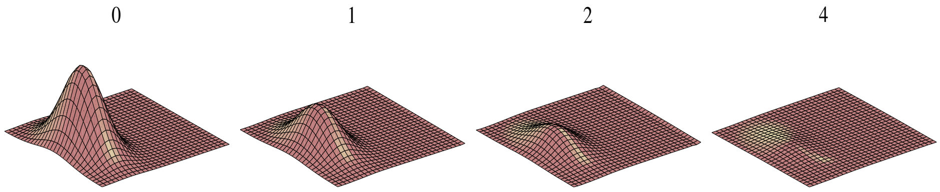

FIG. 6.5. *The density $\tilde{\rho}$ computed using the SC-RK1 method. The pictures show the state after 0, 1, 2 and 4 iterations respectively. Here, $m_1 = m_2 = 32$.*

Finally, we investigate the convergence properties when different amounts of artificial viscosity is added to the difference approximation. These experiments were performed for the straight channel setting. In Table 6.3, the number of iterations and the arithmetic complexity for the MG-RK3 method are shown, and the corresponding results for the SC-RK1 iteration are listed in Table 6.4. In Figure 6.6, the arithmetic complexities for the MG-RK3 and SC-RK1 methods are compared.

TABLE 6.3
Convergence study for MG-RK3. Here, $m_1 = 64$ and $m_2 = 32$.

γ_4	1/8	1/16	1/32	1/64	1/128	1/256	1/512	1/1024
Optimal #grid levels	3	3	3	3	3	3	2	2
CFL	1.1	1.5	1.8	1.8	1.8	1.8	1.7	1.7
#iterations	179	131	106	164	210	407	493	692
#a.o. per grid point [10^3]	167.9	122.8	99.4	153.8	197.0	381.8	354.1	497.1

TABLE 6.4
Convergence study for SC-RK1. Here, $m_1 = 64$ and $m_2 = 32$.

γ_4	1/8	1/16	1/32	1/64	1/128	1/256	1/512	1/1024
#iterations	44	33	38	49	65	82	98	97
#a.o. per grid point [10^3]	29.8	22.3	25.7	33.2	44.1	55.6	66.4	65.8

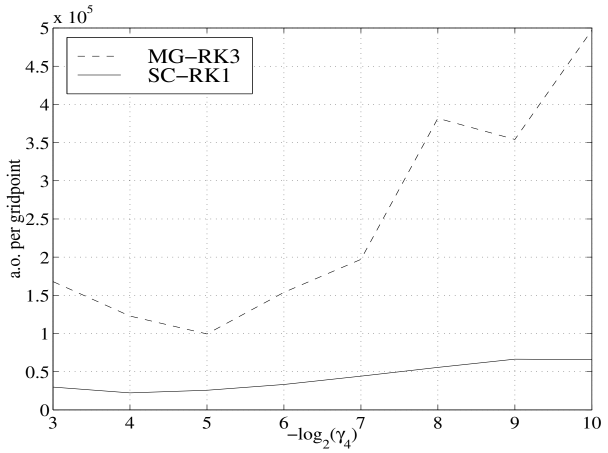

FIG. 6.6. *The number of arithmetic operations required for the MG-RK3 and SC-RK1 methods in the straight channel setting.*

From Figure 6.6, we see that the SC-RK1 method is more robust than the MG-RK3 iteration when the amount of artificial viscosity is changed. This result is consistent with the experiments performed for the model problem in §5.

7. Generalizations. The generalization of the semicirculant preconditioning technique to three space dimensions is straightforward, see, e.g., [23]. Here, the boundary conditions are altered to periodic in two dimensions, and the coefficients in the difference approximation are averaged over these directions.

If a system of time-independent, nonlinear PDE, e.g., the Euler equations (6.1), are discretized, the system (1.4) is nonlinear. As for the linear problems discussed in §1, such systems of equations are often solved by advancing the corresponding time-dependent problem (1.5) in time until steady-state is reached.

The most straight-forward way of employing semicirculant preconditioning for nonlinear problems is to replace $r\left(v^{(j)}\right)$ in the Runge-Kutta iteration (2.1) by $\left(M(v^{(j)})\right)^{-1} r\left(v^{(j)}\right)$. Alternatively, semicirculant approximations may be utilized for preconditioning the linear systems of equations arising if the Newton method, or a modified Newton method employing an augmented Jacobian [18] [15], is applied to (1.4). The linear systems may be approximately solved by performing one or a few iterations of an s-stage Runge-Kutta method, or another Krylov subspace method, e.g., GMRES(s). By utilizing Taylor expansions in (2.1), it may be shown that the two approaches are closely related. Employing semicirculant preconditioners for systems of nonlinear PDE is the field of current research.

A potential problem when implementing semicirculant preconditioning in existing CFD codes is that averages of the entries in the difference operator or the Jacobian over the x_1-direction are required for computing the semicirculant approximation M. In the codes, sometimes only a routine for computing the residual is supplied, and the matrix entries are not explicitly available. However, it is normally possible to derive analytic formulas for the entries of a *modified* matrix, where the boundary conditions are altered to periodic in one, two or three spatial directions, cf. §3.

8. Conclusions. In this paper, the iterative solution of time-independent, hyperbolic systems of PDE is considered. The PDE are discretized by a finite difference or finite volume scheme on a structured grid, and the systems of equations are solved by integrating the corresponding time-dependent problem until steady-state is reached. The integration is performed by utilizing an explicit time-marching method, combined with a preconditioner. The main result is that preconditioners based on a semicirculant approximation of the spatial difference operator may be efficiently employed in this context.

A relevant scalar model problem in two space dimensions is derived, for which the spectrum of the preconditioned coefficient matrix is analyzed. Asymptotically, as the number of grid points approaches infinity, the eigenvalues reside on two curve segments in the complex plane. The convergence properties for the preconditioned one-stage forward Euler scheme (SC-RK1) is studied, and it is shown that the number of iterations is bounded as the number of grid points grows, i.e., the SC-RK1 method yields grid independent convergence. Also, if more grid points are employed in the x_1-direction than in the x_2-direction, the number of iterations required is bounded as the artificial viscosity parameter is decreased. Numerical experiments show that, when the SC-RK1 iteration is employed, convergence is essentially achieved by damping of the error, the wave propagation property of the PDE is removed by the preconditioner. This property is compared to a standard multigrid method (MG-RK3), where improved convergence is achieved essentially by increasing the wave propagation speed. Finally, for the problems studied here, utilizing the SC-RK1 iteration results in lower arithmetic complexities than if the MG-RK3 method is employed. Furthermore, the SC-RK1 iteration is less sensitive to changes in the artificial viscosity.

Also, the Euler equations for flow in a narrowing channel are linearized around an approximate solution, yielding a linear system of PDE with variable coefficients. This system has properties which are very similar to the problems arising in the final phase of a solution process where the full non-linear PDE is solved. The SC-RK1 method is applied to the linearized problem, and the results show that the favorable convergence properties hold also for this, more realistic, problem setting. Again, the SC-RK1 iteration is faster than the standard multigrid method MG-RK3.

For the problems studied in this paper, it is clear that employing semicirculant approximations for preconditioning yields an efficient solution method. Further experiments for the full non-linear equations and for more complex geometries are required to determine if this also holds for realistic application settings.

9. Appendix A. Proof of Lemma 5.1.

Proof. The proof uses the same technique as in the proof of Lemma 6.1 in [22], where the result is established for the case $\gamma_2 = \gamma_4 = 0$.

We begin by proving that $\lambda_k = (1-\gamma_2)\xi_k + 2\gamma_2 - (1+\gamma_2)\xi_k^{-1}$, $k = 1, \ldots, m_d$, where ξ_k are solutions to

$$\xi^{2m_d}(\xi - 1)^2 + (-\delta)^{m_d}(\xi/\sqrt{\delta} + \sqrt{\delta})^2 = 0 \;, \quad \xi \neq \pm i\sqrt{\delta}.$$

Here, $\delta \equiv (1+\gamma_2)/(1-\gamma_2)$.

Let $v = (v_1 \cdots v_{m_d})^T$ and λ denote an eigenvector and eigenvalue of B_{m_d}. The eigenvalue formula $B_{m_d}v - \lambda v = 0$ yields the difference equation

$$(1-\gamma_2)v_{j+1} + (2\gamma_2 - \lambda)v_j - (1+\gamma_2)v_{j-1} = 0, \qquad j = 1, \ldots, m_d - 1,$$

with boundary conditions

$$\begin{cases} v_0 = 0, \\ (2-\lambda)v_{m_d} - 2v_{m_d-1} = 0. \end{cases}$$

The characteristic equation is obtained by making the ansatz $v_j = \nu^j$

$$0 = (1-\gamma_2)\nu^2 + (2\gamma_2 - \lambda)\nu - (1+\gamma_2) \tag{9.1}$$

or

$$\lambda = (1-\gamma_2)\nu + 2\gamma_2 - (1+\gamma_2)\nu^{-1}. \tag{9.2}$$

Since the roots ν_1, ν_2 to (9.1) satisfy $\nu_1\nu_2 = -\delta$, both roots yield the same eigenvalue λ. By contradiction, it is easily proved that there are no double roots $\nu_1 = \nu_2 = \pm i\sqrt{\delta}$. Hence, we may assume that $\nu_1 = \xi$, $\nu_2 = -\delta\xi^{-1}$, where $\xi \neq \pm i\sqrt{\delta}$, and the solution to the difference equation is given by $v_j = c_1\xi^j + c_2\delta^j(-\xi)^{-j}$. The boundary conditions yield

$$\begin{bmatrix} 1 & 1 \\ (2\xi - \xi^2 - 1)\xi^{2m_d} & (2\xi + \xi^2/\delta + \delta)(-\delta)^{m_d} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = 0,$$

which has a nontrivial solution if and only if the determinant is zero, i.e.,

$$P(\xi) \equiv \xi^{2m_d}(\xi - 1)^2 + (-\delta)^{m_d}(\xi/\sqrt{\delta} + \sqrt{\delta})^2 = 0. \tag{9.3}$$

The polynomial equation (9.3) has $2m_d + 2$ roots. The roots $\xi = \pm i\sqrt{\delta}$ must be excluded, since they are double roots to the characteristic equation. The remaining $2m_d$ roots can be labeled so that $(1-\gamma_2)\xi_k + 2\gamma_2 - (1+\gamma_2)\xi_k^{-1} = (1-\gamma_2)\xi_{k+m_d} + 2\gamma_2 - (1+\gamma_2)\xi_{k+m_d}^{-1}$. Thus, the eigenvalues to B_{m_d} are given by $\lambda_k = (1-\gamma_2)\xi_k + 2\gamma_2 - (1+\gamma_2)\xi_k^{-1}$, $k = 1, \ldots, m_d$.

Noting that $\xi = 1$ is not a root to (9.3), and using the polar form $\xi = re^{i\theta}$, we see that necessary conditions for the roots are

$$Q_1 = Q_2, \quad \xi \neq \pm i\sqrt{\delta},$$

where

$$Q_1 = |\xi|^{2m_d} = r^{2m_d},$$
$$Q_2 = \delta^{m_d}\frac{r^2/\delta + 2r\cos(\theta) + \delta}{r^2 - 2r\cos(\theta) + 1}$$

For the case $0 \le r < \sqrt{\delta}$, $-\frac{\pi}{2} \le \theta \le \frac{\pi}{2}$, we have that $Q_1 < \delta^{m_d}$ and $Q_2 \ge \delta^{m_d}$. Also, $r > \sqrt{\delta}$, $\frac{\pi}{2} \le \theta \le \frac{3\pi}{2}$, implies that $Q_1 > \delta^{m_d}$ and $Q_2 \le \delta^{m_d}$. Hence, the roots must lie in the regions defined by

$$D_1: \quad \sqrt{\delta} \le |\xi|, \quad -\frac{\pi}{2} \le \arg(\xi) \le \frac{\pi}{2}, \quad \xi \neq \pm i,$$
$$D_2: 0 < |\xi| \le \sqrt{\delta}, \quad \frac{\pi}{2} \le \arg(\xi) \le \frac{3\pi}{2}, \quad \xi \neq \pm i.$$

An investigation of the derivative of (9.3) yields that the equation

$$\frac{dP}{d\xi} = 0$$

does not have any roots in $D_1 \cup D_2$. Thus, there are no multiple roots to (9.3) in this region. Also noting that $P(\pm i\sqrt{\delta}) = 0$, we conclude that $P(\xi) = 0$, $\xi \neq \pm i\sqrt{\delta}$, has $2m_d$ single roots in $D_1 \cup D_2$.

(i): From the fact that $P(-\delta\xi^{-1}) = (-1)^{m_d}\delta^{m_d+1}\xi^{-2m_d-2}P(\xi)$ it follows that if $P(\xi) = 0$ then $P(-\delta\xi^{-1}) = 0$. We conclude that there must be m_d roots in both

D_1 and D_2. Thus, the roots can be divided into two separate sets $\{\xi_{k,1}\}_{k=1}^{m_d} \in D_1$ and $\{\xi_{k,2}\}_{k=1}^{m_d} \in D_2$, where $\xi_{k,1} \equiv \xi_k$, $\xi_{k,2} = -\delta\xi_{k,1}^{-1} = -\delta\xi_k^{-1}$. The roots in D_1 yield the same eigenvalues as those in D_2. By calculating the eigenvalues $\lambda_k = -(1+\gamma_2)\xi_k^{-1} + 2\gamma_2 + (1-\gamma_2)\xi_k$, where $P(\xi_k) = 0$ and $\{\xi_k\}_{k=1}^{m_d} \in D_1$, we obtain

$$\begin{aligned}\lambda_k - \lambda_j &= (1-\gamma_2)(\xi_k - \xi_j) - (1+\gamma_2)(\xi_k^{-1} - \xi_j^{-1}) \\ &= (1-\gamma_2)(\xi_k - \xi_j)\big(\xi_k - (-\delta\xi_j^{-1})\big)\xi_k^{-1} \\ &\neq 0, \quad \text{for } \ k \neq j,\end{aligned}$$

since $\xi_k \neq \xi_j$ and $D_1 \ni \xi_k \neq -\delta\xi_j^{-1} \in D_2$. Hence, the eigenvalues λ_k are distinct.

(ii): Assume that $\xi \in D_1$. This implies that $\Re e\left((1-\gamma_2)\xi - (1+\gamma_2)\xi^{-1}\right) \geq (1-\gamma_2)\sqrt{\delta} - (1+\gamma_2)(\sqrt{\delta})^{-1} = 0$. Hence, $\Re e(\lambda_k) \geq 2\gamma_2$.

(iii): From the condition $Q_1 = Q_2$, it is clear that, as $m_d \to \infty$, the roots $\xi_k \in D_1$ approach the circle $\sqrt{\delta}e^{i\theta}$, $-\pi/2 < \theta < \pi/2$. Hence, λ_k approaches the line $2\gamma_2 + (1-\gamma_2)\sqrt{\delta}e^{i\theta} - (1+\gamma_2)\sqrt{\delta}e^{-i\theta} = 2\gamma_2 + 2i\sqrt{1-\gamma_2^2}\sin\theta$.
$\square$

REFERENCES

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

[2] R. Enander, *Improved implicit residual smoothing for computations of steady state solutions of first order hyperbolic systems*, J. Comput. Phys., 107:2 (1993), pp. 291–296.

[3] L. Ferm, *The number of coarse-grid iterations every cycle for the two-grid method*, SIAM J. Sci. Comput., 19:2 (1998), pp. 493–501.

[4] B. Gustafsson and P. Lötstedt, *Analysis of the multigrid method applied to first order systems*, in Fourth Copper Mountain Conf. Multigrid Methods, SIAM, Philadelphia, PA, 1989, pp. 181–233.

[5] W. Hackbusch, *Multi-Grid Methods and Applications*, Springer-Verlag, Berlin, 1985.

[6] L. Hemmingsson, *A semi-circulant preconditioner for the convection-diffusion equation*, Numer. Math., 81 (1998), pp. 211–248.

[7] S. Holmgren, *CG-like iterative methods and semicirculant preconditioners on vector and parallel computers*, Report 148, Dept. of Scientific Computing, Uppsala Univ., Uppsala, Sweden, 1992.

[8] S. Holmgren, H. Brandén, and E. Sterner, *Convergence acceleration for the Navier–Stokes equations using optimal semicirculant approximations*, Accepted for publication in SIAM J. Sci. Comput.

[9] S. Holmgren and K. Otto, *A comparison of preconditioned iterative methods for nonsymmetric block-tridiagonal systems of equations*, Report 123 (revised), Dept. of Scientific Computing, Uppsala Univ., Uppsala, Sweden, 1990.

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

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

[12] ———, *Semicirculant solvers and boundary corrections for first-order partial differential equations*, SIAM J. Sci. Comput., 17 (1996), pp. 613–630.

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

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

[15] E. Issman, G. Degrez, and H. Deconinck, *Implicit upwind residual-distribution Euler and Navier–Stokes solver on unstructured meshes*, AIAA J., 34 (1996), pp. 2021–2028.

[16] A. Jameson, *Computational transonics*, Comm. Pure Appl. Math., 41 (1988), pp. 507–549.

[17] 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).

[18] C. T. Kelly and D. E. Keyes, *Convergence analysis of pseudo-transient continuation*, SIAM J. Numer. Anal., 35 (1998), pp. 508–523.

[19] P. Lötstedt, *Grid independent convergence of the multigrid method for first-order systems*, SIAM J. Numer. Anal., 29:5 (1992), pp. 1370–1394.

[20] ———, *Improved convergence to the steady state of the Euler equations by enhanced wave propagation*, J. Comput. Phys., 114:1 (1994), pp. 34–44.

[21] P. Lötstedt and B. Gustafsson, *Fourier analysis of multigrid methods for general systems of PDEs*, Math. Comp., 60:202 (1993), pp. 473–493.

[22] K. Otto, *Analysis of preconditioners for hyperbolic partial differential equations*, SIAM J. Numer. Anal., 33 (1996), pp. 2131–2165.

[23] ———, *A unifying framework for preconditioners based on fast transforms*, Report 187, Dept. of Scientific Computing, Uppsala Univ., Uppsala, Sweden, 1996.

[24] E. Turkel, *Review of preconditioning methods for fluid dynamics*, Appl. Num. Math., 12 (1993), pp. 257–284.