

MULTIGRID SOLVERS FOR NONALIGNED SONIC FLOWS *

ACHI BRANDT[†] AND BORIS DISKIN[†]

Abstract. We investigate an approach to the solution of nonelliptic equations on a rectangular grid. The multigrid algorithms presented here demonstrate the “textbook multigrid efficiency” even in the case that the equation characteristics do not align with the grid. To serve as a model problem, the two-dimensional (2D) and three-dimensional (3D) linearized sonic flow equations have been chosen. Efficient full-multigrid (FMG) solvers for the problems are demonstrated.

Key words. multigrid methods, textbook multigrid efficiency, sonic flow, nonalignment

AMS subject classifications. 65N55, 76H05, 76M20

PII. S1064827598332205

1. Introduction. The full-multigrid (FMG) algorithm is known to be the fastest solver for discretized elliptic partial differential equations (PDEs). General nonlinear elliptic systems can be solved in just a few *minimal work units*, a *minimal unit* being defined as the number of operations necessary for the *discretization* of the system on the target grid. However, attempts to extend the same techniques to nonelliptic equations, such as the transonic potential flow equation or its degenerate case—the sonic flow equation—have met with a more limited success. Although such multigrid methods are usually much more efficient than comparable single-grid methods, the “textbook” goal of solving in just a few minimal work units has not been attained. Indeed, many of the reported solvers require hundreds of minimal work units. Others, such as solvers based on various modifications of the ILU decomposition, being efficient in two dimensions, cannot be directly extended to three dimensions without losing most of their efficiency.

The increased amount of required work results from *several* different factors, and the first important step therefore is to separate these out. The present research addresses one basic difficulty in separation from others: the problem of nonalignment. The problem arises wherever characteristics of a differential equation do not coincide with grid lines. We study the nonalignment effect on our model problems: the two-dimensional (2D) and three-dimensional (3D) sonic-flow equations linearized over a given (variable) velocity field. These problems are reduced to solving degenerate elliptic equations which are elliptic on lower-dimensional manifolds embedded in the original space.

The usual goal of an FMG solver (see, e.g., [2, section 7] or [3, section 7]) is to reduce the target-grid (grid h) *algebraic error* $\|u^h - \tilde{u}^h\|$ below the level of the *discretization error* $\|u^h - U^h\|$, where u^h and \tilde{u}^h are the exact and computed solutions of the discretized equation, respectively; U^h is some target-grid representation of the true solution to the differential equation; and $\|\cdot\|$ is a given norm of interest. Sometimes the *total error* $\|\tilde{u}^h - U^h\|$ is also measured.

*Received by the editors January 5, 1998; accepted for publication (in revised form) May 20, 1998; published electronically September 29, 1999. The research reported here has been supported by grant NAG 1 1823 from NASA, by grant 9680-1-97 from the Israeli Ministry of Science, and by the Carl F. Gauss Minerva Center for Scientific Computation at the Weizmann Institute of Science.
<http://www.siam.org/journals/sisc/21-2/33220.html>

[†]Department of Applied Mathematics and Computer Science, The Weizmann Institute of Science, Rehovot 76100, Israel (achi@wisdom.weizmann.ac.il, diskinn@wisdom.weizmann.ac.il).

In many cases regular FMG algorithms are sufficient to yield a final solution with an algebraic error much smaller than the discretization error. In some problems, however, the usual algorithm cannot efficiently treat certain smooth components of the target-grid solution. The sonic flow problem, as well as other PDEs in which the discretization scheme introduces a numerical dissipation if the characteristic lines are not consistently aligned with the grid, falls into this category. The trouble has been previously observed and treated for convection-diffusion equations (see [1]) and for high-Reynolds incompressible entering flows (see [10]). In such cases the usual multigrid cycles lose their efficiency and so does the FMG algorithm employing the cycles.

A simple explanation can be given when the characteristics of the differential equation emanate from the boundary, in which case the quality of the coarse-grid correction is determined by how well certain cross-characteristic oscillations are advected from the inflow boundary into the domain. The main trouble is the increased numerical dissipation on coarse grids, which causes the decay and phase shift of these cross-characteristic oscillations to differ greatly from their values on the fine grid.

The idea suggested in this paper to overcome the trouble is to use *semicoarsening* together with the introduction of a well-balanced *explicit numerical dissipation* on coarse grids to control the penetration of the incoming cross-characteristic oscillations.

In section 2 we give a general formulation of the problem and introduce the main ideas. This is followed by detailed descriptions of the concrete problems and the solution algorithms in two dimensions (section 3 and section 5) and three dimensions (section 4). A variety of numerical experiments and the Fourier mode analysis (section 3.2.5) confirm the “textbook multigrid efficiency” of the suggested solvers. In section 6 we discuss an approach to applying the developed algorithms for solving more complicated problems, including the full-potential equation and the Euler system.

2. General description.

2.1. Differential problem. The full-potential flow equation has the quasi-linear form

$$(2.1) \quad (\bar{u} \cdot \nabla)^2 \Phi - a^2 \Delta \Phi = f,$$

where \bar{u} is the velocity vector, a is the speed of sound, and the unknown scalar function $\Phi(x, y)$ is the velocity potential of the irrotational flow. The operator appearing in (2.1) is one of the factors of the principal part determinant of the Euler system of equations for compressible flows (see [3, section 20]). Hence, according to rules developed in [3] (see also [2], [10], and [16]), the development of efficient multigrid solvers for the Euler system depends on devising such solvers for the principal part of (2.1), in which \bar{u} and a are *given* fields unrelated to Φ .

The type of this equation depends on the ratio $M = (\bar{u} \cdot \bar{u})^{1/2}/a$, which is called the Mach number. For $M < 1$, the equation is *elliptic*; for $M > 1$, it is *hyperbolic*. When $M = 1$, (2.1) degenerates to the *sonic flow equation*

$$(2.2) \quad \tilde{\Delta} \Phi = -f/a^2,$$

where $\tilde{\Delta}$ is the Laplacian on a manifold orthogonal to the velocity field. We call each such manifold a *characteristic manifold*.

We first study the phenomenon of nonalignment in the case of *constant fields* \bar{u} and a . This assumption implies that the characteristic manifold is a straight line in two dimensions (*characteristic line*) and a plane in three dimensions (*characteristic plane*). Although relatively simple, the problem nevertheless contains one of the main difficulties appearing in flows not consistently aligned with the discretization grid. The constant coefficient solver developed in section 3 will later be extended with some modifications to solve sonic flows linearized over *variable* velocity fields (section 5). Additional details related to the constant coefficient case, including a comprehensive half-space mode analysis, can be found in [9], [11], and [13].

The approach we follow is to use a fixed Cartesian coordinate system independent of the characteristic manifold. We call the x - y plane in three dimensions (or the x -axis in two dimensions) *the reference plane (axis)*. Thus each characteristic manifold is uniquely defined by a point on it and its slopes relative to the coordinate axes of the reference plane. To simplify the description of the algorithm in three dimensions we will use the term “inclination to the x -axis” to denote the angle between the x -coordinate axis and the intersection of the considered characteristic plane with the x - z plane. The inclination to the y -axis is similarly defined. We restrict ourselves to characteristic planes whose inclinations are less than or equal to $\frac{\pi}{4}$ (45°), in other words, to planes that can be written as $\alpha x + \beta y + \gamma z = \delta$, where $|\gamma| \leq |\alpha|$ and $|\gamma| \leq |\beta|$. Otherwise the role of the axes should be interchanged. (See section 5 below concerning the case of variable coefficients.)

2.2. Principles of discretization. The target discretization grid we consider is usually a uniform (square) grid. The coarse grids used in the multigrid construction are rectangular grids with fixed (integer) aspect ratios. The aspect ratio $m = h_{\text{ref}}/h_{\text{vert}}$ is defined as the ratio of the meshsize in the reference plane (axis) to the meshsize in the vertical axis. In three dimensions, grids in the reference plane are always assumed to be uniform.

Let us consider a discretization of the differential operator (2.2) at a given grid point. The characteristic manifold going through the point does not generally contain other grid points. In order to discretize such an operator we introduce ghost points located at the intersections of the characteristic manifold with the adjacent vertical grid lines. The function value at a ghost point is interpolated from its genuine grid neighbors placed on the same vertical grid line. For small aspect ratios this interpolation degrades the discrete operator approximation order and to compensate for this degradation a vertical operator should be added. Thus, the discretization involves the following steps.

Step 1. The target degenerate elliptic operator is first discretized on the characteristic manifold, using the ghost points, employing an *h -elliptic discretization* (see [2, section 2.1] or [3, section 2.1]). We call this lower-dimensional discrete operator the *low-dimensional prototype*.

Step 2. The full-dimension discretization is obtained from the low-dimensional prototype by the vertical linear interpolation to the ghost points, together with the addition of several compensating points on the vertical grid line going through the point where the discrete operator is being defined. This discretization is *h -elliptic* in the full dimension.

Such a discretization possesses a numerical dissipation (because of the nonalignment and the resulting interpolation) that appears to be relatively small, especially on grids with a high aspect ratio. Nevertheless, this dissipation exists on each grid. We will call it the *inherent numerical dissipation* to distinguish it from the “explicit

numerical dissipation” introduced below. Quantitatively, the inherent numerical dissipation is defined as the coefficient of the lowest pure cross-characteristic derivative arising in the *first differential approximation* (FDA) to the discrete operator (see [15]), the cross-characteristic direction at a point being defined as the direction perpendicular to the characteristic manifold. In our model problems the cross-characteristic direction coincides with the velocity direction.

Let us now introduce some useful terms. A function defined on the space under consideration will be referred to as a *characteristic component* if it is a smooth function on the characteristic manifold. The terms *high-frequency characteristic component* and *smooth characteristic component* will refer to components that are highly oscillating and smooth in the *cross-characteristic* direction (but not as smooth as in any of the characteristic directions).

Previous studies on several types of nonelliptic equations (see [1] and [10]) have shown that the basic trouble in constructing an efficient multigrid solver is the poor approximation of smooth characteristic components on coarse grids. The reason is the increased coarse-grid inherent numerical dissipation appearing in cycles with full coarsening, i.e., when the coarse grid has all the meshsizes twice as large as those of the fine grid. A general way to overcome this trouble would be to use *semicoarsening*, with meshsizes being doubled only in the reference plane. When applied in its pure form, semicoarsening also results in some difficulties, since the inherent numerical dissipation of the semicoarsened grid will be much less than that of the fine grid. However, we can supply the operator on the semicoarsened grid with an additional term (explicit numerical dissipation term) so that the total dissipation on the semicoarsened grid would be the same as on the fine grid.

The three- or even four-level version of such a cycle, with two pointwise relaxation sweeps on each level and appropriate intergrid transfers, can already be used to solve efficiently the model problems discretized on a uniform target grid. However, the implementation of a cycle with more levels raises the following new difficulty.

2.3. Strong cross-characteristic coupling. The inherent numerical dissipation in our discretizations arises from the vertical interpolation to the ghost points. To obtain the same total dissipation, we introduce an explicit numerical dissipation on the coarse grids by adding a term which is a discrete approximation to a vertical derivative of a suitable order.

The multigrid theory of h -elliptic discrete operators (see [1], [3]) shows that a pointwise relaxation can reduce only the error components that oscillate in the strong-coupling directions. The coupling analysis of the discretizations considered in sections 3.2.3 and 4.4 below shows that the *target-grid* discrete-operator directions of stronger coupling approximately coincide with the characteristic manifold. Thus a target-grid pointwise relaxation can reduce efficiently the noncharacteristic error components and also some of the high-frequency characteristic components of the error. That is all we need from the relaxation since the smooth characteristic components (and most of the high-frequency characteristic components) are well reduced on the next *semicoarsened* grids. However, successive semicoarsening implies a fast decrease in the *inherent numerical dissipation* on the coarse grids and hence a fast increase in the weight of the compensating explicit numerical dissipation in the coarse-grid discrete operator. Thus, the direction of the strongest coupling after several semicoarsening steps tends to be vertical; hence, any pointwise relaxation on such coarse grids cannot efficiently reduce some *noncharacteristic* components of the error. A way to eliminate this degradation is to use a vertical line relaxation, in which all the points

located on the same vertical grid line are relaxed simultaneously. Thus our general strategy is to use a pointwise relaxation on several of the finest levels and switch to a line relaxation on all the coarser levels. Since

- a) the line solver is just a solver to a five-diagonal matrix;
 - b) line relaxation is used only starting from some coarse grid;
 - c) it remains a line relaxation (rather than a plane relaxation) in three dimensions,
- the total amount of work will be practically the same as for a solver using only a pointwise relaxation.

Instead of switching to a line relaxation on some of the coarser grids, one can avoid creating strong cross-characteristic coupling altogether by replacing part of the semicoarsening steps by *full* coarsening steps (see [11], [12]). This *conditional coarsening* is slightly cheaper in computing time but considerably more complicated to program, especially in extensions to variable coefficients.

3. 2D sonic flow: The constant coefficient case.

3.1. Model problem statement. In two dimensions, (2.2) turns into the simple equation

$$(3.1) \quad \frac{\partial^2 \Phi}{\partial \xi^2} = F,$$

where the differentiation direction ξ coincides with the characteristic direction (orthogonal to the velocity direction).

In our model problem an unknown scalar function $\Phi(x, y)$ is defined on the square $(x, y) \in [0, 1] \times [0, 1]$, and the variable along the characteristic direction is $\xi = (x + ty)(1 + t^2)^{-1/2}$, where $t = \tan \psi$ is the tangent of the *angle of nonalignment*, i.e., the angle between the characteristic direction and the reference x -axis. Usually $|t| \leq 1$; otherwise one can improve the discretization described below by switching the reference axis.

In our model problem we supply (3.1) with Dirichlet boundary conditions in the x direction and periodic conditions in the y direction:

$$(3.2) \quad \Phi(0, y) = g_0(y), \quad \Phi(1, y) = g_1(y), \quad \Phi(x, y) = \Phi(x, y + 1),$$

where $g_0(y)$ and $g_1(y)$ are given functions. The choice of periodic boundary conditions is quite useful for this study, as it precludes boundary layers, which could obscure the phenomena we would like to examine. It also facilitates implementation of any angle of nonalignment.

Let us introduce an auxiliary Cartesian coordinate system (ξ, η) , where ξ is the variable along the characteristic direction defined above and $\eta = (-tx + y)(1 + t^2)^{-1/2}$ is the variable along the cross-characteristic direction.

Consider the nine-point discretization of (3.1) on a grid with aspect ratio $m = h_x/h_y$, where h_x and h_y are the meshsizes in the x and y directions, respectively. For integers (i_1, i_2) the discrete approximation to $\Phi(i_1 h_x, i_2 h_y)$ is denoted by ϕ_{i_1, i_2} , and the discrete approximation to the differential operator in (3.1) is defined by

$$(3.3) \quad L^{(h_x, h_y)} \phi_{i_1, i_2} \equiv \frac{1}{h_x^2 + (k + s)^2 h_y^2} [(1 - s) (\phi_{i_1 - 1, i_2 - k} + \phi_{i_1 + 1, i_2 + k}) \\ + s (\phi_{i_1 - 1, i_2 - (k+1)} + \phi_{i_1 + 1, i_2 + (k+1)}) \\ - 2\phi_{i_1, i_2} - s(1 - s) (\phi_{i_1, i_2 - 1} - 2\phi_{i_1, i_2} + \phi_{i_1, i_2 + 1})] \\ - A \frac{1}{h_y^2} [\phi_{i_1, i_2 + 2} - 4\phi_{i_1, i_2 + 1} + 6\phi_{i_1, i_2} - 4\phi_{i_1, i_2 - 1} + \phi_{i_1, i_2 - 2}],$$

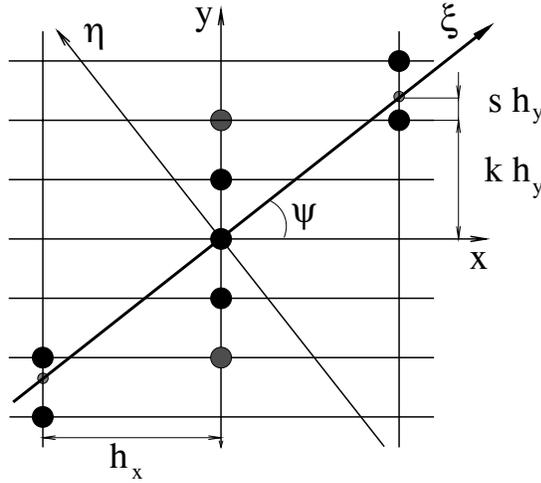


FIG. 3.1. Anisotropic grid; nine-point stencil.

where $k + s = mt$, k is an integer, and $0 \leq s < 1$ (see Figure 3.1); A is the *explicit* numerical dissipation coefficient. Thus the differential problem (3.1)–(3.2) is discretized on the grid as

$$\begin{aligned}
 L^{(h_x, h_y)} \phi_{i_1, i_2} &= f_{i_1, i_2}, \quad i_1 = 1, \dots, n_1 - 1, \\
 \phi_{0, i_2} &= g_0(i_2 h_y), \\
 \phi_{n_1, i_2} &= g_1(i_2 h_y), \\
 \phi_{i_1, i_2 + n_2} &= \phi_{i_1, i_2}, \quad i_1 = 0, 1, \dots, n_1,
 \end{aligned}
 \tag{3.4}$$

where $n_1 = 1/h_x$, $i_2 \in \mathbf{Z}$, $n_2 = 1/h_y$, and $f_{i_1, i_2} = F(i_1 h_x, i_2 h_y)$.

The low-dimensional prototype to (3.4) is a simple one-dimensional (1D) Dirichlet problem for the second derivative operator. It is well known that one V -cycle with two red-black relaxation sweeps per level, the full-weighting residual transfer, and the linear interpolation of the coarse-grid correction exactly solves the 1D-prototype problem. We mimic this cycle in the full 2D solver.

The first differential approximation (cf. [15], [1]) to the operator (3.3) is

$$\phi_{\xi\xi}^h - h_y^2 \left[A + \left(\frac{(1-s)s \cos(\psi)}{2m} \right)^2 \right] \phi_{yyyy},
 \tag{3.5}$$

where $\phi_{\xi\xi}^h$ is the first differential approximation to the 1D prototype. For characteristic components $\phi_{\xi\xi}^h \approx \phi_{\xi\xi}$. On the target grid $A = 0$. On coarser grids we choose A so that the *total cross-characteristic dissipation*, i.e., the coefficient of ϕ_{yyyy} in (3.5), would remain the same as on the target grid. (Note that upon each semicoarsening step the values of h_y and ψ remain unchanged while $s(1-s)/m$ decreases. Note also that the true cross-characteristic dissipation should be defined as the coefficient of the fourth derivative with respect to η , but for characteristic components that is just proportional to the fourth derivative with respect to y .)

3.2. Multigrid cycles. In this section the basic parts of the multigrid cycles, such as relaxation, residual transfer, and correction interpolation are described. Numerical two-level tests together with the two-level Fourier mode analysis discussed here allow us to choose parameters for efficient multilevel cycles, which are then examined at the end of this section.

3.2.1. Relaxation schemes. Two types of relaxation are considered here: pointwise and “zebra.”

The elementary step of the *pointwise relaxation* is to change the solution approximation at the point (i_1, i_2) so as to satisfy (3.4). The order of performance of the elementary steps obeys the following rules:

1) Odd vertical lines (the vertical lines with odd i_1 coordinate) are relaxed before even ones.

2) The relaxation in each vertical line consists of four sweeps. Each sweep performs the elementary step for every fourth point in the line. The first sweep starts from the point with vertical coordinate $i_2 = 0$; the second, from the point with $i_2 = 2$; the third, from the point with $i_2 = 1$; and the last, from the point with $i_2 = 3$.

This eight-color order of relaxation is not necessary for efficient smoothing. It is chosen to enable full parallelization and precludes the appearance of relaxation “boundary layers.” To be sure, the usual red-black relaxation order would be efficient as well, but then the results would depend slightly on *where* the sweeps start and end, which we wanted to avoid.

The elementary step of the “zebra” *line relaxation* is to solve (or solve approximately) the system of all the discrete equations centered at the same vertical grid line. This step results in simultaneous replacement of the solution approximation at all the grid points belonging to that line. All the residuals on this line are thereby reduced to zero (or near zero). The order of lines remains as above: all the odd lines are relaxed before all the even ones (hence the name “zebra”).

3.2.2. Intergrid transfers. In any cycle there are two types of intergrid communication. The fine-to-coarse transfer (restriction) produces a coarse-grid approximation to the fine-grid residual function

$$r_{i_1, i_2} = f_{i_1, i_2} - L^{(h_x, h_y)} \phi_{i_1, i_2}.$$

The coarse-to-fine transfer (prolongation) is the coarse-grid correction interpolation. In the present algorithm both intergrid transfers are anisotropic. They roughly simulate the corresponding transfers of the 1D-prototype solver.

Residual transfer to the semicoarsened grid is given by

$$(3.6) \quad R_{i_1, i_2} = \left(I_h^H r \right)_{i_1, i_2} = .5 r_{2i_1, i_2} + .25 \left[(1-s) \left(r_{2i_1-1, i_2-k} + r_{2i_1+1, i_2+k} \right) + s \left(r_{2i_1-1, i_2-k-1} + r_{2i_1+1, i_2+k+1} \right) \right],$$

where R and r denote the coarse- and fine-grid residual functions, respectively.

Notice that the weighted average $(1-s)r_{2i_1-1, i_2-k} + sr_{2i_1-1, i_2-k-1}$ defines the residual value at the ghost point r_{2i_1-1, i_2-k-s} ; hence, (3.6) corresponds to the standard 1D full-weighting residual transfer.

The scheme is described in Figure 3.2. The solid lines show to where a fine-grid point residual is sent. The dashed arrows exhibit all the fine-grid points sending their residuals to a given coarse-grid point.

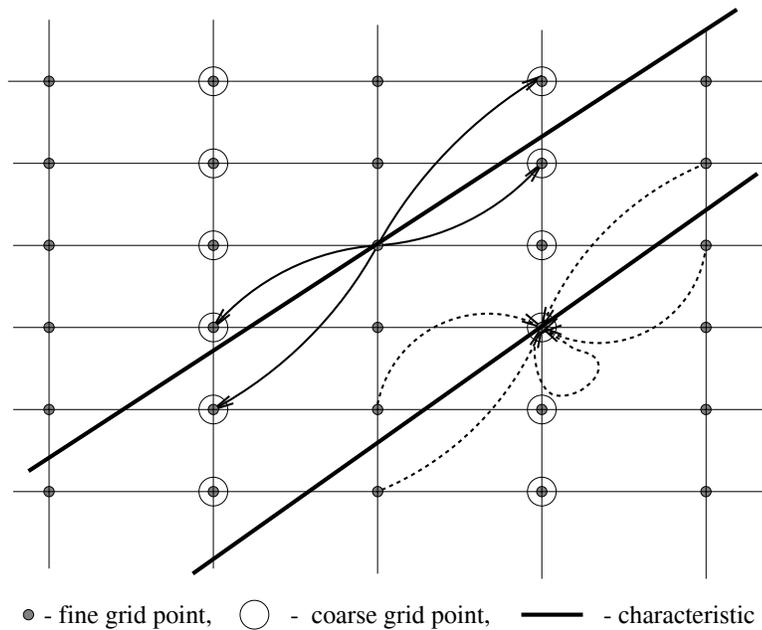


FIG. 3.2. Residual transfer to semicoarsened grid.

Interpolation of the coarse-grid correction simulates the 1D linear interpolation, giving the operation adjoint to (3.6):

$$(3.7) \quad \begin{cases} v_{2i_1, i_2} &= V_{i_1, i_2}, \\ v_{2i_1+1, i_2} &= .5 \left[(1 - S_1) (V_{i_1, i_2 - K_1} + V_{i_1+1, i_2 + K_1}) \right. \\ &\quad \left. + S_1 (V_{i_1, i_2 - K_1 - 1} + V_{i_1+1, i_2 + K_1 + 1}) \right], \end{cases}$$

where V denotes the solution of the coarse-grid problem, v denotes the correction to the fine-grid solution approximation, and K_1 is an integer such that $(K + S)/2 = K_1 + S_1$, $0 \leq S_1 < 1$, K and S being the parameters of the coarse-grid discretization (defined like k and s in Figure 3.1).

3.2.3. Switching criterion. The condition of switching from pointwise to zebra relaxation can be derived from the coupling analysis of the FDA approximation (3.5). The term of the FDA responsible for the “characteristic” coupling is $\phi_{\xi\xi}^h$, and a quantitative measure of this coupling is

$$h_{\xi}^{-2} = \left((m^2 + (k + s)^2) h_y^2 \right)^{-1}.$$

The “dissipative” coupling is maintained by the second term in (3.5), in which ϕ_{yyyy} has the coupling strength h_y^{-4} . We switch to the zebra scheme when this “dissipative” coupling becomes larger than the “characteristic” one, i.e., when the ratio between them, which we call the *relative coupling (RC)*, becomes larger than one. The derived criterion completely agrees with the experimental one obtained from the asymptotic convergence rates of two-level cycles.

3.2.4. Two-level cycles. A two-level cycle $V_2(\nu_1, \nu_2)$ can be defined as the following six steps:

Step 1. Prerelaxation sweeps. Improve the initial fine-grid approximation by ν_1 relaxation sweeps.

Step 2. Residual transfer. Build the coarse-grid approximation R to the fine-grid residual function $r_{i_1, i_2} = f_{i_1, i_2} - L^{(h_x, h_y)} \phi_{i_1, i_2}$, namely, calculate $R = I_h^H r$, where I_h^H is the fine-to-coarse residual transfer defined by (3.6).

Step 3. Coarse-grid equation. Form the coarse-grid equation

$$L^{(H_x, H_y)} V = R.$$

At this stage we choose the precise dose of coarse-grid explicit numerical dissipation A and recalculate the new values of the discretization parameters S and K depending on the coarse-grid aspect ratio.

Step 4. Exact solution. Solve the coarse-grid equation by the desired method.

Step 5. Coarse-grid correction. Interpolate the obtained coarse-grid solution V to the fine grid using (3.7); the result v is added to the current fine-grid approximation.

Step 6. Postrelaxation sweeps. Improve the corrected fine-grid approximation by ν_2 relaxation sweeps.

We ran two-level cycles with either the pointwise or the zebra relaxation on grids with different aspect ratios and with either full or semicoarsening. Using zero right-hand sides and zero boundary conditions, the function $U(x, y) \equiv 0$ is the exact solution of the differential problem (3.1)–(3.2). This choice of data together with a random choice of the initial approximation facilitates the observation of the cycle asymptotic behavior.

$V_2(1, 1)$ cycles were performed on fine grids with aspect ratios $m = 1, 2, 4, 8, 16$. In these experiments we chose the explicit numerical dissipation factor A on the fine grid assuming that that grid itself was obtained by ($\log_2 m$ steps of) semicoarsening, starting with a uniform target grid. In other words, the total dissipation TV of the algorithm was equal to the inherent numerical dissipation of a uniform grid with meshsize h_y .

Each experiment included three different runs, each starting from a random initial error. Run I used the pointwise relaxation and semicoarsening; Run II, the pointwise relaxation and full coarsening (the coarse-grid discretization being then similar to that on the fine grid); Run III employed the zebra relaxation and semicoarsened coarse grid. Each run consisted of at least 12 cycles, stopping further cycling when the convergence factor was stabilized, in the sense that the largest difference between the convergence factors of the last three cycles did not exceed .01.

The results of these experiments are collected in Table 1. The notation used in the table is the following: $m = h_y/h_x$ is the aspect ratio; h_y is the vertical meshsize. The meshsize in the reference axis always remains the same $h_x = .03125$, except for the last group of experiments for $m = 16$, where $h_x = .125$; t is the nonalignment parameter defined in section 3.1; RC is the relative coupling (cf. section 3.2.3). The column “No.” shows the number of cycles performed until the convergence factor has been stabilized. In the column “Final” the convergence factor of this last cycle (the L_2 -error norm before the cycle divided by that norm after the cycle) is printed. The column “Aver.” exhibits the convergence factor averaged over *all* the cycles performed in the experiment.

The presence of some very large *average* convergence factors in Table 1 emphasizes the astounding efficiency of the cycles in reducing most error components. Indeed, all

TABLE 1
 Multigrid solver for $\frac{\partial^2 U}{\partial \xi^2} = F$ in two dimensions.

Two-level algorithm											
m	t	RC	Asymptotic convergence rate								
			Run I			Run II			Run III		
			No.	Final	Aver.	No.	Final	Aver.	No.	Final	Aver.
1	0.1	0.002	13	4.14	22.5	15	1.10	22.1	15	4.25	56.2
1	0.3	0.011	25	3.49	5.34	20	1.31	9.2	26	3.59	13.2
1	0.5	0.016	26	5.01	6.65	22	1.43	10.5	29	4.99	15.0
1	0.7	0.011	18	3.49	5.89	15	1.32	9.96	20	3.58	14.4
1	0.9	0.002	18	4.19	15.9	16	1.10	20.3	16	4.23	50.1
2	0.1	0.008	18	3.40	7.58	15	1.23	11.8	19	3.73	17.7
2	0.3	0.044	26	6.05	8.67	19	1.96	15.3	21	6.15	23.6
2	0.5	0.063	19	13.7	16.6	30	1.60	12.8	17	20.0	48.8
2	0.7	0.044	26	6.07	8.64	18	1.93	16.0	27	6.05	19.8
2	0.9	0.008	15	3.39	8.51	16	1.22	10.5	16	3.73	20.0
4	0.1	0.032	26	5.34	7.7	19	1.72	13.7	20	5.31	22.1
4	0.3	0.176	17	18.0	23.8	26	3.73	20.3	15	25.8	91.8
4	0.5	0.250	17	21.1	24.3	22	3.05	22.9	17	20.1	56.3
4	0.7	0.176	17	18.2	24.6	28	3.75	19.7	15	25.9	92.3
4	0.9	0.032	22	5.35	8.02	19	1.71	12.4	22	5.3	19.4
8	0.1	0.130	18	16.1	22.2	22	3.00	22.4	16	20.4	80.0
8	0.3	0.706	28	6.18	6.89	27	6.26	14.1	14	31.2	113
8	0.5	1.000	14	3.69	4.60	18	3.7	7.95	17	20.0	44.3
8	0.7	0.706	28	6.19	6.91	26	6.32	14.5	14	31.3	112
8	0.9	0.130	17	16.2	22.8	16	2.98	28.7	16	20.4	78.9
16	0.1	0.518	22	9.56	10.6	22	9.57	20.8	14	31.8	113
16	0.3	2.822	15	1.63	2.43	15	1.64	4.95	13	42.6	135
16	0.5	4.000	13	1.42	2.58	14	1.42	4.86	14	31.3	113
16	0.7	2.822	14	1.63	2.63	13	1.64	5.46	13	41.2	136
16	0.9	0.518	21	9.41	10.7	22	9.69	20.5	13	32.4	119

the components which are smooth in the cross-characteristic direction are converged by the cycle with the efficiency similar to that of the 1D-prototype solver (where just one cycle precisely solves the problem). The exceedingly fast rates in the first few cycles are thus typical to our 2D case, disappearing in three dimensions. Our main concern, though, is the asymptotic (“final”) convergence, which always shows the worst-converging components. Regarding these asymptotic convergence rates, the first obvious result is the superiority of the semicoarsening algorithms over the algorithm with full coarsening. The two semicoarsening algorithms show similar asymptotic convergence factors on grids with small RC ($m = 1, 2, 4$); but when $RC > 1$, only the algorithm with zebra relaxation provides a good asymptotic convergence.

The tremendous discrepancy between “average” and “asymptotic” convergence rates observed in these experiments raises two important issues:

1. The “average” convergence can be misleading, hiding the slowest-to-converge components.
2. The “asymptotic” convergence rates are defined by some of the characteristic error components which happened to be the worst-converging. The residuals of these components are extremely small compared to typical residuals of non-characteristic components of comparable size. This implies that in nonelliptic problems one should not rely on the residual size as a convergence criterion. Other criteria based on direct estimations of the approximation error, e.g., FMG convergence (see section 3.3), should be involved.

3.2.5. Two-level mode analysis. The task of the *full space Fourier mode analysis* exhibited below is to give some confidence that the convergence rates of the two-level cycles presented in the previous section do not deteriorate when the meshsizes tend to zero. Usually, in nonelliptic problems, a *half-space analysis* taking into consideration the boundary influence (see [1], [3, section 7.5], and [10]) should be used, in order to bring out the trouble with characteristic components. However, for our type of solver, where the trouble is removed, the two-level full space analysis gives a good prediction of the overall cycle amplification factor as well.

To simplify the problem we analyze only the two-level cycle with two zebra relaxation sweeps on the fine grid. This simplification allows us to consider only two coupled components at a time, differing by π in their x -directional frequency, while analyzing the eight-color pointwise relaxation would require considering eight components at a time.

The zebra relaxation symbol $Z(\theta)$ is the 2-by-2 amplification matrix acting on a pair of Fourier modes $e^{i(\theta_x i_1 + \theta_y i_2)}$ and $e^{i(\tilde{\theta}_x i_1 + \theta_y i_2)}$, $\tilde{\theta}_x = \theta_x - \text{sign}(\theta_x)\pi$.

$$Z(\theta) = \begin{pmatrix} \frac{C(\theta)(1+C(\theta))}{2} & \frac{C(\tilde{\theta})(1-C(\tilde{\theta}))}{2} \\ \frac{C(\theta)(1-C(\theta))}{2} & \frac{C(\tilde{\theta})(1+C(\tilde{\theta}))}{2} \end{pmatrix},$$

$$C(\theta) = \frac{(1-s)\cos(\theta_x + k\theta_y) + s\cos(\theta_x + (k+1)\theta_y)}{1-s(1-s)(1-\cos(\theta_y)) + A(m^2(1+t^2))(\cos(2\theta_y) - 4\cos(\theta_y) + 3)},$$

where the parameters A , m , k , and s are defined in section 3.1, $\theta = (\theta_x, \theta_y)$, $\tilde{\theta} = (\tilde{\theta}_x, \theta_y)$.

Let θ_ξ be a normalized frequency defined by $\theta_\xi + \pi = (\theta_x + (k+s)\theta_y) \bmod 2\pi$, $|\theta_\xi| \leq \pi$. Following [14] the smoothing factor Sm of the relaxation scheme is defined as the spectral radius of matrix product $Q(\theta)Z(\theta)$, where

$$Q(\theta) = \begin{pmatrix} q & 0 \\ 0 & \tilde{q} \end{pmatrix},$$

if $\frac{\pi}{2} < |\theta_\xi| \leq \pi$, then $q = 1$ and $\tilde{q} = 0$,

otherwise $q = 0$ and $\tilde{q} = 1$.

For any possible slopes t and aspect ratios m (all other parameters (A , k , and s) are derived from these) $Sm < .45$. Thus, the smoothing factor is excellent throughout the relevant range of parameters.

The symbol $V_2(\theta)$ of a two-level $V_2(1, 1)$ cycle is defined by

$$V_2(\theta) = Z(\theta)\left(I - P(\theta)L_c^{-1}(\theta)R(\theta)L_f(\theta)\right)Z(\theta),$$

where

$$L_f(\theta) = \begin{pmatrix} L(\theta) & 0 \\ 0 & L(\tilde{\theta}) \end{pmatrix},$$

TABLE 2
Two-level cycle $V_2(1,1)$ Fourier mode analysis.

m	t	A_{fine}	A_{coarse}	θ_x	θ_y	θ_ξ	Conv. rate
1	0.15	0.0000	0.0012	-0.074	2.528	0.381	0.32
1	0.54	0.0000	0.0117	-2.013	-1.473	-2.808	0.23
1	0.81	0.0000	0.0015	-2.258	2.356	-0.350	0.32
1	0.99	0.0000	0.0000	-3.142	-3.142	0.003	0.04
2	0.54	0.0117	0.0117	-2.011	-1.080	3.006	0.02
2	0.81	0.0015	0.0033	-2.651	1.473	-0.265	0.23
2	0.99	0.0000	0.0000	-3.142	-3.142	-3.079	0.07
4	0.54	0.0117	0.0118	-1.718	-0.736	2.974	0.02
4	0.81	0.0033	0.0034	-2.994	-1.080	-0.210	0.08
4	0.99	0.0000	0.0000	-3.142	-3.142	-3.016	0.14

$$\begin{aligned}
L(\theta) &= \frac{2}{h_y^2} \left(\frac{1}{m^2(1+t^2)} \left((1-s) \cos(\theta_x + k\theta_y) + s \cos(\theta_x + (k+1)\theta_y) \right. \right. \\
&\quad \left. \left. - 1 + s(1-s)(1 - \cos(\theta_y)) \right) - A_{\text{fine}} \left(\cos(2\theta_y) - 4 \cos(\theta_y) + 3 \right) \right), \\
L_c(\theta) &= \frac{2}{h_y^2} \left(\frac{1}{(2m)^2(1+t^2)} \left((1-S) \cos(2\theta_x + K\theta_y) + S \cos(2\theta_x + (K+1)\theta_y) \right. \right. \\
&\quad \left. \left. - 1 + S(1-S)(1 - \cos(\theta_y)) \right) - A_{\text{coarse}} \left(\cos(2\theta_y) - 4 \cos(\theta_y) + 3 \right) \right), \\
P(\theta) &= \left(\begin{array}{c} \frac{1}{2} \left((1-s) \cos(\theta_x + k\theta_y) + s \cos(\theta_x + (k+1)\theta_y) + 1 \right) \\ \frac{1}{2} \left((1-s) \cos(\tilde{\theta}_x + k\theta_y) + s \cos(\tilde{\theta}_x + (k+1)\theta_y) + 1 \right) \end{array} \right), \\
R(\theta) &= P^T(\theta).
\end{aligned}$$

$L(\theta)$ and $L_c(\theta)$ are the symbols of the fine-grid and coarse-grid operators, respectively. Vector-column $P(\theta)$ is the prolongation (3.7) symbol. Its adjoint (transposed) $R(\theta)$ is the symbol of restriction (3.6).

The two-level cycle convergence rate is defined as $\max \rho(V_2(\theta))$, where $\rho(V_2(\theta))$ is the spectral radius of $V_2(\theta)$. The maximum is taken over all $\theta = (\theta_x, \theta_y) \neq (0, 0)$. Some representative samples of two-level convergence rates computed on different grids for different slopes are exhibited in Table 2.

In the table m is the aspect ratio, t is the nonalignment (slope) parameter, A_{fine} and A_{coarse} are the fine-grid and coarse-grid explicit dissipation coefficients, ‘‘Conv. rate’’ is the two-level convergence rate, $e^{i(\theta_x i_1 + \theta_y i_2)}$ is the Fourier component for which this convergence rate was achieved, and θ_ξ is the normalized frequency.

The results of the analysis are in very good agreement with the results of two-level numerical tests exhibited in Table 1.

3.2.6. Multilevel cycle. We performed experiments with a *multilevel* $V(1,1)$ cycle, using the switching criterion introduced in section 3.2.3 above. The multilevel cycle can be defined similarly to the two-level cycle, but Step 4 is replaced with the recursive call to the same cycle applied to the coarse-grid problem. The target-grid experiments showed a stable asymptotic convergence with rates ranging from 2.3

(near-alignment case $t \approx 0$ or $t \approx 1$) to 5 ($t \approx .5$). Convergence rates on coarse grids ($m = 2, 4, 8, 16$) are even better. For detailed tables see [9] and [13].

3.3. FMG solver and numerical experiments. In this section we present the FMG solver (see, e.g., [3]) based on the $V(1, 1)$ cycle. Its *setup* work can be described by the following four steps:

Step 1. Target-grid problem. We formulate the discrete equation (3.3) on the chosen target grid. The parameter A for this grid is set to zero. The total dissipation value for the entire algorithm is defined as this target-grid's inherent numerical dissipation. A proper discretization of the right-hand side f and boundary condition functions g_0 and g_1 is also performed. In our implementation these discrete functions are simply injected from the corresponding continuous ones.

Step 2. Next coarse-grid construction. The next coarse grid is constructed by semicoarsening, as in the cycles described above.

Step 3. Coarse-grid problem. The discretization parameters such as the aspect ratio, the new K, S parameters, and the artificial dissipation coefficient A are calculated for the new grid. The general form of the coarse-grid operator remains the same. The coarse-grid right-hand side function F is formed by the same averaging procedure that is used for the residual transfer inside the cycles, i.e.,

$$F_{i_1, i_2} = .5 f_{2i_1, i_2} + .25 \left[(1 - s) \left(f_{2i_1 - 1, i_2 - k} + f_{2i_1 + 1, i_2 + k} \right) + s \left(f_{2i_1 - 1, i_2 - k - 1} + f_{2i_1 + 1, i_2 + k + 1} \right) \right].$$

The coarse-grid boundary conditions are *injected* from the previous fine grid (averaging could be used as well).

Step 4. Steps 2 and 3 are repeated until the coarsest possible grid is reached and its problem is defined.

The *execution* of the FMG algorithm then involves the following four steps.

Step 1. The coarsest-grid problem is solved by some method.

Step 2. The solution obtained on the current grid is interpolated to the next fine grid to serve as an initial approximation to the fine-grid solution. The "FMG interpolation" used in this step is of the fourth order in the characteristic direction and of the second order in the vertical direction. (The experiments show that even with this lower order vertical interpolation the algorithm successfully reduces all the algebraic errors well below the level of the discretization errors. Nevertheless, in the 3D case below we do use the fourth order interpolation throughout).

Step 3. The obtained initial approximation is improved by one $V(1, 1)$ cycle.

Step 4. We repeat Steps 2 and 3 until the target grid is reached. There we perform one additional improving cycle (mainly for checking purposes).

In our experiments, the right-hand side f and the boundary conditions g_0 and g_1 were chosen so that the function $\sin(\theta_x x + \theta_y y)$ was the exact solution of the differential problem (3.1)–(3.2). Six-level experiments were performed for different values of parameters θ_x and θ_y . The target (finest) grid throughout our experiments was a uniform grid with meshsizes $h_x = h_y = 2^{-7}$. For each component we checked five different characteristic inclinations $t = \tan \psi$. Some representative results are collected in Table 3, where the target-grid discretization error is compared with the algebraic errors at three stages: immediately after the FMG interpolation of the coarse-grid solution and at the end of the first and the second improving cycles. Here

TABLE 3
Multigrid FMG solver for $\frac{\partial^2 U}{\partial \varepsilon^2} = F$ in two dimensions.

t	$\beta_\xi h_y$	$\theta_y h_y$	$\theta_x h_x$	Discr. error	Algebraic error after		
					Interp.	1 cycle	2 cycles
Characteristic components							
0.1	0.04711	0.39270	0.00761	0.0122	0.00619	0.000348	0.000148
0.3	0.04894	0.39270	-0.07093	0.0104	0.0280	0.00204	0.000327
0.5	0.05241	0.39270	-0.14947	0.0424	0.0198	0.00106	0.000127
0.7	0.05722	0.39270	-0.22801	0.151	0.0131	0.000882	0.000206
0.9	0.06306	0.39270	-0.30655	0.354	0.00554	0.000185	$2.16 \cdot 10^{-05}$
0.1	0.04711	1.96350	-0.14947	0.523	0.0424	0.0104	0.00414
0.3	0.04894	1.96350	-0.54217	0.653	0.0487	0.00746	0.00174
0.5	0.05241	1.96350	-0.93487	0.664	0.0414	0.00630	0.000926
0.7	0.05722	1.96350	-1.32757	0.657	0.0492	0.00750	0.00173
0.9	0.06306	1.96350	-1.72027	0.581	0.0398	0.0116	0.00489
Intermediate components							
0.1	0.78515	0.39270	0.74198	0.0708	0.0657	0.00259	0.0012
0.3	0.81565	0.39270	0.66344	0.159	0.0791	0.00331	0.000424
0.5	0.87346	0.39270	0.58490	0.319	0.107	0.00586	0.000241
0.7	0.95364	0.39270	0.50636	0.561	0.106	0.00295	0.000202
0.1	0.78515	1.96350	0.58490	0.353	0.132	0.00509	0.00201
0.3	0.81565	1.96350	0.19220	0.537	0.540	0.0749	0.0131
0.5	0.87346	1.96350	-0.20050	0.326	0.592	0.0646	0.00749
0.7	0.95364	1.96350	-0.59320	0.292	0.308	0.00643	0.00113
0.9	1.05106	1.96350	-0.98590	0.545	0.229	0.0120	0.00486
Noncharacteristic components							
0.1	2.04138	0.39270	1.99198	0.392	1.19	0.00233	0.000666
0.3	2.12069	0.39270	1.91344	0.497	1.30	0.0201	0.000582
0.5	2.27101	0.39270	1.83490	0.697	1.47	0.0555	0.00179
0.7	2.47946	0.39270	1.75636	0.998	1.77	0.0440	0.00169
0.9	2.73277	0.39270	1.67782	1.4	2.13	0.00293	0.000500
0.1	2.04138	1.96350	1.83490	0.555	1.37	0.0284	0.00900
0.3	2.12069	1.96350	1.44220	1.05	1.85	0.0204	0.00343
0.5	2.27101	1.96350	1.04950	1.32	1.84	0.0259	0.0036
0.7	2.47946	1.96350	0.65680	1.32	2.31	0.0492	0.00579
0.9	2.73277	1.96350	0.26410	1.34	2.50	0.0334	0.00857

β_ξ is the characteristic frequency of a given component $\beta_\xi h_y = \theta_x h_x + t\theta_y h_y$. (For more extensive tables see [9] and [13].)

The results show that for all the components the algebraic error after the first cycle is *much less* than the discretization error. In fact, in the case of characteristic components the algebraic error is less than the discretization error already after the solution interpolation from the coarse grid. This is due to the artificial dissipation introduced at the coarse levels, ensuring nearly the same characteristic-component discretization error on all the grids (the differences are proportional to h_y^4). The situation is different for noncharacteristic components since they cannot be well approximated on coarse grids. However, it is exactly these components that are reduced remarkably well by the target-grid cycle. Thus we can conclude that the FMG algorithm requires only one $V(1,1)$ cycle per FMG level, or a total of about 13 “minimal work units” to reach the discretization accuracy for the target-grid approximation.

Note 1. The FMG algorithm work-unit count is about twice larger than usual in uniformly elliptic problems, due to the somewhat more expensive coarser levels, using semicoarsening instead of full coarsening. This increased expense will almost

disappear in three dimensions.

Note 2. In nonelliptic problems there are some “pathological” noncharacteristic components which exhibit unusually small discretization errors. Noncharacteristic components usually possess relatively large discretization errors (compared with characteristic components). However, a very special choice of parameters (solution component U and slope angle ψ) can result in vanishing discretization errors. It is clear that in such special situations we cannot expect the algebraic error to be smaller than or comparable to the discretization error at any stage of the algorithm. In spite of the fact that the algorithm fails to reach the discretization accuracy for these components, the *total* (algebraic plus discretization) error in these special cases is much smaller than in neighboring regular cases. Moreover, upon any reasonable perturbation the behavior becomes normal: the algebraic error after the first improving cycle is already substantially below the level of the discretization error. It is thus clear in any case that the statement that the algebraic error after one cycle is much less than the discretization error will most likely hold in any *real* calculations (where mostly nonpathological components and slope values exist). One can find a detailed analysis of this phenomenon in [9] and [13].

4. 3D sonic flow: The constant coefficient case.

4.1. Problem statement. Let μ and β be arbitrary orthonormal coordinates in \mathbb{R}^3 and the scalar function $\Phi(x, y, z)$ be defined in the unit cube $(x, y, z) \in [0, 1] \times [0, 1] \times [0, 1]$. Then the 3D differential equation we consider is equivalent to

$$(4.1) \quad L\Phi \equiv \frac{\partial^2 \Phi}{\partial \mu^2} + \frac{\partial^2 \Phi}{\partial \beta^2} = F.$$

The “characteristic plane” defined by μ and β is assumed to be *horizontally inclined*; i.e., its normal is closer to the vertical than to any of the horizontal axes. For convenience we assume vertical periodicity: $\Phi(x, y, z) = \Phi(x, y, z + 1)$. On the vertical faces of the cube we assume the Dirichlet boundary condition: $\Phi(x, y, z)$ is prescribed for $x = 0, 1$ and $y = 0, 1$.

Natural (but nonorthogonal) coordinates in the characteristic plane are determined by the intersections of this plane with the x - z and y - z planes; they are

$$\xi = \frac{x + t_x z}{\sqrt{1 + t_x^2}} \quad \text{and} \quad \eta = \frac{y + t_y z}{\sqrt{1 + t_y^2}},$$

where $t_x = \tan(\psi_x)$ is the tangent of the angle between the x -axis and the intersection of the characteristic plane with the x - z coordinate plane; $t_y = \tan(\psi_y)$ is the same for the y - z coordinate plane. The horizontal-inclination assumption means that $0 \leq |t_x|, |t_y| \leq 1$. For simplicity we can assume that $0 \leq t_x, t_y \leq 1$. Let $t_\alpha = t_x - t_y$. We will use the auxiliary characteristic variable

$$\alpha = \frac{x - y + t_\alpha z}{\sqrt{2 + t_\alpha^2}} = \left(\frac{1 + t_x^2}{2 + t_\alpha^2} \right)^{\frac{1}{2}} \xi - \left(\frac{1 + t_y^2}{2 + t_\alpha^2} \right)^{\frac{1}{2}} \eta.$$

Equation (4.1) can be written as

$$(4.2) \quad C_\xi \frac{\partial^2 \Phi}{\partial \xi^2} + C_\eta \frac{\partial^2 \Phi}{\partial \eta^2} + C_\alpha \frac{\partial^2 \Phi}{\partial \alpha^2} = F,$$

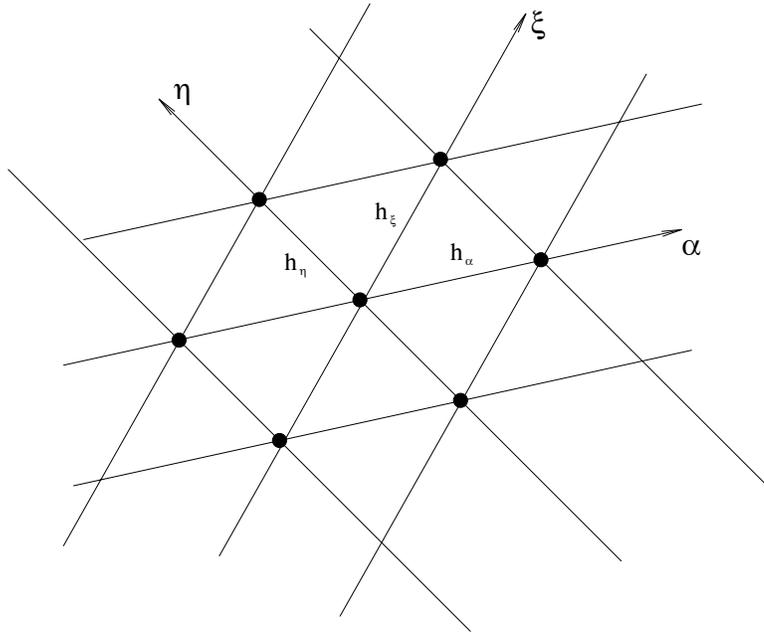


FIG. 4.1. Nonorthogonal grid; seven-point stencil for Laplace operator.

where

$$C_\xi = \frac{(1 + t_x^2) \left[(1 - t_y t_\alpha)^2 + (1 - t_y t_\alpha) (1 + t_x t_\alpha) \right]}{2 + 2t_\alpha^2 + t_\alpha^2 (t_x^2 + t_y^2) + (t_x + t_y)^2};$$

$$C_\eta = \frac{(1 + t_y^2) \left[(1 + t_x t_\alpha)^2 + (1 - t_y t_\alpha) (1 + t_x t_\alpha) \right]}{2 + 2t_\alpha^2 + t_\alpha^2 (t_x^2 + t_y^2) + (t_x + t_y)^2};$$

$$C_\alpha = 1 - \frac{(1 - t_y t_\alpha) (1 + t_x t_\alpha) (2 + t_\alpha^2)}{2 + 2t_\alpha^2 + t_\alpha^2 (t_x^2 + t_y^2) + (t_x + t_y)^2}.$$

Following the guiding principle formulated in section 2.2, we first state the 2D discrete prototype problem on an auxiliary grid induced on a characteristic plane and study its multigrid solver.

4.2. 2D-prototype: Laplacian on a nonorthogonal grid. Let the discrete function ϕ_{i_1, i_2} , defined at the nodes of the grid induced on a representative characteristic plane, be a discrete approximation to function $\Phi(i_1 h, i_2 h, \tilde{z})$, where h is the meshsize in the reference x - y plane and \tilde{z} is uniquely calculated from the condition that the point belongs to the characteristic plane. The seven points used for discretizing (4.2) in this plane are shown in Figure 4.1. The discrete approximation to (4.2) on these points is

$$(4.3) \quad L^h \phi_{i_1, i_2} \equiv \frac{C_\xi}{h_\xi^2} \left(\phi_{i_1+1, i_2} - 2\phi_{i_1, i_2} + \phi_{i_1-1, i_2} \right) + \frac{C_\eta}{h_\eta^2} \left(\phi_{i_1, i_2+1} - 2\phi_{i_1, i_2} + \phi_{i_1, i_2-1} \right) + \frac{C_\alpha}{h_\alpha^2} \left(\phi_{i_1+1, i_2-1} - 2\phi_{i_1, i_2} + \phi_{i_1-1, i_2+1} \right) = f_{i_1, i_2}.$$

This “2D-prototype” equation is approximately satisfied by any vertically smooth solution of the real 3D problem.

The characteristic-plane horizontal-inclination assumption implies that the angle between the axes ξ and η ranges within the relatively narrow interval $[\pi/3, \pi/2]$. In particular, when the angle is $\pi/2$ the grid becomes orthogonal and the variable α disappears ($C_\alpha = 0$). The meshsizes used in (4.3) are $h_\xi = \sqrt{1 + t_x^2}h$; $h_\eta = \sqrt{1 + t_y^2}h$; $h_\alpha = \sqrt{2 + (t_x - t_y)^2}h$. Thus, the discretization (4.3) is h -elliptic and suffers no substantial anisotropy. The boundary condition for the problem is straightforwardly discretized, since the boundary strictly aligns with the grid.

The *multigrid cycle* $V(1, 1)$ which we have tested for the discrete prototype equation consists of a pointwise Gauss–Seidel relaxation, the full-weighting residual transfer, and the linear correction interpolation. The relaxation sweep is performed in the following order: the points with both i_1 and i_2 odd are relaxed first; then the points with both coordinates even; then those with i_1 odd and i_2 even; and the remaining group of points is relaxed last.

Only points with both coordinates even are present on the coarse grid. The full-weighting residual transfer and the linear correction interpolation used in the cycle mimic those usually used with the five-point Laplacian solver on an orthogonal uniform grid. For example, a fine-grid point with coordinates $(2i_1 + 1, 2i_2 + 1)$ sends its residual to (and gets its correction from) the coarse-grid points (i_1, i_2) , $(i_1 + 1, i_2)$, $(i_1, i_2 + 1)$, and $(i_1 + 1, i_2 + 1)$ in equal proportions. Due to the near isotropy we need not use here a line relaxation and/or semicoarsening.

This cycle proved to be very efficient for the prototype equation (4.3) always reducing the error by more than an order of magnitude.

4.3. Discretization. Let a 3D grid have meshsizes h_x , h_y , and h_z in the corresponding directions. Assuming h_z to be the smallest of those, we define the *aspect ratios* of the grid $m_x = h_x/h_z$ and $m_y = h_y/h_z$. Taking into account that the multigrid cycle for the 2D prototype employs full coarsening and also assuming that the target grid is always horizontally uniform, we can restrict our considerations to 3D grids with equal aspect ratios ($h_x = h_y = h$, hence $m_x = m_y = m$). To discretize (4.2) at a given grid node, we consider *ghost points* located at the intersections of the characteristic plane (going through the given node) with the adjacent vertical grid lines. The function values at the ghost points are defined by the linear interpolation from the vertically nearest genuine grid neighbors. This and the addition of an explicit dissipation term result in the following discretization:

$$\begin{aligned}
L^{(h,h,h_z)}\phi_{i_1,i_2,i_3} \equiv & \frac{C_\xi}{m^2(1+t_x^2)h_z^2} \left[s_x (\phi_{i_1+1,i_2,i_3+(k_x+1)} + \phi_{i_1-1,i_2,i_3-(k_x+1)}) \right. \\
& - 2\phi_{i_1,i_2,i_3} + (1-s_x)(\phi_{i_1+1,i_2,i_3+k_x} + \phi_{i_1-1,i_2,i_3-k_x}) \\
& \left. - s_x(1-s_x)(\phi_{i_1,i_2,i_3+1} - 2\phi_{i_1,i_2,i_3} + \phi_{i_1,i_2,i_3-1}) \right] \\
+ \frac{C_\eta}{m^2(1+t_y^2)h_z^2} & \left[s_y (\phi_{i_1,i_2+1,i_3+(k_y+1)} + \phi_{i_1,i_2-1,i_3-(k_y+1)}) - 2\phi_{i_1,i_2,i_3} \right. \\
& + (1-s_y)(\phi_{i_1,i_2+1,i_3+k_y} + \phi_{i_1,i_2-1,i_3-k_y}) \\
(4.4) \quad & \left. - s_y(1-s_y)(\phi_{i_1,i_2,i_3+1} - 2\phi_{i_1,i_2,i_3} + \phi_{i_1,i_2,i_3-1}) \right]
\end{aligned}$$

$$\begin{aligned}
 & + \frac{C_\alpha}{m^2(2+t_\alpha^2)h_z^2} \left[s_\alpha (\phi_{i_1+1,i_2-1,i_3+(k_\alpha+1)} + \phi_{i_1-1,i_2+1,i_3-(k_\alpha+1)}) - 2\phi_{i_1,i_2,i_3} \right. \\
 & \quad + (1-s_\alpha)(\phi_{i_1+1,i_2-1,i_3+k_\alpha} + \phi_{i_1-1,i_2+1,i_3-k_\alpha}) \\
 & \quad \left. - s_\alpha(1-s_\alpha)(\phi_{i_1,i_2,i_3+1} - 2\phi_{i_1,i_2,i_3} + \phi_{i_1,i_2,i_3-1}) \right] \\
 & - A \frac{1}{h_z^2} \left[\phi_{i_1,i_2,i_3+2} - 4\phi_{i_1,i_2,i_3+1} + 6\phi_{i_1,i_2,i_3} - 4\phi_{i_1,i_2,i_3-1} + \phi_{i_1,i_2,i_3-2} \right] = f_{i_1,i_2,i_3},
 \end{aligned}$$

where the integers k_x, k_y , and k_α and the real numbers $0 \leq s_x, s_y, s_\alpha < 1$ are defined by

$$mt_x = k_x + s_x, \quad mt_y = k_y + s_y, \quad \text{and} \quad mt_\alpha = k_\alpha + s_\alpha,$$

and A is the “explicit numerical dissipation” coefficient.

This is a 17-point h -elliptic discretization. Its first differential approximation is

$$(4.5) \quad \tilde{\Delta}^h \phi - h_z^2 \left[A + C_\xi \frac{(1-s_x)^2 s_x^2}{4m^2(1+t_x^2)} + C_\eta \frac{(1-s_y)^2 s_y^2}{4m^2(1+t_y^2)} + C_\alpha \frac{(1-s_\alpha)^2 s_\alpha^2}{4m^2(1+t_\alpha^2)} \right] \phi_{zzzz},$$

where $\tilde{\Delta}^h \phi$ is the first differential approximation to the 2D-prototype discretization (4.3) and ϕ_{zzzz} is the fourth derivative with respect to z . The explicit numerical dissipation parameter A is chosen to ensure the same total numerical dissipation TND on all the grids, where TND is defined to be the coefficient of ϕ_{zzzz} in (4.5). The value of TND is determined by its value at the target grid, where we set $A = 0$.

4.4. Multigrid cycle. The cycle employed here is again the $V(1, 1)$ cycle defined in section 3.2.4, featuring the following components. (Details are omitted, since they are similar to the 2D case and more cumbersome; they can be found in [11].)

4.4.1. Coarse grids. The $V(1, 1)$ cycle employs *semicoarsening*, i.e., the mesh-sizes in the reference plane are doubled at each coarsening step, while the z -direction meshsize remains the same throughout the cycle. (Another approach involving conditional coarsening is discussed in [11].)

4.4.2. Relaxation. Similarly to the 2D case discussed in section 3.2.3, we can separate “dissipative” and “characteristic” couplings; the relative coupling RC defined as their ratio turns out to be

$$(4.6) \quad RC = - \left(\frac{m}{h_z} \right)^2 \frac{TND}{\frac{C_\xi}{1+t_x^2} + \frac{C_\eta}{1+t_y^2} + \frac{C_\alpha}{1+t_\alpha^2}}.$$

Any pointwise relaxation can be unconditionally efficient only on grids where RC is not large. Hence, the global relaxation policy remains the same as in the 2D case: as long as $RC \leq 1$ we apply a point-by-point Gauss–Seidel relaxation; on grids with $RC > 1$ we use a relaxation which simultaneously updates all the points placed on the same vertical grid line. In either case the vertical lines are taken in the red-black ordering. In the pointwise relaxation each such vertical line is relaxed in the four-color order described in section 3.2. One could of course simplify the algorithm and use the vertical line relaxation throughout.

4.4.3. Restriction. The fine-to-coarse residual transfer is of the “full-weighting” type (see [2, section 4.4] or [3, section 4.4]); i.e., the residual (divided by 4) at each fine-grid point is distributed to neighboring coarse-grid points. (The division by 4 expresses the coarse-grid-to-fine-grid mesh-volume ratio.) This distribution follows two rules:

1. A fine-grid point which is geometrically present on the next coarse grid sends its whole residual ($/4$) to its coarse-grid representative.
2. A fine-grid point located in a vertical grid line absent from the coarse grid sends its residual ($/4$) to ghost points (points on its characteristic planes) placed on neighboring vertical grid-lines which do belong to the coarse grid. The transfer is the same as in the 2D-prototype solver. The values received at the ghost points are then redistributed: each ghost point sends its value to the two vertically nearest coarse-grid neighbors. The ratio between the fractions received by each of the neighbors is inversely proportional to the ratio of their distances from the ghost point.

As a result, each coarse-grid point receives residual fractions from 17 fine-grid points.

4.4.4. Prolongation. The coarse-grid correction interpolation is the linear interpolation which constitutes the adjoint to the residual ($/4$) transfer described above.

4.4.5. Numerical experiments. We experimented with two- and five-level $V(1, 1)$ cycles. The results show that the asymptotic convergence rate is good enough (in our two-level tests it is better than three per cycle even in the worst cases) and it does not essentially depend on the cycle depth. Since the behavior of the FMG algorithm is more important, we do not present detailed tables of cycle convergence rates. (They can be found in [11].)

4.5. FMG solver: Numerical results. The full algorithm for solving (4.4) with the boundary conditions mentioned above is the FMG algorithm defined in section 3.3. The total cost of the algorithm in 3D is about *six minimal work units*.

We performed the numerical experiments with a five-level FMG algorithm, having a uniform target grid with meshsize $h = .03125$. The continuous problem’s right-hand side and boundary conditions were chosen so that the solution was $U(x, y, z) = \sin(\theta_x x + \theta_y y + \theta_z z)$. Let ξ and η be the characteristic plane directions defined in section 4.1 and let ζ be the coordinate perpendicular to the characteristic plane. Then the same component U can be expressed in the new variables as $U = \sin(\omega_\xi \xi + \omega_\eta \eta + \omega_\zeta \zeta)$.

Remember that the ξ and η axes can be nonorthogonal. A component is considered as being “characteristic” if both $h_\xi \omega_\xi$ and $h_\eta \omega_\eta$ are small. We tested in our experiments several representative components and various characteristic-plane slopes. A sample of numerical results is shown in Table 4. (More are given in [9], [13], and [11].) The target-grid discretization error is compared in the table with the algebraic error of the target-grid approximations obtained after the FMG interpolation and after the improving $V(1, 1)$ cycle. The results confirm the top efficiency of the algorithm in the sense that after just one target-grid cycle the algebraic error is always *much less* than the discretization error (except for some pathological components (see Note 2 in section 3.3)).

5. 2D sonic flow: The variable coefficient case. The subject of this section is to demonstrate a solver to the 2D sonic flow equation (3.1) where the differentiation direction is no longer constant; i.e., the variable $\xi(x, y)$ is a given nonlinear function of the spatial variables.

TABLE 4
 Multigrid five-level FMG solver for $\frac{\partial^2 \phi}{\partial \mu^2} + \frac{\partial^2 \phi}{\partial \beta^2} = F$ in three dimensions.

Characteristic components							
s_x	s_y	$h_\xi \omega_\xi$	$h_\eta \omega_\eta$	$h_z \theta_z$	Discr. err.	Algebraic error	
						Interp.	V cycle
0.100	0.200	0.01973	0.04005	0.19635	0.000588	0.000323	$4.08 \cdot 10^{-05}$
0.100	0.400	0.01973	0.04229	0.19635	0.000967	0.000527	$6.54 \cdot 10^{-05}$
0.100	0.990	0.01973	0.05526	0.19635	0.00431	0.000107	$1.19 \cdot 10^{-05}$
0.450	0.500	0.02153	0.04391	0.19635	0.00473	0.000458	$4.49 \cdot 10^{-05}$
0.850	0.900	0.02577	0.05283	0.19635	0.0271	$5.57 \cdot 10^{-05}$	$3.54 \cdot 10^{-06}$
0.300	0.200	0.02050	0.04005	0.58905	0.0673	0.00879	0.00172
0.300	0.400	0.02050	0.04229	0.58905	0.0744	0.00908	0.00187
0.300	0.990	0.02050	0.05526	0.58905	0.0978	0.00427	0.000322
0.650	0.500	0.02342	0.04391	0.58905	0.0953	0.00286	0.000487
0.750	0.900	0.02454	0.05283	0.58905	0.117	0.00157	0.000143
0.100	0.200	0.01973	0.04005	1.37445	0.449	0.0210	0.00510
0.100	0.990	0.01973	0.05526	1.37445	0.472	0.00755	0.000742
0.450	0.500	0.02153	0.04391	1.37445	0.494	0.0337	0.00536
0.850	0.900	0.02577	0.05283	1.37445	0.491	0.00922	0.00127
0.200	0.200	0.02002	0.04005	2.35619	0.606	0.0342	0.00521
0.200	0.400	0.02002	0.04229	2.35619	0.619	0.0400	0.00501
0.200	0.990	0.02002	0.05526	2.35619	0.607	0.0307	0.00299
0.350	0.500	0.02080	0.04391	2.35619	0.627	0.0478	0.00294
0.950	0.900	0.02708	0.05283	2.35619	0.596	0.00734	0.000462
Noncharacteristic components							
0.100	0.200	2.07195	2.46293	0.19635	0.399	1.08	0.0685
0.100	0.400	2.07195	2.60114	0.19635	0.432	1.11	0.0658
0.100	0.990	2.07195	3.39843	0.19635	0.633	1.32	0.0291
0.450	0.500	2.26080	2.70016	0.19635	0.423	1.10	0.0746
0.850	0.900	2.70582	3.24918	0.19635	0.142	0.87	0.0280
0.300	0.200	2.15245	2.46293	0.58905	0.440	1.13	0.0893
0.300	0.400	2.15245	2.60114	0.58905	0.467	1.15	0.0905
0.300	0.990	2.15245	3.39843	0.58905	0.506	1.19	0.0233
0.650	0.500	2.45893	2.70016	0.58905	0.439	1.14	0.136
0.750	0.900	2.57709	3.24918	0.58905	0.227	0.943	0.0344
0.100	0.200	2.07195	2.46293	1.37445	0.534	1.25	0.0770
0.100	0.500	2.07195	2.70016	1.37445	0.647	1.36	0.111
0.100	0.990	2.07195	3.39843	1.37445	0.694	1.37	0.0255
0.450	0.500	2.26080	2.70016	1.37445	0.709	1.54	0.301
0.850	0.900	2.70582	3.24918	1.37445	0.200	0.916	0.0715
0.200	0.200	2.10250	2.46293	2.35619	0.989	1.73	0.0399
0.200	0.400	2.10250	2.60114	2.35619	1.30	2.03	0.0526
0.200	0.990	2.10250	3.39843	2.35619	0.854	1.53	0.0298
0.350	0.500	2.18430	2.70016	2.35619	1.55	2.30	0.120
0.950	0.900	2.84369	3.24918	2.35619	0.0845	0.769	0.0406

5.1. Problem statement and discretization. We are still considering (3.1) on the unit square. In the case of variable velocity fields the nonalignment parameter t depends on the spatial coordinates and is defined as the ratio $t = \tan \psi = \frac{\partial \xi}{\partial y} / \frac{\partial \xi}{\partial x}$. We assume some restrictions on the absolute value of t : $|t| \leq 2$. (See discussion in section 6.) In fact, instead of an explicit function $\xi = \xi(x, y)$, we define a smooth velocity field. This means that at each target-grid point (i_1, i_2) we define a unit vector $\bar{u}_{i_1, i_2} = (u_{i_1, i_2}, v_{i_1, i_2})$. This vector shows the velocity direction at the point and yields parameter $t = \tan \psi = -u_{i_1, i_2} / v_{i_1, i_2}$.

As before, the x -axis is the reference one and the boundary conditions remain

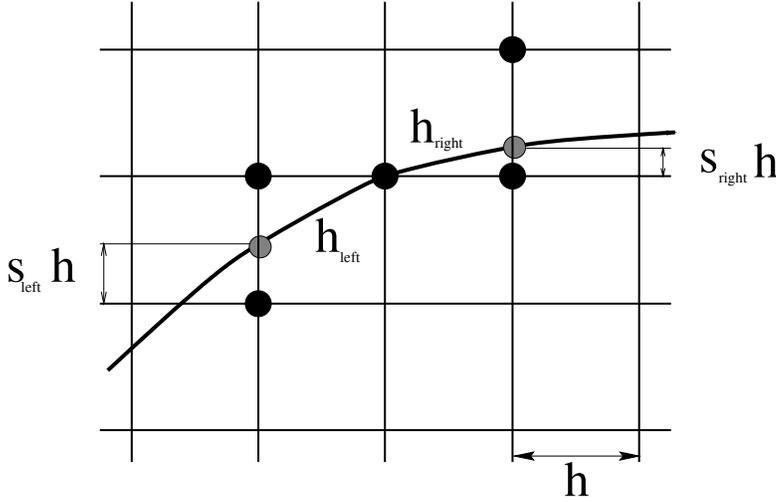


FIG. 5.1. Variable coefficients; low-dimensional prototype. The curved line is the characteristic $y - t(x, y)x = \text{const}$ through the node of discretization.

the same, i.e., Dirichlet conditions in the x direction and periodicity in the vertical y direction. In discretizing this problem we utilize again the low-dimensional prototype. The main difference now is the lack of symmetry and, therefore, parameters related to the left and the right ghost points should be defined separately. At each grid node (i_1, i_2) we now define the following set of stencil parameters (see Figure 5.1):

- a) The right and left stencil half-lengths h_{right} and h_{left} which are the (approximate) distances along the characteristic from the point (i_1, i_2) to the adjacent right and left vertical lines, respectively.
- b) Parameters identifying the vertical coordinates of the ghost points. These are integers k_{right} and k_{left} denoting vertical displacements (in meshsizes) with respect to the point (i_1, i_2) and $s_{\text{right}}, s_{\text{left}}$ ($0 \leq s_{\text{right}}, s_{\text{left}} < 1$) which are the tuning parameters.
- c) The velocity vectors $\bar{\mathbf{u}}_{\text{left}}$ and $\bar{\mathbf{u}}_{\text{right}}$ at the ghost points.

Figure 5.2 explains these parameters pictorially.

The discrete low-dimensional prototype for the variable coefficient problem is still a three-point discretization of the second derivative but on a *nonuniform* grid.

$$\begin{aligned}
 (5.1) \quad & \frac{1}{h_{\text{left}} h_{\text{avg}}} \Phi \left((i_1 - 1)h_x, (i_2 - (k_{\text{left}} + s_{\text{left}}))h_y \right) \\
 & - \frac{2}{h_{\text{left}} h_{\text{right}}} \Phi \left(i_1 h_x, i_2 h_y \right) \\
 & + \frac{1}{h_{\text{right}} h_{\text{avg}}} \Phi \left((i_1 + 1)h_x, (i_2 + (k_{\text{right}} + s_{\text{right}}))h_y \right),
 \end{aligned}$$

where $h_{\text{avg}} = .5(h_{\text{right}} + h_{\text{left}})$. Generally speaking this discrete scheme is just first-order accurate. However, the velocity field smoothness assumption implies that on fine enough grids $h_{\text{right}} = h_{\text{left}} + O(h^2)$, yielding the second-order approximation. Utilizing the same smoothness assumption we can approximate (only in the target-grid discretization) the genuine h_{right} and h_{left} with $\sqrt{h_x^2 + (k_{\text{right}} + s_{\text{right}})^2 h_y^2}$ and $\sqrt{h_x^2 + (k_{\text{left}} + s_{\text{left}})^2 h_y^2}$, respectively. To calculate these parameters on coarse grids

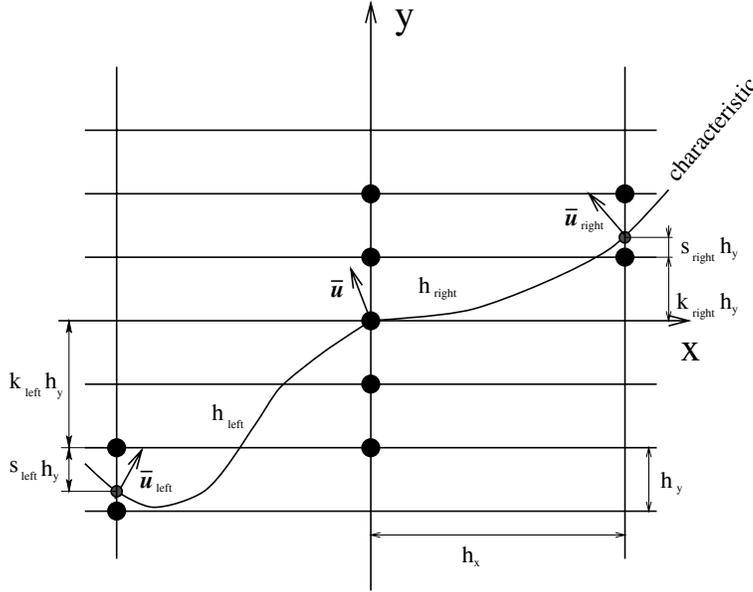


FIG. 5.2. Variable coefficients; nine-point stencil.

we use some cost-effective recursive procedure allowing us to approximate the coarse-grid operator ghost point location with the target-grid accuracy.

In spite of the nonuniformity, there is a $V(1,1)$ -cycle employing red-black relaxations and appropriate intergrid transfers which *solves exactly* this 1D-prototype problem. The 2D V -cycle mimics it in solving the problem in the full dimension.

As before, the coarsening we are using is semicoarsening and the discretization on the semicoarsened grid is derived from the discrete low-dimensional prototype by vertically interpolating values to the ghost points and adding some vertical derivatives to ensure the second-order approximation. The result is a nine-point discretization:

$$\begin{aligned}
 L^{(h_x, h_y)} \phi_{i_1, i_2} \equiv & \frac{1}{h_{\text{left}} h_{\text{avg}}} \left((1 - s_{\text{left}}) \phi_{i_1-1, i_2 - k_{\text{left}}} + s_{\text{left}} \phi_{i_1-1, i_2 - (k_{\text{left}}+1)} \right) \\
 & + \frac{1}{h_{\text{right}} h_{\text{avg}}} \left((1 - s_{\text{right}}) \phi_{i_1+1, i_2 + k_{\text{right}}} + s_{\text{right}} \phi_{i_1+1, i_2 + (k_{\text{right}}+1)} \right) \\
 (5.2) \quad & - \frac{2}{h_{\text{right}} h_{\text{left}}} \phi_{i_1, i_2} + A_2 \frac{1}{h_y^2} \left(\phi_{i_1, i_2-1} - 2\phi_{i_1, i_2} + \phi_{i_1, i_2+1} \right) \\
 & + A_3 \frac{1}{h_y^2} \left(\phi_{i_1, i_2+2} - 2\phi_{i_1, i_2+1} + 2\phi_{i_1, i_2-1} - \phi_{i_1, i_2-2} \right) \\
 & + A_4 \frac{1}{h_y^2} \left(\phi_{i_1, i_2+2} - 4\phi_{i_1, i_2+1} + 6\phi_{i_1, i_2} - 4\phi_{i_1, i_2-1} + \phi_{i_1, i_2-2} \right).
 \end{aligned}$$

The parameters A_2 and A_3 are chosen to keep the second-order accuracy

$$\begin{aligned}
 A_2 = & - \frac{h_y^2}{v^2 (h_{\text{left}} + h_{\text{right}})} \left(\frac{1}{h_{\text{left}}} \left((1 - s_{\text{left}}) s_{\text{left}} v_{\text{left}}^2 \right) \right. \\
 & \left. + \frac{1}{h_{\text{right}}} \left((1 - s_{\text{right}}) s_{\text{right}} v_{\text{right}}^2 \right) \right);
 \end{aligned}$$

$$(5.3) \quad A_3 = -\frac{h_y^2}{3v^3(h_{\text{left}} + h_{\text{right}})} \left(\frac{1}{h_{\text{left}}} \left((1 - 2s_{\text{left}})(1 - s_{\text{left}})s_{\text{left}}v_{\text{left}}^3 \right) - \frac{1}{h_{\text{right}}} \left((1 - 2s_{\text{right}})(1 - s_{\text{right}})s_{\text{right}}v_{\text{right}}^3 \right) \right).$$

Here $v = \cos \psi$ is the cosine of the nonalignment angle at point (i_1, i_2) .

$$\begin{aligned} \bar{\mathbf{u}}_{\text{left}} &= (1 - s_{\text{left}})\bar{\mathbf{u}}_{i_1-1, i_2-k_{\text{left}}} + s_{\text{left}}\bar{\mathbf{u}}_{i_1-1, i_2-(k_{\text{left}}+1)}; \\ \bar{\mathbf{u}}_{\text{right}} &= (1 - s_{\text{right}})\bar{\mathbf{u}}_{i_1+1, i_2+k_{\text{right}}} + s_{\text{right}}\bar{\mathbf{u}}_{i_1+1, i_2+(k_{\text{right}}+1)}. \end{aligned}$$

The first differential approximation to (5.2) with the given choice of A_2 and A_3 is

$$(5.4) \quad \phi_{\xi\xi}^h - h_y^2 [A_4 v^4 + IND_4] \phi_{\eta\eta\eta\eta},$$

where $\phi_{\xi\xi}^h$ is the first differential approximation to the 1D prototype (5.1), η is the cross-characteristic (streamwise) variable, and IND_4 is the inherent numerical dissipation coefficient:

$$\begin{aligned} IND_4 &= \frac{h_y^2}{12(h_{\text{left}} + h_{\text{right}})} \left(\frac{1}{h_{\text{left}}} \left((1 - 3s_{\text{left}} + 3s_{\text{left}}^2)(1 - s_{\text{left}})s_{\text{left}}v_{\text{left}}^4 \right) \right. \\ &\quad \left. + \frac{1}{h_{\text{right}}} \left((1 - 3s_{\text{right}} + 3s_{\text{right}}^2)(1 - s_{\text{right}})s_{\text{right}}v_{\text{right}}^4 \right) \right) + \frac{1}{12}v^4 A_2. \end{aligned}$$

The inherent numerical dissipation coefficient decreases (roughly by a factor of 4) in semicoarsening since the values h_{left} and h_{right} are about doubled in each semicoarsening step.

The parameter A_4 is the explicit numerical dissipation coefficient. On the target grid it is set to zero. On coarse grids its value is chosen to retain the same total cross-characteristic numerical dissipation ($TND = A_4 v^4 + IND_4$) as on the target grid. In fact, the target-grid cross-characteristic numerical dissipation can differ from point to point. Therefore, the coarse-grid TND at a given point should represent some local average of the target-grid IND_4 .

5.2. Multigrid cycle. The algorithm described in this section is very efficient, even though not fully optimal. To simplify the presentation we avoid here the mechanism of switching between pointwise and “zebra” relaxation, opting to start line relaxation already on the target grid. We are focusing here on the main (although just technical) difficulty of intergrid transfers. We adopt the following notation conventions:

1) The superscript of each parameter denotes the point this parameter belongs to. For example, $h_{\text{left}}^{(i_1, i_2)}$ (formerly h_{left}) is the distance along the characteristic from the point (i_1, i_2) to the left neighboring vertical grid line.

2) Uppercase letters denote parameters related to the coarse-grid discretization while lowercase letters refer to fine-grid parameters. For example, $h_{\text{right}}^{(2i_1, i_2)}$ and $H_{\text{right}}^{(i_1, i_2)}$ are the right stencil half-length at the same geometric point on the fine and coarse grids, respectively.

5.2.1. Fine-to-coarse parameter transfers. *The coarse-grid velocity field is obtained from the fine-grid one by simple injection.*

$$(5.5) \quad \bar{\mathbf{U}}^{(i_1, i_2)} = \bar{\mathbf{u}}^{(2i_1, i_2)}.$$

The cost-effective accurate procedure for obtaining coarse-grid coordinates of ghost points is presented in this paragraph. We first remark that all the necessary geometrical information about the (say, left) ghost point location can be kept by storing just its full vertical coordinate. Storing two parameters (k_{left} and s_{left}) instead is preferred for algorithm efficiency considerations to avoid multiple recalculations of these two parameters required at each stage of the multilevel cycle. In this paragraph, though, the one-parameter representation is more convenient. Therefore, at each point we introduce parameters y_{left} and y_{right} (on coarse grid Y_{left} and Y_{right}) which represent the vertical coordinate of the left and right ghost points, respectively. In these parameters, the recursive relations are

$$\begin{aligned} Y_{\text{left}}^{(i_1, i_2)} &= \left(1 - s_{\text{left}}^{(2i_1, i_2)}\right) y_{\text{left}}^{(2i_1-1, i_2 - k_{\text{left}}^{(2i_1, i_2)})} + s_{\text{left}}^{(2i_1, i_2)} y_{\text{left}}^{(2i_1-1, i_2 - (k_{\text{left}}^{(2i_1, i_2)} + 1))}, \\ Y_{\text{right}}^{(i_1, i_2)} &= \left(1 - s_{\text{right}}^{(2i_1, i_2)}\right) y_{\text{right}}^{(2i_1+1, i_2 + k_{\text{right}}^{(2i_1, i_2)})} + s_{\text{right}}^{(2i_1, i_2)} y_{\text{right}}^{(2i_1+1, i_2 + (k_{\text{right}}^{(2i_1, i_2)} + 1))}. \end{aligned}$$

Having these coordinates we reconstruct the K and S parameters:

$$\begin{aligned} K_{\text{left}}^{(i_1, i_2)} &= \text{floor}\left(\frac{i_2 H_y - Y_{\text{left}}^{(i_1, i_2)}}{H_y}\right), & S_{\text{left}}^{(i_1, i_2)} &= \frac{i_2 H_y - Y_{\text{left}}^{(i_1, i_2)}}{H_y} - K_{\text{left}}^{(i_1, i_2)}, \\ K_{\text{right}}^{(i_1, i_2)} &= \text{floor}\left(\frac{Y_{\text{right}}^{(i_1, i_2)} - i_2 H_y}{H_y}\right), & S_{\text{right}}^{(i_1, i_2)} &= \frac{Y_{\text{right}}^{(i_1, i_2)} - i_2 H_y}{H_y} - K_{\text{right}}^{(i_1, i_2)}, \end{aligned}$$

where $\text{floor}(x)$ is the largest integer which is not greater than x .

The coarse-grid stencil half-lengths are computed from the fine-grid stencil half-lengths:

$$\begin{aligned} H_{\text{left}}^{(i_1, i_2)} &= h_{\text{left}}^{(2i_1, i_2)} + \left(1 - s_{\text{left}}^{(2i_1, i_2)}\right) h_{\text{left}}^{(2i_1-1, i_2 - k_{\text{left}}^{(2i_1, i_2)})} + s_{\text{left}}^{(2i_1, i_2)} h_{\text{left}}^{(2i_1-1, i_2 - (k_{\text{left}}^{(2i_1, i_2)} + 1))}, \\ H_{\text{right}}^{(i_1, i_2)} &= h_{\text{right}}^{(2i_1, i_2)} + \left(1 - s_{\text{right}}^{(2i_1, i_2)}\right) h_{\text{right}}^{(2i_1+1, i_2 + k_{\text{right}}^{(2i_1, i_2)})} + s_{\text{right}}^{(2i_1, i_2)} h_{\text{right}}^{(2i_1+1, i_2 + (k_{\text{right}}^{(2i_1, i_2)} + 1))}. \end{aligned}$$

5.2.2. Residual and total dissipation transfers. *The fine-to-coarse residual transfer approximating its 1D-prototype counterpart is given by*

$$\begin{aligned} R_{i_1, i_2} &= \left(I_h^H r\right)_{i_1, i_2} = \frac{1}{H_{\text{left}}^{(i_1, i_2)} + H_{\text{right}}^{(i_1, i_2)}} \left(r_{2i_1, i_2} \left(h_{\text{left}}^{(2i_1, i_2)} + h_{\text{right}}^{(2i_1, i_2)} \right) \right. \\ &\quad + \left(1 - s_{\text{left}}\right) r_{2i_1-1, i_2 - k_{\text{left}}} h_{\text{left}}^{(2i_1-1, i_2 - k_{\text{left}})} \\ &\quad + s_{\text{left}} r_{2i_1-1, i_2 - (k_{\text{left}} + 1)} h_{\text{left}}^{(2i_1-1, i_2 - (k_{\text{left}} + 1))} \\ &\quad + \left(1 - s_{\text{right}}\right) r_{2i_1+1, i_2 + k_{\text{right}}} h_{\text{right}}^{(2i_1+1, i_2 + k_{\text{right}})} \\ &\quad \left. + s_{\text{right}} r_{2i_1+1, i_2 + (k_{\text{right}} + 1)} h_{\text{right}}^{(2i_1+1, i_2 + (k_{\text{right}} + 1))} \right). \end{aligned}$$

As already mentioned, the fine-grid total cross-characteristic dissipation is also subject to averaging. The target-grid parameters A_2 and A_3 (see (5.3)) were chosen to annihilate the coefficients of the second and the third cross-characteristic (η -) derivatives in the first differential approximation (5.4). Setting $A_4 = 0$ means that the target-grid inherent numerical dissipation constitutes the total numerical cross-characteristic dissipation at each point ($TND = IND_4$). The fine-to-coarse total dissipation transfer is performed in the same way as the residual transfer.

$$\begin{aligned} TND_{\text{coarse}}^{(i_1, i_2)} &= \left(I_h^H TND_{\text{fine}} \right)^{(i_1, i_2)} \\ &= \frac{1}{H_{\text{left}}^{(i_1, i_2)} + H_{\text{right}}^{(i_1, i_2)}} \left(TND_{\text{fine}}^{(2i_1, i_2)} \left(h_{\text{left}}^{(2i_1, i_2)} + h_{\text{right}}^{(2i_1, i_2)} \right) \right. \\ &\quad + \left(1 - s_{\text{left}} \right) TND_{\text{fine}}^{(2i_1-1, i_2-k_{\text{left}})} h_{\text{left}}^{(2i_1-1, i_2-k_{\text{left}})} \\ &\quad + s_{\text{left}} TND_{\text{fine}}^{(2i_1-1, i_2-(k_{\text{left}}+1))} h_{\text{left}}^{(2i_1-1, i_2-(k_{\text{left}}+1))} \\ &\quad + \left(1 - s_{\text{right}} \right) TND_{\text{fine}}^{(2i_1+1, i_2+k_{\text{right}})} h_{\text{right}}^{(2i_1+1, i_2+k_{\text{right}})} \\ &\quad \left. + s_{\text{right}} TND_{\text{fine}}^{(2i_1+1, i_2+(k_{\text{right}}+1))} h_{\text{right}}^{(2i_1+1, i_2+(k_{\text{right}}+1))} \right). \end{aligned}$$

On the coarse grid the inherent dissipation IND_4 decreases and, therefore, to satisfy

$$TND = A_4 v^4 + IND_4,$$

the coarse-grid parameter A_4 is set to a suitable positive value at each coarse-grid point.

5.2.3. Correction interpolation. This is a linear interpolation derived from the interpolation used in the 1D-prototype solver.

$$(5.6) \quad \left\{ \begin{array}{l} v_{2i_1, i_2} = V_{i_1, i_2}, \\ v_{2i_1+1, i_2} = \frac{1}{h_{\text{left}} + h_{\text{right}}} \left(\left((1 - s_{\text{left}}) V_{i_1, i_2-k_{\text{left}}} + s_{\text{left}} V_{i_1, i_2-(k_{\text{left}}+1)} \right) h_{\text{right}} \right. \\ \quad \left. + \left((1 - s_{\text{right}}) V_{i_1+1, i_2+k_{\text{right}}} + s_{\text{right}} V_{i_1+1, i_2+(k_{\text{right}}+1)} \right) h_{\text{left}} \right), \end{array} \right.$$

where V denotes the solution to the coarse-grid problem, v denotes the correction to the fine-grid solution approximation, and the fine-grid parameters $h_{\text{left}}, h_{\text{right}}, k_{\text{left}}, k_{\text{right}}, s_{\text{left}},$ and s_{right} written without superscripts are taken at the fine-grid point $(2i_1 + 1, i_2)$.

5.2.4. Multilevel cycle: Numerical experiments. The formal description of the multilevel $V(1, 1)$ cycle is essentially the same as in section 3.2. Now, however, Step 3 involves transferring all the necessary parameters, including the velocity field and the total numerical dissipation.

Numerical experiments with multigrid $V(1, 1)$ cycles were performed for a smooth variable vector velocity field. Let $x_0 = \sin(\arctan(2))$. Then the characteristic field (orthogonal to this velocity field) can be described as follows: the characteristic going

TABLE 5
Asymptotic convergence rate.

m	Cycles	Final	Average
1	14	2.67	3.89
2	17	4.52	5.56
4	19	6.93	8.39
8	39	12.88	15.28

through the point (x_0, y) is an arc of the unit circle centered at $(x_0, y + 1)$. This simple field depends on the x -coordinate only and satisfies the following properties:

- 1) It is a smooth variable field.
- 2) The inclination parameter t is bounded: $|t| \leq 2$.
- 3) The maximal value of $|t|$ ($t = -2$) is reached at $x = 0$.

As before, we impose zero boundary conditions $g_1(y) \equiv g_2(y) \equiv 0$ and zero right-hand side $f_{i_1, i_2} \equiv 0$. This implies the zero function as the exact solution of (3.4). The initial approximation inside the domain is taken to be random. We perform experiments on grids with the aspect ratios $m = 1, 2, 4, 8$. The y -directional meshsize is $h_y = 2^{-7}$ for all the experiments. This means that the target grid is always the uniform grid of 129×129 points. Parameter values for cycles with fine-grid aspect ratios $m \neq 1$ are derived from the assumption that the grid itself was obtained by $\log_2 m$ semicoarsening steps starting from that target grid. Every experiment consists of at least 12 successive $V(1, 1)$ cycles. The convergence factor for each cycle is defined as the ratio of the L2-norm of the residual function before the cycle to that norm after the cycle. The cycling is stopped if the maximal difference between convergence rates of the last three cycles does not exceed 0.01. The results of the experiments are collected in Table 5.

The notation is the following: m is the aspect ratio, the column ‘‘Cycles’’ shows the number of cycles performed in each experiment, the column ‘‘Final’’ exhibits the asymptotic convergence rates, and the column ‘‘Average’’ gives the convergence factor averaged over all the cycles performed in the experiment.

5.3. FMG solver. A five-level FMG algorithm employing just one $V(1, 1)$ cycle on each level is tested in this section. The variable velocity field is chosen so that the velocity direction at each point coincides with the outward normal to a circle centered at $(1, -1)$. For this choice the new curvilinear coordinates are $\xi = \arctan(\frac{1-x}{1+y})$; $\eta = \sqrt{(1-x)^2 + (1+y)^2}$. The domain of interest is still the unit square and, therefore, the inclination parameter $t = \tan(\xi) = \frac{1-x}{1+y}$ satisfies the inequalities $0 \leq t \leq 1$. This velocity field is not smooth because of the periodic boundary conditions in the y direction. It implies velocity direction discontinuity at the line $y = 0$. However, even in this case the tested algorithm demonstrates good convergence properties that highlight its efficiency and robustness. The target grid we use in the tests is a uniform grid with meshsizes $h_x = h_y = 1/64$. The boundary conditions for (3.4) are $g_0(y) = \sin(\omega y)$; $g_1(y) = \sin(\theta + \omega y)$. We tested the FMG solver for problems with different right-hand side functions. The target-grid approximate solutions obtained in these experiments always attained algebraic errors well below the discretization errors. For presentation purposes here we choose the right-hand side function

$$F(x, y) = -\beta_1^2 \sin(\theta x + \omega y) + \beta_2 \cos(\theta x + \omega y),$$

$$\beta_1 = \frac{\eta(\theta + \omega t)}{\sqrt{1+t^2}}, \quad \beta_2 = \frac{\eta(\theta t - \omega)}{\sqrt{1+t^2}}.$$

TABLE 6
Five-level 1FMG solver.

$\beta_\xi h_x$	ωh_y	θh_x	Discretization error	Algebraic error		
				Interp.	1 cycle	2 cycles
0.147	0.098	0.098	0.10299	0.10877	0.010929	0.0017899
0.540	0.491	0.098	0.85212	0.19956	0.038575	0.0047681
1.031	0.982	0.098	0.5311	0.50828	0.036816	0.006457
2.013	1.963	0.098	0.89483	3.211	0.17133	0.034948
0.344	0.098	0.491	0.25935	0.25098	0.030913	0.0058224
0.736	0.491	0.491	1.0205	0.31052	0.054264	0.0059497
1.227	0.982	0.491	0.65877	0.8174	0.048544	0.0089859
2.209	1.963	0.491	1.1011	3.7617	0.21005	0.046631
0.589	0.098	0.982	0.72293	0.59177	0.12307	0.022576
0.982	0.491	0.982	1.4687	0.65473	0.070375	0.0080511
1.473	0.982	0.982	0.90164	1.4269	0.058127	0.010506
2.454	1.963	0.982	1.4712	4.9979	0.25662	0.059136
1.080	0.098	1.963	2.4668	2.7428	0.62663	0.13894
1.473	0.491	1.963	2.4685	2.1171	0.10683	0.021413
1.964	0.982	1.963	2.0744	3.4209	0.19794	0.044562
2.945	1.963	1.963	3.6965	14.576	1.3898	0.15049

This choice of F would imply that the function $U(x, y) = \sin(\theta x + \omega y)$ is the exact solution of the differential problem, but the periodicity of the velocity field breaks this claim. That is why in all the experiments the discretization error function is approximated by calculating the difference between the exact solutions of the problem discretized on the target grid and on a finer grid with $h_x = h_y = 1/128$. The algebraic error function is, as before, the difference between the exact and the current approximate solutions on the target grid. The results of numerical experiments are exhibited in Table 6. The most important property demonstrated in the experiments is that the algorithm produces an accurate solution for both characteristic and non-characteristic components. The choice of the right-hand side function F allows us to separate roughly these two types of components. Small values (in comparison with π) of the normalized characteristic frequency $\beta_\xi h_x = (\beta_1^4 + \beta_2^2)^{\frac{1}{4}} h_x$ correspond to a characteristic component while $\beta_\xi h_x \geq \pi/2$ indicates a noncharacteristic component. We have experimented with different values of θ and ω . In Table 6 we compare the L2-norm of the target-grid discretization error function with the L2-norms of the algebraic error functions after the FMG interpolation (Interp.), after the first target-grid cycle (1 cycle) and after the second target-grid cycle (2 cycles). The second cycle is performed just for comparison since the algebraic error after the first cycle is always below the discretization error. Similar to the constant coefficient case, for characteristic components, the algebraic error is better than the discretization one already after the FMG interpolation. For noncharacteristic components, one $V(1, 1)$ cycle is enough to drastically reduce the initial error.

6. Discussion. The research reported in this paper was mainly motivated by the need to develop “textbook multigrid efficiency” for compressible fluid dynamics. For any *near-sonic* flow regimes, both the fundamental difficulty (poor coarse-grid correction for smooth characteristic components) and an efficient way to treat it (semi-coarsening and artificially balanced coarse-grid added dissipation) are essentially the same as described above. In fact, in the *pure* sonic case this difficulty is most pronounced, which is why we have researched it first. Extensions to near-sonic cases of

the linearized equation (2.1) are discussed in [8].

Developing efficient solvers for the linearized equation paves the way toward constructing a robust solver to the transonic nonlinear full potential equation (see [8]) which, in turn, should result in a very efficient solver to the Euler system.

Indeed, the full potential operator (2.1) is a factor of the principal determinant of the Euler system. As shown in the past (see [3], [4], [5], and [10]), to obtain the textbook multigrid efficiency for any discretized PDE system it is necessary and usually (with proper boundary treatment) also sufficient to attain that efficiency for each factor of the PDE principal determinant. The way to separate the factors is by *distributed* (and possibly also *weighted*) relaxation schemes in which there corresponds to each factor a “ghost” discrete function. The latter can be directly relaxed for its corresponding factor, dictating a resulting pattern of changes to be distributed to the *actual* discrete functions (see details in [3, section 3.7] and [16]). In this framework the entire multigrid solver described in this paper (extended to near-sonic regimes) can be used as one of several distributed-relaxation *steps* (applied to the ghost function associated with the full-potential factor) in an *outer* multigrid solver for the Euler system. (See also [6] and [7, section 2].)

More generally, each of the multigrid algorithms presented in this article can serve as a *relaxation step* in the framework of an outer solver to a more complicated problem. In this way the approach can be efficiently applied even when the characteristic manifold changes its general orientation over various parts of the domain. In such a case one should divide the entire near-sonic domain into subdomains (each occupying an $O(1)$ part of the total volume and having a unique reference axis (or axes) compatible throughout with the characteristic manifold orientation), applying the above algorithms separately on each of the subdomains, as part of the outer relaxation scheme. The outer solver can be a multigrid solver using, say, full coarsening. This approach is still very cost effective (especially in three dimensions).

Acknowledgment. We thank the anonymous reviewer whose insightful comments allowed us to improve our presentation.

REFERENCES

- [1] A. BRANDT, *Multigrid Solvers for Non-elliptic and Singular-perturbation Steady-state Problems*, The Weizmann Institute of Science, Rehovot, Israel, 1981, manuscript.
- [2] A. BRANDT, *Guide to multigrid development*, in *Multigrid Methods*, Lecture Notes in Math. 960, W. Hackbusch and U. Trottenberg, eds., Springer-Verlag, Berlin, 1982, pp. 220–312.
- [3] A. BRANDT, *Multigrid Techniques: 1984 Guide with Applications to Fluid Dynamics*, Comput. Math. Group, University of Colorado, Denver, CO, unpublished. Also available from Secretary, Department of Mathematics, The Weizmann Institute of Science, Rehovot, Israel.
- [4] A. BRANDT, *Rigorous local mode analysis of multigrid*, in *Proc. 4th Copper Mountain Conf. on Multigrid Methods*, Copper Mountain, Colorado, 1989.
- [5] A. BRANDT, *Rigorous quantitative analysis of multigrid. I: Constant coefficients two-level cycle with L_2 -norm*, *SIAM J. Numer. Anal.*, 31 (1994), pp. 1695–1730.
- [6] A. BRANDT, *Barriers to Achieving Textbook Multigrid Efficiency in CFD*, Interim report 32, ICASE, 1998.
- [7] A. BRANDT, *The Gauss center research in scientific computation*, *Electron. Trans. Numer. Anal.*, 6 (1997), pp. 1–34.
- [8] A. BRANDT AND B. DISKIN, *Efficient multigrid solvers for the linearized transonic full potential operator*, *J. Comput. Phys.*, submitted.
- [9] A. BRANDT AND B. DISKIN, *Multigrid Solvers for the Non-aligned Sonic Flow: The Constant Coefficient Case*, Gauss Center Report WI/GC-8, The Weizmann Institute of Science, Rehovot, Israel, 1997.

- [10] A. BRANDT AND I. YAVNEH, *On multigrid solution of high-Reynolds incompressible entering flow*, J. Comput. Phys., 101 (1992), pp. 151–164.
- [11] B. DISKIN, *Multigrid Solver for Potential-flow Equation*, Intermediate report on research leading to Ph.D., The Weizmann Institute of Science, Rehovot, Israel, 1995.
- [12] B. DISKIN, *Multigrid algorithm with conditional coarsening for the non-aligned sonic flow*, Electron. Trans. Numer. Anal., 6 (1997), pp. 106–120.
- [13] B. DISKIN, *Multigrid Solver for Potential-flow Equation*, Final report on research leading to Ph.D., The Weizmann Institute of Science, Rehovot, Israel, 1997.
- [14] K. STUBEN AND U. TROTTEBERG, *Multigrid methods: Fundamental algorithms, model problem analysis and application*, in Multigrid Methods, Lecture Notes in Math. 960, W. Hackbusch and U. Trottenberg, eds., Springer-Verlag, Berlin, 1982, pp. 1–176.
- [15] N. N. YANENKO AND Y. I. SHOKIN, *Correctness of first differential approximations of difference schemes*, Dokl. Akad. Nauk SSSR (N.S.), 182 (1968), pp. 776–778 (in Russian).
- [16] I. YAVNEH, *A method for devising efficient multigrid smoothers for complicated PDE systems*, SIAM J. Sci. Comput., 14 (1993), pp. 1437–1463.