Improved Boundary Conditions for Cell-centered Difference Schemes

Rob F. Van der Wijngaart, MRJ Technology Solutions
Goetz H. Klopfer, MCAT Inc.
NASA Ames Research Center, Moffett Field, CA 94035

Abstract—Cell-centered finite-volume (CCFV) schemes have certain attractive properties for the solution of the equations governing compressible fluid flow. Among others, they provide a natural vehicle for specifying flux conditions at the boundaries of the physical domain. Unfortunately, they lead to slow convergence for numerical programs utilizing them. In this report a method for investigating and improving the convergence of CCFV schemes is presented, which focuses on the effect of the numerical boundary conditions. The key to the method is the computation of the spectral radius of the iteration matrix of the entire discretized system of equations, not just of the interior point scheme or the boundary conditions.

1 Introduction

The traditional finite-volume schemes are based on cell-centered schemes. In these schemes the flow quantities are stored at the center of the computational cell and the metrics defining the shape of the cell are at the cell vertices or the nodes of the computational grid. An alternative method is the cell-vertex based scheme, where the physical and the geometric quantities are stored at the cell vertices. In both methods the viscous flow equations are solved using the integral form of the Navier-Stokes equations, which guarantees that the conservation principles of the governing equations are also satisfied in the discretized form.

Cell-centered schemes have been in wide-spread use for a number of years [5, 6, 9], but more recently cell-vertex schemes have been gaining popularity [4, 8, 10, 11]. Cell-centered finite-volume schemes have certain desirable properties for the solution of compressible fluid flow problems. They are generally more accurate than cell-vertex based schemes, use fewer operations per point since averaging is done over smaller stencils, and also lead to more physical boundary conditions, since flux computations at the boundary of the computational domain coincide with actual grid boundaries. In contrast, cell-vertex schemes usually employ extrapolations from the interior of the grid to produce boundary fluxes. In addition, cell-vertex schemes require an extra step to recover the update of the physical quantities at the cell vertices.

The accuracy claim made above for the cell-centered scheme is disputed by Roe [10] and Rossow [11], who maintain that for non-smooth meshes that are typical of realistic geometries cell-vertex schemes are at least first order accurate, whereas cell-centered schemes are of zeroth order. In other

words, the truncation error remains finite for vanishing grid spacing. However, it has been shown [13, 14] that even for non-smooth meshes, cell-centered schemes are always first order if the fluxes at the cell faces are computed properly. On smooth meshes cell-centered schemes are second order accurate, but cell-vertex schemes are second order only under special circumstances.

One reason why cell-centered schemes are still not much used is that they tend to lead to very slow convergence in viscous flow situations with explicit boundary value updates. This has not been a major difficulty in the earlier application of these schemes when the flow solvers were explicit. However, with the advent of implicit methods such as the alternating-direction implicit schemes [1] developed for the Navier-Stokes equations, the slow convergence rates have been a major handicap. The slow convergence rates of cell-centered schemes were first noted when some of the popular finite-difference codes, such as CNS [2] and ARC2D [12], were converted to cell-centered finite-volume codes (see for example the papers by Klopfer [7] and Yoon [15]). The time steps had to be reduced by more than an order of magnitude to maintain numerical stability. A similar deterioration of convergence was noted with cell-centered unstructured grid schemes [3]. As will be shown in this study, the major cause of the slow convergence is the explicit application of certain boundary conditions. Finite-difference and cell-vertex finite-volume schemes are impervious to this problem.

The slow convergence can be relieved by imposing the boundary conditions implicitly. However implicit boundary condition implementations are often difficult or impossible to achieve with popular structured-grid solution algorithms such as Beam-Warming ADI schemes [1] or LUSGS [15], and with unstructured-grid algorithms [3] as well. Here we will demonstrate, through the study of a succession of model equations of increasing complexity, that the judicious choice of relaxation factors for the explicit update of boundary values can render such updates as efficient numerically as would be a completely implicit implementation.

2 Steady, scalar 1-D model equation

A simple model equation for viscous fluid flow is the linearized, steady, one-dimensional convection/diffusion equation.

$$c\frac{du}{dx} = \mu\frac{d^2u}{dx^2} \tag{1}$$

Assuming unit density throughout, the convection speed c and viscosity μ can be combined into the Reynolds number: $Re \stackrel{\text{def}}{=} c/\mu$. We also introduce the cell Reynolds number on a uniform grid with mesh spacing Δ_x: $Re_c \stackrel{\text{def}}{=} Re\Delta_x$. Applying central differencing to equation 1 leads to the following discrete equation.

$$(1 + Re_c/2)u_{i-1} - 2u_i + (1 - Re_c/2)u_{i+1} = 0\,. \tag{2}$$

A second-order-accurate finite-volume formulation of the Dirichlet boundary conditions $u|_{x=a} = u_a$, $u|_{x=b} = u_b$ is obtained by setting

$$\frac{1}{2}(u_1 + u_2) = u_a,\ \tfrac{1}{2}(u_{n-1} + u_n) = u_b\,, \tag{3}$$

where the true physical boundaries at $x = a$ and $x = b$ are at half-way points x_a and x_b between interior grid points and ghost points. See Figure 1.

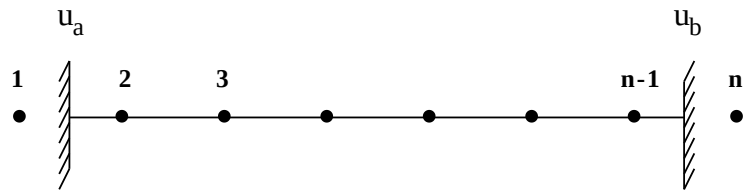


Figure 1: One-dimensional finite-volume grid

If the one-dimensional problem is embedded in an operator-split multi-dimensional problem, boundary conditions schemes are often applied separately from the interior-point advancement. Such schemes are usually implemented by first updating the ghost-point values u_1 and u_n using Equation 3 to ensure that the boundary conditions are satisfied, meanwhile keeping the interior-point values constant. Subsequently, the interior-point values are updated (in this case by using Equation 2), but now the ghost-point values are kept unchanged. If we use the tilde symbol '~' to indicate the intermediate ghost-point values after application of the boundary conditions, we can write the above scheme as a two-step semi-implicit procedure:

Step 1:

$$\begin{pmatrix} \widetilde{u_1} \\ u_2 \\ \vdots \\ u_{n-1} \\ u_n \end{pmatrix}^{k+1} = \begin{pmatrix} 0 & -1 & & & \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & \\ & & & -1 & 0 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_{n-1} \\ u_n \end{pmatrix}^{k} + 2 \begin{pmatrix} u_a \\ 0 \\ \vdots \\ 0 \\ u_b \end{pmatrix} \tag{4}$$

Step 2:

$$\begin{pmatrix} 1 & & & & \\ 1+\frac{Re}{2} & -2 & 1-\frac{Re}{2} & & \\ & & \ddots & & \\ & & 1+\frac{Re}{2} & -2 & 1-\frac{Re}{2} \\ & & & & 1 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_{n-1} \\ u_n \end{pmatrix}^{k+1} = \begin{pmatrix} 1 & & & \\ & 0 & & \\ & & \ddots & \\ & & & 0 \\ & & & & 1 \end{pmatrix} \begin{pmatrix} \widetilde{u_1} \\ u_2 \\ \vdots \\ u_{n-1} \\ u_n \end{pmatrix}^{k+1} \tag{5}$$

Combining Equations 4 and 5, we obtain a single-step iterative scheme whose iteration matrix N is given by:

$$N = \begin{pmatrix} 1 & & & & \\ 1+\frac{Re}{2} & -2 & 1-\frac{Re}{2} & & \\ & \ddots & & & \\ & 1+\frac{Re}{2} & -2 & 1-\frac{Re}{2} & \\ & & & 1 & \end{pmatrix}^{-1} \begin{pmatrix} 0 & -1 & & & \\ & 0 & & & \\ & & \ddots & & \\ & & & 0 & \\ & & & -1 & 0 \end{pmatrix} \tag{6}$$

The eigenvalues of N determine the convergence rate of the scheme. These are found by setting $N\mathbf{q} = \lambda\mathbf{q}$:

$$\begin{aligned} -q_2 &= \lambda q_1 \,, \\ 0 &= \lambda \left\{ (1 + Re_c/2) q_{i-1} - 2q_i + (1 - Re_c/2) q_{i+1} \right\} \,, \; i = 2, \ldots, n-1 \,, \\ -q_{n-1} &= \lambda q_n \,. \end{aligned} \tag{7}$$

The solution to this set consists of two distinct families. The first one has $n-2$ eigenvectors that all correspond to an eigenvalue of zero. This eigenspace is spanned by the $n-2$ basis vectors: $span\{\mathbf{e}_1, \mathbf{e}_3, \mathbf{e}_4, \ldots, \mathbf{e}_{n-3}, \mathbf{e}_{n-2}, \mathbf{e}_n\}$. The second—non-trivial—family has nonzero eigenvalues, which determine the actual convergence of the scheme. The general solution to the equations for the eigenvectors at interior grid points is of the form

$$q_i = \alpha + \beta \tau^i \,, \quad \text{with} \;\; \tau = \frac{1 + \frac{Re}{2}}{1 - \frac{Re}{2}} \,. \tag{8}$$

The constants α and β are determined by applying the two 'boundary conditions' of Equation 7, which are written in matrix form:

$$\begin{pmatrix} \lambda + 1 & \lambda\tau + \tau^2 \\ \lambda + 1 & \lambda\tau^n + \tau^{n-1} \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = 0 \,. \tag{9}$$

This equation only has nontrivial solutions if the determinant of the matrix on the left hand side is zero, which yields the following eigenvalues:

$$\lambda_{1,2} = -1, \frac{\tau - \tau^{n-2}}{\tau^{n-1} - 1} \,. \tag{10}$$

Weak convection

In case of weak convection we can write $\tau \approx 1 + Re_c$ and $\tau^k \approx 1 + kRe_c$, so

$$\lambda_{1,2} = -1, \frac{3 - n}{n - 1} + \mathcal{O}(Re_c^2) \,. \tag{11}$$

Apparently, the effect of convection on the rate of convergence of the scheme vanishes (to first order) for small values of the cell Reynolds number. Since $n \geq 3$ for a finite-volume scheme, λ_2 is always negative, and smaller in absolute value than 1. But λ_2 approaches -1 for large n, and $\lambda_1 = -1$, so the method will never converge, although the interior points are updated using an implicit scheme. The lack of convergence is due to the boundary-value update scheme. The fact that the eigenvalues of the iterative scheme are both (close to) -1 suggests, however, that underrelaxation may help.

Consequently, we replace the boundary value update Equation 4 by:

$$\begin{pmatrix} \widetilde{u_1} \\ u_2 \\ \vdots \\ u_{n-1} \\ u_n \end{pmatrix}^{k+1} = \begin{pmatrix} \theta & \theta - 1 & & & \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & \\ & & & \theta - 1 & \theta \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_{n-1} \\ u_n \end{pmatrix}^{k} + 2(1 - \theta) \begin{pmatrix} u_a \\ 0 \\ \vdots \\ 0 \\ u_b \end{pmatrix} \tag{12}$$

All other steps in the determination of the eigenvalues of the iteration matrix stay the same, and we now find:

$$\lambda_{1,2} \approx (2\theta - 1), \frac{(2\theta - 1)n + 3 - 4\theta}{n - 1}. \tag{13}$$

Minimizing the spectral radius of the iteration matrix yields:

$$\theta = \frac{2 - n}{3 - 2n} \quad \rightarrow \rho(N) \approx \frac{1}{3 - 2n}. \tag{14}$$

Obviously, excellent convergence is obtained with this choice of the underrelaxation factor θ. For large grid sizes the value of $1/2$ for θ is completely satisfactory.

Strong convection

The situation changes rather dramatically in case the convection is strong (large Re_c). We introduce the small parameter $\epsilon = 1/Re_c$. Again, the first non-trivial eigenvalue of N is -1. But the value of λ_2 depends on the parity of n:

$$\lambda_2 = 1 - 8(n-2)\epsilon^2 + \mathcal{O}(\epsilon^3) \quad \text{for } n \text{ even} \tag{15}$$

$$= \frac{n-3}{n-1} + \mathcal{O}(\epsilon^2) \quad \text{for } n \text{ odd} \tag{16}$$

Regardless of the parity of n, however, λ_2 approaches 1 for large grid sizes as the cell Reynolds number goes to infinity, which indicates that underrelaxation will not speed up convergence. Indeed, if we underrelax the boundary conditions, we find:

$$\lambda_{1,2} \approx (2\theta - 1), \;\; 1 \,\text{or}\, \frac{n-3}{n-1}(1-\theta) + \theta. \tag{17}$$

It follows that λ_2 is always close to or equal to 1, independent of the value of θ.

The situation may be remedied by using a more physically relevant boundary condition at the downstream boundary of the domain. Assuming a positive convection speed, we replace the Dirichlet boundary condition at $x = b$ by a Neumann condition. A second-order-accurate implementation of the condition $\frac{du}{dx}|_{x=b} = 0$ is obtained by setting:

$$\tilde{u}_n^{k+1} = u_{n-1}^k. \tag{18}$$

Underrelaxing the Neumann and Dirichlet conditions using θ_N and θ_D, respectively, produces the following system:

$$\begin{pmatrix} \widetilde{u_1} \\ u_2 \\ \vdots \\ u_{n-1} \\ u_n \end{pmatrix}^{k+1} = \begin{pmatrix} \theta_D & \theta_D - 1 & & & \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & \\ & & & 1-\theta_N & \theta_N \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_{n-1} \\ u_n \end{pmatrix}^k + 2(1-\theta_D) \begin{pmatrix} u_a \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}. \tag{19}$$

We repeat the eigenvalue/eigenvector analysis carried out above. The components of the non-trivial eigenvector $\mathbf{q}$ can again be written as $q_i = \alpha + \beta\tau^i$, but now the definition of τ is changed to: $\tau = (2\epsilon - 1)/(2\epsilon + 1)$. Notice that now $\lim_{\epsilon \downarrow 0} \tau = -1$ (large cell Reynolds number), whereas before it was 1 (small cell Reynolds number). The small (2×2)-determinant equation producing the non-trivial eigenvalues of the iteration matrix (cf. Equation 9) becomes:

$$\begin{vmatrix} \lambda - 1 & \lambda\tau - \tau^2 + \theta_N(\tau^2 - \tau) \\ \lambda + 1 - 2\theta_D & \lambda\tau^n + \tau^{n-1} - \theta_D(\tau^n + \tau^{n-1}) \end{vmatrix} = 0\,. \tag{20}$$

The expressions for the roots $\lambda_{1,2}$ are not very insightful in general. We investigate the case where the cell Reynolds number approaches infinity, so τ approaches -1.

For an even number of grid points (n), Equation 20 reduces to:

$$2\lambda^2 - (\theta_D + \theta_N)\lambda + 1 + (1 - 2\theta_D)(1 - 2\theta_N) = 0\,. \tag{21}$$

The roots of this equation are both zero (and hence produce a spectral radius equal to zero) for $\theta_D = \pm 1/\sqrt{2}\,, \theta_N = -\theta_D$. Notice, however, that a negative value of a relaxation factor corresponds to an overrelaxation (i.e. an extrapolation) of a boundary value, which is prone to instability. If we limit the relaxation factors to positive values and set θ_N equal to zero, we find an optimal spectral radius of $\rho(N) = \sqrt{2} - 1 (\approx 0.4)$ by choosing $\theta_D = 2(\sqrt{2} - 1)$.

Unfortunately, the picture is not so rosy in case the number of grid points is odd. Equation 20 becomes singularly perturbed for large cell Reynolds numbers (coefficient of λ^2 disappears), and we have to take limits carefully. Whereas the root corresponding to the reduced equation ($Re_c \equiv \infty$) is given by

$$\lambda = \frac{1 - (1 - 2\theta_N)(1 - 2\theta_D)}{4 - 2(\theta_N + \theta_D)}\,, \tag{22}$$

which can always be made to equal zero, the other root is asymptotically proportional to the cell Reynolds number, irrespective of the choice for the relaxation factors; the scheme always diverges for large Re_c.

The solution to the problem—not too surprisingly—is to use upwind differencing in case of strong convection. The resulting iteration matrix for the Neumann/Dirichlet condition is:

$$N = \begin{pmatrix} 1 & & & \\ 1 + Re_c & -2 - Re_c & 1 & \\ & \ddots & & \\ & 1 + Re_c & -2 - Re_c & 1 \\ & & & 1 \end{pmatrix}^{-1} \begin{pmatrix} \theta_D & \theta_D - 1 & & \\ & 0 & & \\ & & \ddots & \\ & & & 0 \\ & & 1 - \theta_N & \theta_N \end{pmatrix} \tag{23}$$

Equation 20 is still valid, provided τ is defined by $1 + Re_c$. For large cell Reynolds numbers it reduces to:

$$(2\theta_D - \lambda - 1)(\theta_N - \lambda) = 0\,, \tag{24}$$

so a spectral radius of approximately zero can be obtained by setting $\theta_D = 1/2$ and $\theta_N = 0$. For small cell Reynolds numbers the eigenvalue equation is:

$$(n-1)\lambda^2 + (2\theta_D(1-n) + 2 - \theta_N)\lambda + (2\theta_D - 1)(n - 2 + \theta_N) - 1 = 0\,. \tag{25}$$

For large n (many grid points) the spectral radius has the form: $\rho(N) = 1 - \mathcal{O}(1/n)$; the convergence deteriorates for vanishing convection and large grid sizes, and the Neumann outflow boundary condition should not be used. Reverting to Dirichlet conditions, underrelaxed by 50%, restores the excellent convergence also obtained when using central differencing.

We now combine the previous analyses into a rule of thumb:

Apply Dirichlet boundary conditions on inflow boundaries and always underrelax by 50%. In case of strong convection, use upwind differencing and apply Neumann boundary conditions on outflow boundaries, but do not underrelax these conditions.

3 Time-dependent, scalar 1-D model equation

We now investigate the convergence properties of finite-volume formulations of the linearized, time-dependent, one-dimensional convection/diffusion equation:

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = \mu\frac{\partial^2 u}{\partial x^2}\,. \tag{26}$$

We apply the Euler-implicit method for the time integration (k-superscript) and central differencing in space (i-subscript) at all interior grid points.

$$(1 + Re_c/2)u_{i-1}^{k+1} - (2+\nu)u_i^{k+1} + (1 - Re_c/2)u_{i+1}^{k+1} = -\nu u_i^k\,, \quad \text{with} \tag{27}$$

$$\mathrm{CFL} \stackrel{\mathrm{def}}{=} \frac{c\Delta t}{\Delta x}\,, \quad \nu \stackrel{\mathrm{def}}{=} \frac{Re_c}{\mathrm{CFL}} = \frac{\Delta x^2}{\mu \Delta t}\,.$$

Again, boundary values are updated as in Equation 4, where k now signifies the time step instead of the iteration number. Advancement of the interior-point values takes place through solution of Equation 27. The resulting two-step scheme can again be combined into a single equation, whose amplification matrix is given by (cf. Equation 6):

$$N = \begin{pmatrix} 1 & & & & \\ 1+\frac{Re}{2} & -2-\nu & 1-\frac{Re}{2} & & \\ & \ddots & & & \\ & 1+\frac{Re}{2} & -2-\nu & 1-\frac{Re}{2} & \\ & & & 1 & \end{pmatrix}^{-1} \begin{pmatrix} 0 & -1 & & & \\ & -\nu & & & \\ & & \ddots & & \\ & & & -\nu & \\ & & & -1 & 0 \end{pmatrix} \tag{28}$$

Explicit expressions for the eigenvalues of N cannot be obtained in general, but some limiting cases are easily identified.

$$\lim_{\nu\to+\infty} N = 1\,, \quad \lim_{\nu\downarrow 0} N = 1\,. \tag{29}$$

Whereas the first limit is expected and acceptable—for infinitely small time steps convergence will be infinitely slow as well—the second is highly undesirable, since it implies that very large time steps will lead to very slow convergence as well. This is an artifact of the application of the boundary conditions of the finite-volume scheme, and we attempt to solve the convergence problem in the same vein as before.

Weak convection

In Figure 2 the effect of underrelaxing Dirichlet boundary conditions for a weakly convective case ($Re_c = 2$) on a small grid ($n = 40$) is depicted. The cell Reynolds number is chosen such that the discretization matrix for the interior point scheme is guaranteed to be diagonally dominant for all finite sizes of the time step, a property often judged desirable in fluid flow computations. The dashed and solid lines indicate the amplification factor in the absence of relaxation and with 50% underrelaxation, respectively. For curiosity's sake we also plot the amplification factor corresponding to 30% underrelaxation (dotted line).

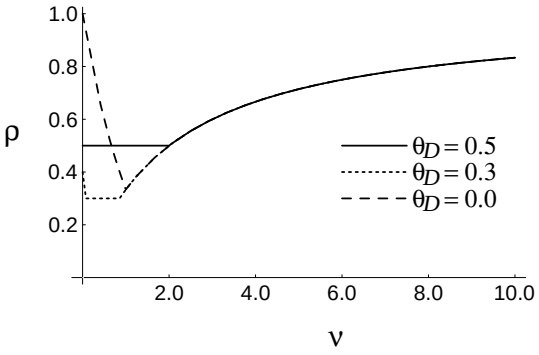


Figure 2: Amplification factor for underrelaxed Dirichlet boundary conditions; weak convection ($Re_c = 2$)

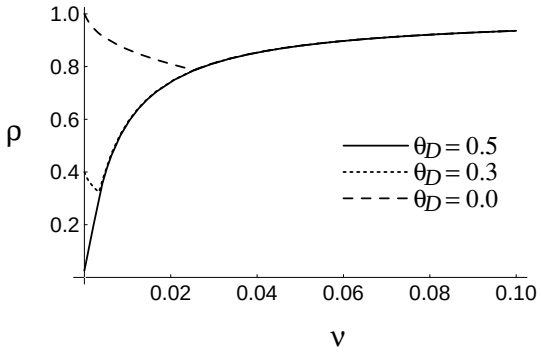


Figure 3: Amplification factor for underrelaxed Dirichlet boundary conditions; no convection ($Re_c = 0$)

Notice that all three curves collapse onto the same branch for larger values of ν, i.e. as the time step goes to zero. The improvement through the 50% underrelaxation for small values of ν is apparent. The fact that the amplification factor does not decrease strictly monotonically as ν approaches zero is due to the presence of convection. As the cell Reynolds number vanishes, the underrelaxed amplification factor becomes strictly monotone and reduces to zero for infinite time step (see Figure 3). The curve corresponding to the nonrelaxed boundary conditions, however, remains non-monotone.

If we underrelax the mildly convective case between 50% and 30% , the intersection point with the vertical axis (amplification factor for infinite time step) does not change much, although the minimum amplification factor for nonzero ν may decrease. For relaxation factors outside the above range the intersection point actually moves up, and gains are diminished.

It is also apparent from Figures 2 and 3 that even underrelaxation by 50% does not completely save the difference scheme from deterioration in the case of total absence of convection. Only

for infinite time step does the amplification factor go to zero, but for finite time step the factor approaches 1 ever more rapidly as the cell Reynolds number drops.

Strong convection

As in the steady analysis, we apply upwind differencing and Neumann boundary conditions at the outflow boundary for the strongly convective case, and modify Equation 28 accordingly. Now it is most useful to study the behavior of the difference scheme for a certain fixed CFL number. We first investigate the case of infinitely strong convection. The outflow boundary condition cannot be enforced anymore since the spatial differencing reduces in the limit to a one-sided difference, so the set of equations determining the eigenvalues of the amplification matrix becomes (cf. Equation 7):

$$\begin{aligned} -x_2 &= \lambda x_1 \,, \\ -x_i/\mathrm{CFL} &= \lambda \left(x_{i-1} - (1 + 1/\mathrm{CFL}) x_i \right) \,. \end{aligned} \tag{30}$$

Solving the recursive interior-point equation and substituting the boundary condition yields the following asymptotic result for large CFL numbers: $\rho(N) = 1 - 2/\mathrm{CFL}$. Applying underrelaxation of the Dirichlet (upstream) boundary condition gives for large values of the CFL number:

$$\begin{aligned} \rho(N) &= \frac{1}{\sqrt{2\mathrm{CFL}}} + \mathcal{O}\left(\frac{1}{\mathrm{CFL}}\right) \,, \quad \text{for } \theta_D = \frac{1}{2} \,, \\ &= 2\theta_D - 1 + \mathcal{O}\left(\frac{1}{\mathrm{CFL}}\right) \,, \quad \text{for } \theta_D \neq \frac{1}{2} \,. \end{aligned} \tag{31}$$

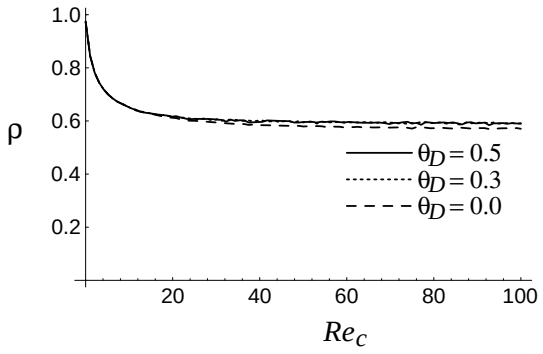


Figure 4: Amplification factor for underrelaxed Dirichlet boundary conditions; small CFL number (CFL = 1)

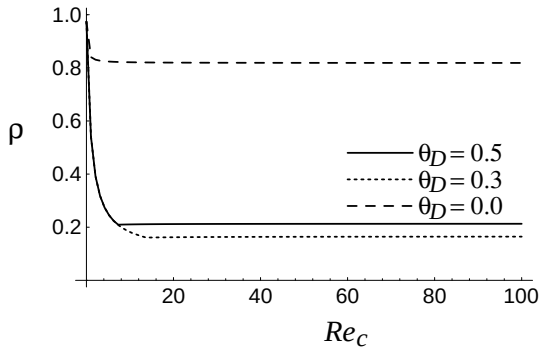


Figure 5: Amplification factor for underrelaxed Dirichlet boundary conditions; medium CFL number (CFL = 10)

Obviously, $\theta_D = 1/2$ is again a good choice. For cell Reynolds numbers of finite magnitude we apply the same underrelaxation of the Dirichlet boundary condition, and leave the Neumann boundary condition again untouched. The results for varying values of the CFL number are show in Figures 4, 5, and 6, together with the amplification factors for no underrelaxation of the Dirichlet condition (dashed line), and underrelaxation of 30%. Although $\theta_D = 1/2, \theta_N = 0$ is not always the optimal choice, it is always good, asymptotically best (large CFL), and easy to apply.

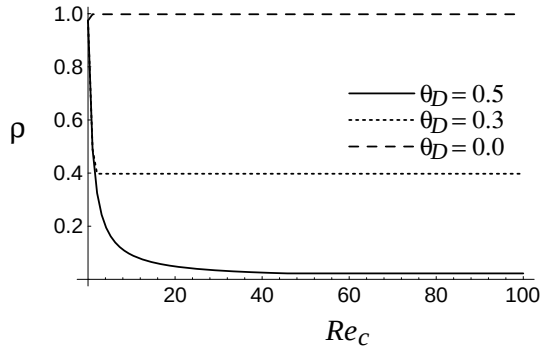


Figure 6: Amplification factor for underrelaxed Dirichlet boundary conditions; large CFL number (CFL = 1000)

4 Unsteady, scalar 2-D model equation

We now extend the analysis to a two-dimensional, linear model equation for the convection/diffusion process.

$$\frac{\partial u}{\partial t} + c_x \frac{\partial u}{\partial x} + c_y \frac{\partial u}{\partial y} = \mu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) . \tag{32}$$

This problem is solved on a square mesh of $(n-2) \times (n-2)$ interior points The grid is depicted in Figure 7, including the ghost points (crosses) that lie outside the grid boundaries. We also employ the ghost points on the corners of the exterior grid (open squares) to arrive at a square total number of points. But since they never enter the discretization scheme or the solution, we keep the values at these corner ghost points fixed. We apply central differencing to all spatial derivatives (δ_{x_i} denotes the central difference operator in the x_i direction), and backward differencing (Euler Implicit) to the time derivative to arrive at the following discretization.

$$\left[I + \mathrm{CFL}_x \left(\delta_x - \frac{1}{Re_{c_x}} \delta_x^2 \right) + \mathrm{CFL}_y \left(\delta_y - \frac{1}{Re_{c_y}} \delta_y^2 \right) \right] u^{k+1} = \widetilde{u^{k+1}} . \tag{33}$$

Here we have defined: $Re_{c_{x_i}} = \frac{c_{x_i} \Delta x_i}{\mu}$ and $\mathrm{CFL}_{x_i} = \frac{c_{x_i} \Delta t}{\Delta x_i}$. If one or both of the convection speeds c_{x_i} are zero, equation 33 is simplified accordingly. As in the one-dimensional finite-volume case, we update ghost point values such that the average of every couple of values at grid points straddling the boundary equals the Dirichlet boundary value, or symbolically: $u_\bullet + u_\times = u_b$, where the subscript b stands for 'boundary'. Application of the ghost point value update operator Ω, which is the direct analogue of the one-dimensional operator, gives the following time advancement scheme.

$$(I + L_x + L_y) u^{k+1} = \Omega u^k . \tag{34}$$

Note the introduction of L_x and L_y as shorthand for the spatial discretization operators. From equation 34 we derive two additional schemes through ADI approximate factorization. Under the assumption that L_x and L_y are small compared to I, a reasonable approximation is:

$$(I + L_x)(I + L_y) u^{k+1} = \Omega u^k . \tag{35}$$

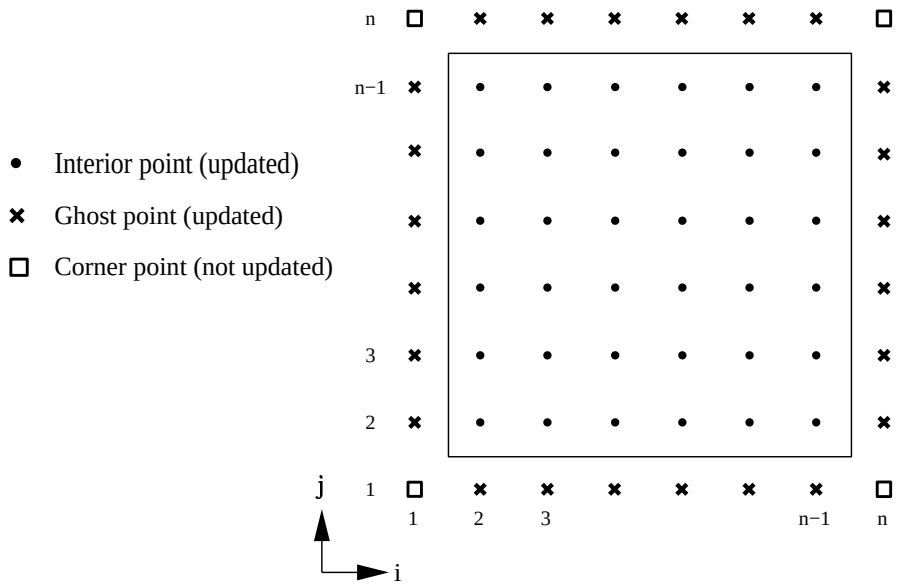


Figure 7: Two-dimensional finite-volume grid

For large Δt, however, this assumption is not valid, and we have to adopt the so-called delta form of the ADI method. This is derived from equation 33 by introducing the update quantity $\Delta u^k = u^{k+1} - \Omega u^k$, and by applying the approximate factorization only to the implicit part ('left hand side') of the operator that multiplies Δu^k. The resulting delta form can be combined with the non-delta form into a single equation.

$$(I + L_x)(I + L_y)u^{k+1} = (I + (1 - \alpha)L_x L_y)\Omega u^k \,. \tag{36}$$

If α equals zero we obtain the delta form, whose final solution is independent of the size of the time step. For any other α, the solution at convergence depends on the size of the time step, due to the factorization error, which is proportional to $\alpha L_x L_y$. Algebraic analysis of the spectral radius of the iteration matrix corresponding to equation 36 is too complex, so we resort to numerical investigation instead. Figure 8 shows the spectral radius of the iteration matrix on a grid of 40×40 points for the non-delta form of the ADI algorithm, and 9 for the delta form. Since the direct-inversion scheme does not incur a factorization error, its delta and non-delta forms are identical. From Figure 8 we deduce that the non-delta forms benefit significantly from underrelaxation of the boundary conditions for large time steps (small ν). In fact, the ADI scheme converges as rapidly as the direct-inversion scheme. But if we insist on a solution independent of the size of the time step, the delta form of the ADI algorithm must be used, which gains virtually nothing from underrelaxation (see Figure 9, where the curves for ADI with and without boundary update relaxation coincide). Since direct inversion is not practical for many calculations of engineering value, we must find ways of speeding up the ADI algorithm without giving up solution accuracy at convergence.

One approach to accomplishing this is to let the switch parameter α in the hybrid scheme represented by equation 36 slide from one to zero during the course of the iteration process. We use the fact that the convergence factor for the underrelaxed hybrid scheme behaves approximately as

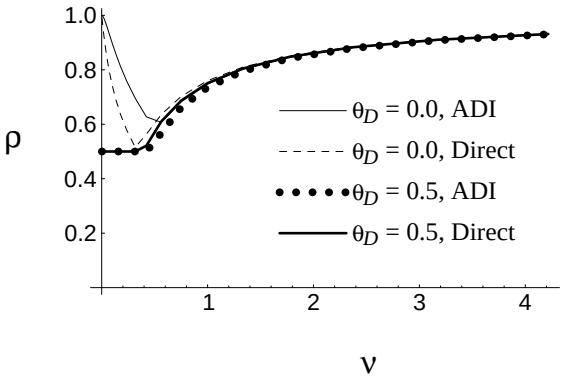


Figure 8: Amplification factor for non-delta form ($\alpha = 1$); no convection

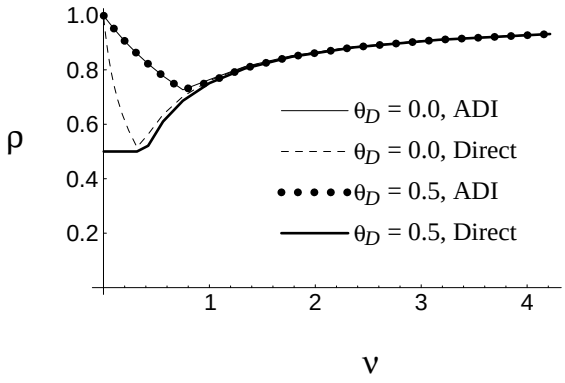


Figure 9: Amplification factor for delta form ($\alpha = 0$); no convection

follows for large time steps and for small α.

$$\rho(\alpha) \approx 1 - s\alpha\,, \tag{37}$$

where s is a positive constant. This relationship is depicted in Figure 10, where the computed spectral radius for the iteration matrix belonging to the 50% underrelaxed ADI scheme on the 40×40 grid is shown as a function of the switch parameter. The non-dimensional time step is set at an arbitrarily large value of 1000. Note that keeping α constant at some value other than zero will always speed up the convergence, but at the cost of final solution accuracy. We now analyze the convergence of the entire iterative process for some choice of the way in which α goes to zero, i.e. $\lim_{k\to\infty} \alpha_k = 0$, where k is the number of the iteration or time step. Since the convergence is faster

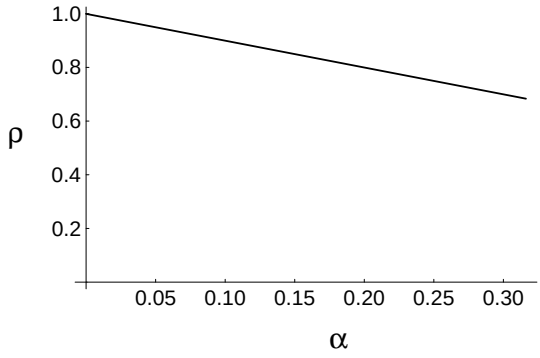


Figure 10: Amplification factor for hybrid ADI scheme; no convection

for values of α away from 0, we select a fairly slowly decreasing function, namely $\alpha = z/(k+1)$, where z is some positive constant. Without loss of generality, we can set z equal to the computationally convenient $1/s$. Consequently, we obtain $\rho_k \stackrel{\text{def}}{=} \rho(\alpha_k) \approx 1 - 1/(k+1)$. The 'convergence' factor for p steps of the hybrid iterative scheme is:

$$\prod_{k=1,p} \rho_k \approx \prod_{k=1,p} \left(\frac{k}{k+1}\right) = \frac{1}{p+1}\,. \tag{38}$$

Evidently, the scheme is converging very slowly, since after p steps the compound convergence factor is only $1/(p+1)$. The *error* of the scheme, defined by $e^p = u^\infty - u^p$, is estimated as follows. At convergence, the solution u^∞ satisfies equation 34, i.e.,

$$(I + L_x + L_y)u^\infty = \Omega u^\infty \,. \tag{39}$$

Subtracting equation 36 from equation 39 yields the following expression:

$$\begin{aligned} e^{p+1} &= \left[(I+L_x)(I+L_y)\right]^{-1}\left[(I+(1-\alpha_p)L_xL_y\right]\Omega e^p + \\ &\quad \alpha_p\left[(I+L_x)(I+L_y)\right]^{-1} L_xL_y\Omega u^\infty \\ &= R(\alpha_p)e^p + \alpha_p t^\infty \\ &= \prod_{k=1}^{p} R(\alpha_k)e^1 + \left(\sum_{k=1}^{p}\alpha_{p-k+1}\prod_{i=1}^{p-k}R(\alpha_i)\right)t^\infty \,, \end{aligned} \tag{40}$$

where R is the iteration matrix of the scheme defined by equation 36, and t^∞ is a quantity dependent only on the converged solution. The norm of the error can be bounded as follows:

$$\|e^{p+1}\| \le \prod_{k=1}^{p}\rho_k\|e^1\| + \left(\sum_{k=1}^{p}\alpha_{p-k+1}\prod_{i=1}^{p-k}\rho_i\right)\|t^\infty\| \,. \tag{41}$$

Consequently, the solution error generally decreases at least as slowly as indicated by the compound convergence factor (equation 38). For the choice of $\alpha_k = 1/(s(k+1))$, we find

$$\|e^{p+1}\| \le \frac{\|e^1\|}{p+1} + \sum_{k=1}^{p}\frac{1}{k^2}\frac{\|t^\infty\|}{s} \,, \tag{42}$$

which leads to the limiting result:

$$\lim_{p\to\infty}\|e^{p+1}\| \le \frac{\pi^2\|t^\infty\|}{6s} \,. \tag{43}$$

This means that the scheme never converges to the correct solution, which appears to contradict the fact that the compound convergence factor does approach zero.. But the computed convergence factors apply to an ever-changing discretization scheme, due to the 'hybridization', and no conclusions can be drawn about the eventual vanishing of their product per se. We note that even in the case of a fairly slowly changing α, the scheme stalls. If we make α approach 0 more slowly, the final solution error will be even larger. If more quickly, then convergence will be even slower. It thus follows that the hybrid scheme does not have a fundamentally better convergence behavior than the plain delta form of the ADI scheme, and relaxation of the boundary condition application will not improve its performance.

5 Conclusions

We have shown that it is possible to overcome the problems associated with the application of time-lagged boundary value updates for cell-centered finite-volume schemes for simple one-dimensional scalar, linear model equations. The appropriate technique relies on underrelaxing the boundary value updates to annihilate or reduce oscillations. In two dimensions the same technique works well for an approximately factored scheme, but the factorization error renders the solution useless for large time steps. When the delta form of the scheme is used to reduce the factorization error, convergence becomes very slow, even for a diffusion-dominated flow, and using underrelaxation of boundary value updates is ineffective.

References

[1] Beam, R. and Warming, R.; "An Implicit Factored Scheme for the Compressible Navier-Stokes Equations", AIAA Journal, Vol. 16, 1978, pp. 393-402.

[2] Flores, J., and Chaderjian, N. M.; "Zonal Navier-Stokes Methodology for Flow Simulations about Complete Aircraft", Journal of Spacecraft and Rockets, Vol. 27, No. 7, 1990, pp. 583-590.

[3] Frink, N.; Private Communication, 1996.

[4] Hall, M. G.; "Cell Vertex Multigrid Scheme for Solution of the Euler Equations", Proceedings of the Conference on Numerical Methods in Fluids, Vol. 5, 1985, pp. 483-500.

[5] Jameson, A., Schmidt, W., and Turkel, E.; "Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time Stepping Schemes", AIAA-Paper 81-1259, 1981.

[6] Jameson, A. and Baker, T. J.; "Solution of the Euler Equations for Complex Configurations", AIAA-Paper 83-1929, 1983.

[7] Klopfer, G. H. and Molvik, G. A.; "Conservative Multizonal Interface Algorithm for the 3-D Navier-Stokes Equations", AIAA Paper 91-1601-CP, 1991.

[8] Ni, R. H.; "Multiple Grid Scheme for Solving the Euler Equations", AIAA-Paper 81-1025, 1981.

[9] Rizzi, A.; "Damped Euler Equation Method to Compute Transonic Flow Around Wing-Body Combinations", AIAA Journal, Vol. 20, 1982, pp. 1321-1328.

[10] Roe, P. L.: "Error Estimates for Cell-Vertex Solutions of the Compressible Euler Equations", ICASE Report No. 87-6, 1987.

[11] Rossow, C., Kroll, N., Radespiel, R., and Scherr, S.; "Investigation of the Accuracy of Finite Volume Methods for 2- and 3-Dimensional Flows", AGARD-CPP-437, Validation of Computational Fluid Dynamics, 1988, pp. p17-1,p17-11.

[12] Steger, J. L.; "Implicit Finite-Difference Simulation of Flow about Arbitrary Two-Dimensional Geometries," AIAA Journal, Vol. 16, No. 7, 1978, pp. 679-686.

[13] Turkel, E.; "Accuracy of Schemes with Nonuniform Meshes for Compressible Fluid Flows", ICASE Report No. 85-43, 1985.

[14] Turkel, E.; "Accuracy of Schemes for the Euler Equations with Nonuniform Meshes", ICASE Report No. 85-59, 1985.

[15] Yoon, S. and Jameson, A.; "Lower-Upper Symmetric Gauss Seidel Method for the Euler and Navier-Stokes Equations", AIAA Journal, Vol. 26, 1987, pp. 1025-1026.