

Chapter 5

An Explicit Finite-Volume Algorithm with Multigrid

5.1 Introduction

The salient features of the algorithm presented in this chapter are as follows (the reader is urged to contrast these with the key characteristics of the algorithm presented in Chap. 4, which are listed in Sect. 4.1):

- cell-centered data storage; the numerical solution for the state variables is associated with the cells of the grid
- second-order finite-volume spatial discretization with added numerical dissipation; a simple shock-capturing device
- applicable to structured grids (see Sect. 4.2)
- explicit multi-stage time marching with implicit residual smoothing and multigrid

Key contributions to the development of this algorithm were made by Jameson et al. [1], Baker et al. [2], Jameson and Baker [3], Jameson [4, 5], and Swanson and Turkel [6, 7]. The reader is referred to Swanson and Turkel [7] for further analysis and description of the algorithm.

The exercises at the end of this chapter again provide an opportunity to apply the algorithm presented to several one-dimensional problems.

5.2 Spatial Discretization: Cell-Centered Finite-Volume Method

The cell-centered approach contrasts with the node-centered approach described in Chap. 4. The meshes described thus far are known as *primary* meshes. One can also construct a *dual* mesh by joining the centroids of the cells associated with the primary mesh. In the case of a two-dimensional structured mesh, the dual mesh also consists of quadrilaterals and is qualitatively similar to the primary mesh. For more general unstructured meshes this is not the case. For example, for a primary mesh consisting of regular triangles the dual mesh consists of hexagons. A scheme that is

cell centered on the primary mesh can be considered to be node centered on the dual mesh. Hence, in the case of quadrilateral structured meshes, the cell-centered nature has little impact on the spatial discretization in the interior, and both cell-centered and node-centered finite-volume schemes are in common use on both structured and unstructured meshes. The main differences between the two arise at boundaries and in the construction of coarse meshes for multigrid. This will be discussed further below.

A finite-volume method numerically solves the governing equations in integral form, as presented in Sect. 3.1.2. In their most general coordinate-free form, conservation laws can be written as

$$\frac{d}{dt} \int_{V(t)} Q dV + \oint_{S(t)} \hat{n} \cdot \mathcal{F} dS = \int_{V(t)} P dV, \quad (5.1)$$

where P is a source term, and the other variables are defined in Chap. 3. If we restrict our interest to two-dimensional problems without source terms and meshes that are static with respect to time, we obtain

$$\frac{d}{dt} \int_A Q dA + \oint_C \hat{n} \cdot \mathcal{F} dl = 0, \quad (5.2)$$

where A is a control volume bounded by a contour C . Writing the flux tensor \mathcal{F} in Cartesian coordinates and separating inviscid and viscous fluxes gives

$$\frac{d}{dt} \int_A Q dA + \oint_C \hat{n} \cdot (E\hat{i} + F\hat{j}) dl = \oint_C \hat{n} \cdot (E_v\hat{i} + F_v\hat{j}) dl. \quad (5.3)$$

Finally, writing the product of the outward normal and the length of the cell edge in Cartesian coordinates as

$$\hat{n} dl = dy\hat{i} - dx\hat{j} \quad (5.4)$$

gives the final form to be discretized using the finite volume method:

$$\frac{d}{dt} \int_A Q dA + \oint_C (E dy - F dx) = \oint_C (E_v dy - F_v dx). \quad (5.5)$$

The semi-discrete form of (5.5) is written as

$$A_{j,k} \frac{d}{dt} Q_{j,k} + \mathcal{L}_i Q_{j,k} + \mathcal{L}_{ad} Q_{j,k} = \mathcal{L}_v Q_{j,k}, \quad (5.6)$$

where $A_{j,k}$ is the area of the cell, \mathcal{L}_i is the discrete approximation to the inviscid flux integral, \mathcal{L}_{ad} is the artificial dissipation operator, \mathcal{L}_v is the discrete approximation to the viscous flux integral, and $Q_{j,k}$ denotes the conservative variables averaged over cell j, k as follows:

$$Q_{j,k} = \frac{1}{A_{j,k}} \int_{A_{j,k}} Q dA. \quad (5.7)$$

The terms $\mathcal{L}_i Q$, $\mathcal{L}_v Q$, and $\mathcal{L}_{ad} Q$ are described next.

5.2.1 Inviscid and Viscous Fluxes

The inviscid flux integral is approximated by summing over the four edges of the cell as follows:

$$\mathcal{L}_i Q = \sum_{l=1}^4 (\mathcal{F}_i)_l \cdot \mathbf{s}_l, \quad (5.8)$$

where

$$\mathbf{s}_l = (\Delta y)_l \hat{i} - (\Delta x)_l \hat{j}. \quad (5.9)$$

is the discrete analog of (5.4) for straight cell edges, and $(\mathcal{F}_i)_l$ is an approximation to the inviscid flux tensor at the cell edge. We use boldface to emphasize that \mathbf{s}_l is a vector. The terms $(\Delta x)_l$ and $(\Delta y)_l$ must be defined such that the normal vector points out of the cell. Since the cell edges are straight, the outward normal is constant along each edge. The only exception might arise at the body surface; there the approximation of the edge as straight is adequate for a second-order discretization, but the curvature of the boundary must be taken into account if higher-order accuracy is desired.

In Sect. 2.4.2, we saw that the combination of a piecewise constant reconstruction with a simple average for resolving the discontinuity in fluxes at cell interfaces leads to a second-order centered finite-volume scheme that is analogous to a second-order centered finite-difference scheme on a uniform mesh. The same approach is taken here. With the minus sign superscript defining quantities in the cell on one side of the interface and the plus sign indicating the other side, the averaged flux on a given cell edge is given by

$$(\mathcal{F}_i)_l = \frac{1}{2}(\mathcal{F}_i^- + \mathcal{F}_i^+) = \frac{1}{2}(Q^- \mathbf{v}^- + Q^+ \mathbf{v}^+)_l + \bar{P}_l, \quad (5.10)$$

where $\mathbf{v} = u\hat{i} + v\hat{j}$, and

$$\bar{P}_l = [0, \frac{1}{2}(p^- + p^+)_l \hat{i}, \frac{1}{2}(p^- + p^+)_l \hat{j}, \frac{1}{2}(p^- \mathbf{v}^- + p^+ \mathbf{v}^+)_l]^T. \quad (5.11)$$

This scheme is second order and nondissipative. Numerical dissipation must be added, as described in Sect. 5.2.2.

For the viscous terms, we turn again to Sect. 2.4.2. In that section, a second-order finite-volume scheme was derived for the diffusion equation using two different approaches. The first approach is based on the use of a one-dimensional version of (2.58), which is given by

$$\int_A \nabla Q dA = \oint_C \hat{n} Q dl. \quad (5.12)$$

This approach is simple to extend to multidimensions but is restricted to second-order accuracy. Given that we seek a second-order approximation, we will follow this approach to obtain a discretization for the viscous flux terms.

The difficulty associated with the viscous fluxes is that they include velocity gradients, and these cannot be obtained directly from the solution vector. In order to obtain a suitable approximation to the velocity gradients at the cell edges, (5.12) is applied to auxiliary cells that surround each edge of the cell in question. When applied to the Cartesian velocity components, (5.12) gives the components of the velocity gradient as follows:

$$\begin{aligned} \int_{A'} \frac{\partial u}{\partial x} dA &= \oint_{C'} u dy \\ \int_{A'} \frac{\partial u}{\partial y} dA &= - \oint_{C'} u dx, \end{aligned} \quad (5.13)$$

with analogous expressions for the components of the gradient of v , where the primes are added to remind the reader that these expressions are used for auxiliary cells surrounding the edges of the finite volume. A second-order approximation to the integrals on the right-hand side of these expressions divided by the cell area provides an approximation to the average gradient in the cell. This then provides an approximation, valid to second order, to the gradient along the edge contained in the auxiliary cell.

A sample auxiliary cell is depicted in Fig. 5.1 [7]. The cell in question is cell j, k defined by ABCD. The auxiliary cell A'B'C'D' provides the approximation to the velocity gradient on edge BC. In order to evaluate the integrals on the right-hand side of (5.13), the midpoint rule is applied on each edge of cell A'B'C'D'. The velocity at the midpoint of edge A'B' is taken as the average of the velocities associated with the four cells surrounding this edge. The same applies to edge C'D'. The velocity at the midpoint of edge B'C' is simply that associated with cell $j, k + 1$, while the velocity on edge D'A' is that associated with cell j, k . Once the velocity gradients are approximated, all other quantities needed to form the viscous fluxes on the edges of cell j, k , including the viscosity, are obtained by averaging the quantities associated with the cells on either side of the edge in question.

An alternative auxiliary cell can be formed with the vertices being the end points of the edge and the centroids of the cells on either side of the edge, sometimes called

Fig. 5.1 Auxiliary cell $A'B'C'D'$ for computing viscous fluxes

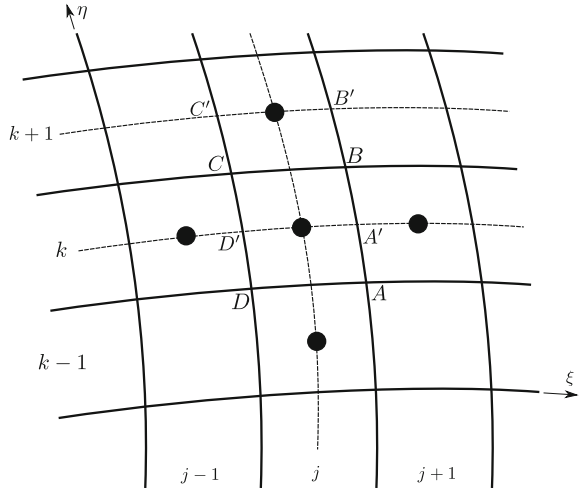
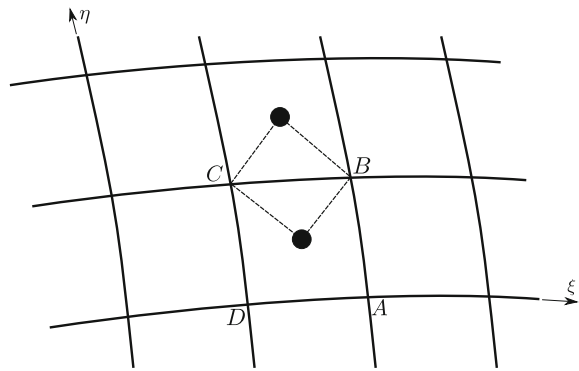


Fig. 5.2 Alternative auxiliary cell based on diamond path



a diamond path, as shown in Fig. 5.2. In this case, the trapezoidal rule is used for the integration to calculate the velocity gradients.

Once the viscous flux tensor $(\mathcal{F}_v)_l$ has been approximated at the cell edges, the net flux is determined from

$$\mathcal{L}_v Q = \sum_{l=1}^4 (\mathcal{F}_v)_l \cdot \mathbf{s}_l. \tag{5.14}$$

5.2.2 Artificial Dissipation

In analogy to the inviscid fluxes, we write the dissipation model in the following form:

$$\mathcal{L}_{\text{ad}} Q = \sum_{l=1}^4 \mathcal{D}_l \cdot \mathbf{s}_l, \quad (5.15)$$

where \mathcal{D}_l is the numerical dissipation tensor associated with each cell edge. We exploit the fact that the algorithm is applied to structured meshes. Although there is no coordinate transformation, there are effectively two coordinate directions ξ and η associated with each cell, as depicted in Fig. 5.1. Hence there are two opposing cell edges along which η varies but ξ does not, and there are two opposing cell edges along which ξ varies but η does not. For the two edges at constant ξ , the artificial dissipation tensor is given by

$$\mathcal{D} = -\epsilon^{(2)}(|A\hat{i} + B\hat{j}|)\Delta_\xi Q + \epsilon^{(4)}(|A\hat{i} + B\hat{j}|)\Delta_\xi \nabla_\xi \Delta_\xi Q, \quad (5.16)$$

where the superscripts (2) and (4) denote second- and fourth-difference dissipation, respectively, the meaning of $|A\hat{i} + B\hat{j}|$ is consistent with (2.103), A and B are the Jacobians of the inviscid flux vectors E and F , and Δ_ξ and ∇_ξ represent undivided differences in the ξ direction. For example, $\Delta_\xi Q$ is the difference between the Q values in the cells on either side of the edge. The coefficients $\epsilon^{(2)}$ and $\epsilon^{(4)}$ control the relative contribution from the two terms, analogous to the artificial dissipation scheme described in Chap. 4, and are defined below.

The reader should observe the similarity between (5.16) and (4.85). The artificial dissipation scheme described in this section is a finite-volume analog to the scheme presented in Sect. 4.4.3. Therefore it has the same basic properties. For example, the second-difference term is first order and is used near shocks, while the fourth-difference term is third order and is used in smooth regions of the flow.

Substituting the definition of \mathbf{s}_l given in (5.9), we obtain for the two edges with constant ξ

$$\mathcal{D}_l \cdot \mathbf{s}_l = -\epsilon_l^{(2)}(|A_l \Delta y_l - B_l \Delta x_l|)\Delta_\xi Q + \epsilon_l^{(4)}(|A_l \Delta y_l - B_l \Delta x_l|)\Delta_\xi \nabla_\xi \Delta_\xi Q. \quad (5.17)$$

The flux Jacobians are based on an average of the two states on either side of the edge. The Roe average (Sect. 6.3) can be used. A scalar form is obtained as follows:

$$\mathcal{D}_l \cdot \mathbf{s}_l = -\epsilon_l^{(2)}(\lambda_\xi)_l \Delta_\xi Q + \epsilon_l^{(4)}(\lambda_\xi)_l \Delta_\xi \nabla_\xi \Delta_\xi Q, \quad (5.18)$$

where

$$\lambda_\xi = |u \Delta y - v \Delta x| + a \sqrt{\Delta y^2 + \Delta x^2} \quad (5.19)$$

is the appropriate spectral radius for edges of constant ξ (see Warming et al. [8]). The spectral radius term in the η direction has the same form, but the values of Δx and Δy are associated with edges of constant η .

The treatment of the pressure sensor is consistent with (4.83), giving for the edge $j + \frac{1}{2}, k$:

$$\begin{aligned}\epsilon_l^{(2)} &= \kappa_2 \max(\Upsilon_{j+2,k}, \Upsilon_{j+1,k}, \Upsilon_{j,k}, \Upsilon_{j-1,k}) \\ \Upsilon_{j,k} &= \left| \frac{p_{j+1,k} - 2p_{j,k} + p_{j-1,k}}{p_{j+1,k} + 2p_{j,k} + p_{j-1,k}} \right| \\ \epsilon_l^{(4)} &= \max(0, \kappa_4 - \epsilon_l^{(2)}),\end{aligned}\tag{5.20}$$

where typical values of the constants are $\kappa_2 = 1/2$ and $\kappa_4 = 1/32$.

The artificial dissipation terms for the edges with constant η are analogous. They are obtained by replacing ξ with η in (5.16) and (5.17).

As described, this artificial dissipation model parallels that used with the implicit algorithm described in Chap. 4. When used with an explicit multigrid algorithm, it is sometimes modified in the following manner [7]. The spectral radius associated with the ξ direction given in (5.19) is multiplied by $\phi(r)$, which is given by

$$\phi(r_{\eta\xi}) = 1 + r_{\eta\xi}^{\zeta},\tag{5.21}$$

with

$$r_{\eta\xi} = \frac{\lambda_{\eta}}{\lambda_{\xi}},\tag{5.22}$$

where ζ is typically equal to $2/3$. The spectral radius in the η direction λ_{η} is multiplied by $\phi(r^{-1})$. This increases the amount of numerical dissipation, thus improving the high-frequency damping properties of the scheme and leading to better convergence rates with the multigrid method. This is particularly important in the case of high aspect ratio cells, for example in high Reynolds number boundary layers. In such cases, the ratio $\lambda_{\eta}/\lambda_{\xi}$ approximates the cell aspect ratio. With a cell aspect ratio of 1000, for example, ϕ is on the order of 100, and the numerical dissipation in the streamwise direction is greatly increased.

5.3 Iteration to Steady State

5.3.1 Multi-stage Time-Marching Method

The semi-discrete form (5.6) can be written as

$$\frac{d}{dt} Q_{j,k} = -\frac{1}{A_{j,k}} \mathcal{L} Q_{j,k},\tag{5.23}$$

where $\mathcal{L} = \mathcal{L}_i + \mathcal{L}_{\text{ad}} - \mathcal{L}_v$. Here we will concentrate on an explicit multi-stage time-marching method which can be used for steady flows or to solve the nonlinear problem arising at each time step in the dual-time-stepping approach to unsteady flows (see Sect. 4.5.7). In both of these settings there is no benefit to higher-order accuracy in time, and we will consider methods designed specifically for rapid convergence to steady state when used in conjunction with the multigrid method.

The effectiveness of a time-marching method for convergence to a steady state can be assessed in terms of the amplification factor (based on the σ eigenvalues in the terminology of Chap. 2) arising from the λh eigenvalues resulting from a specific spatial discretization. This is discussed further below, but we begin with a more qualitative discussion. When iterations are performed from an arbitrary initial condition to the steady-state solution, we can consider the difference between the initial condition and the steady solution to be an *error* that must be removed. Since the time-marching iterations represent a physical process, one can give a physical interpretation of the path to steady state. The error is removed through two mechanisms associated with the governing PDEs: (1) it convects out of the domain through the boundary, and (2) it dissipates within the domain through both physical and numerical dissipation. If one thinks of the error as being decomposed into modes, then low frequency error modes will typically be eliminated through convection and high frequency modes through dissipation.

A time-marching method with good convergence properties addresses these two mechanisms in the following manner. In order to enable convection of the error through the boundary, the method should be at least second-order accurate, so that the physics of convection is accurately represented, and when combined with a particular spatial discretization, the maximum stable Courant number should be as large as possible. The method should also provide damping of high frequency modes, again in combination with the spatial discretization. The latter property is particularly important in the context of the multigrid method, which will be discussed in Sect. 5.3.3. Finally, the computational cost per time step is also an important consideration.

We will begin by considering a time-marching method for the spatially discretized Euler equations, i.e. applied to the ODE system

$$\frac{d}{dt} Q_{j,k} = -\frac{1}{A_{j,k}} (\mathcal{L}_i + \mathcal{L}_{\text{ad}}) Q_{j,k} = -R(Q_{j,k}). \quad (5.24)$$

Consider a multi-stage time marching method in the following form

$$\begin{aligned} Q_{j,k}^{(0)} &= Q_{j,k}^{(n)} \\ Q_{j,k}^{(m)} &= Q_{j,k}^{(0)} - \alpha_m h R(Q_{j,k}^{(m-1)}), \quad m = 1, \dots, q \\ Q_{j,k}^{(n+1)} &= Q_{j,k}^{(q)}, \end{aligned} \quad (5.25)$$

where n is the time index, $h = \Delta t$, q is the number of stages, and the coefficients α_m , $m = 1, \dots, q$ define the method. The reader should recognize that this is not a

general form for explicit Runge-Kutta methods. For example, the classical fourth-order method given in Sect. 2.6 cannot be written in this form. Nevertheless, this form is equivalent to the more general form with respect to homogeneous ODEs and thus enables the design of schemes with tailored convergence properties.

For the purpose of discussing the analysis of such methods we will concentrate on five-stage methods, i.e. $q = 5$. Consider the homogeneous scalar ODE given by

$$\frac{du}{dt} = \lambda u, \quad (5.26)$$

where λ represents an eigenvalue of the linearized semi-discrete system. When applied to this ODE, the method given by (5.25) with $q = 5$ produces the solution

$$u_n = u_0 \sigma^n, \quad (5.27)$$

where u_0 is the initial condition, and σ is given by

$$\sigma = 1 + \beta_1 \lambda h + \beta_2 (\lambda h)^2 + \beta_3 (\lambda h)^3 + \beta_4 (\lambda h)^4 + \beta_5 (\lambda h)^5, \quad (5.28)$$

with

$$\begin{aligned} \beta_1 &= \alpha_5 \\ \beta_2 &= \alpha_5 \alpha_4 \\ \beta_3 &= \alpha_5 \alpha_4 \alpha_3 \\ \beta_4 &= \alpha_5 \alpha_4 \alpha_3 \alpha_2 \\ \beta_5 &= \alpha_5 \alpha_4 \alpha_3 \alpha_2 \alpha_1. \end{aligned} \quad (5.29)$$

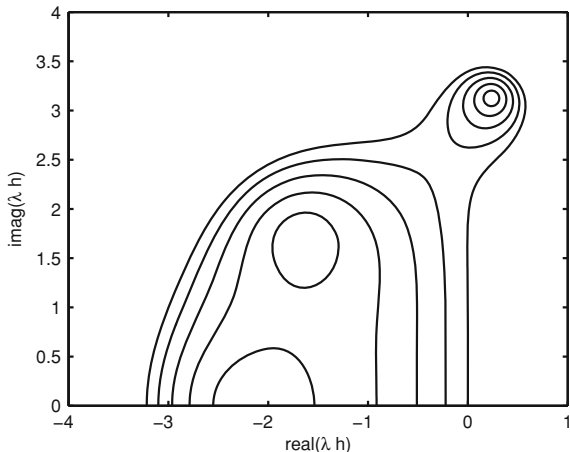
Second-order accuracy is obtained by choosing $\alpha_5 = 1$ and $\alpha_4 = 1/2$, giving $\beta_1 = 1$ and $\beta_2 = 1/2$. This leaves three free parameters that can be chosen from the perspective of optimizing convergence to steady state.

The values $\beta_3 = 1/6$, $\beta_4 = 1/24$, and $\beta_5 = 1/120$ lead to a σ that approximates $e^{\lambda h}$, which maximizes the order of accuracy of the method, at least for homogeneous ODEs such as (5.26). This is obtained with $\alpha_1 = 1/5$, $\alpha_2 = 1/4$, and $\alpha_3 = 1/3$. Figure 5.3 shows contours of $|\sigma|$ for this method plotted in the complex λh plane. The method has a large region of stability that includes a portion of the imaginary axis.

The convergence rates this method will produce depend upon the specific spatial discretization and the time step. To examine this, consider the linear convection equation

$$\frac{\partial u}{\partial t} + a \frac{\partial u}{\partial x} = 0, \quad (5.30)$$

Fig. 5.3 Contours of $|\sigma|$ for the five-stage time-marching method with $\beta_3 = 1/6$, $\beta_4 = 1/24$, and $\beta_5 = 1/120$. Contours shown have $|\sigma|$ equal to 1, 0.8, 0.6, 0.4, and 0.2



with $a > 0$ and periodic boundary conditions. Apply second-order centered differences with fourth-difference artificial dissipation to approximate the spatial derivative term:

$$-a\delta_x u = -\frac{a}{\Delta x} \left[\frac{u_{j+1} - u_{j-1}}{2} + \kappa_4(u_{j-2} - 4u_{j-1} + 6u_j - 4u_{j+1} + u_{j+2}) \right]. \quad (5.31)$$

Since the boundary conditions are periodic, Fourier analysis can be used to obtain the λ eigenvalues of the resulting semi-discrete form. They are given by

$$\lambda_m = -\frac{a}{\Delta x} \left\{ i \sin \left(\frac{2\pi m}{M} \right) + 4\kappa_4 \left[1 - \cos \left(\frac{2\pi m}{M} \right) \right]^2 \right\}, \quad m = 0 \dots M-1, \quad (5.32)$$

where M corresponds to the number of nodes in the mesh. Multiplying by the time step gives

$$\lambda_m h = -C_n \left\{ i \sin \left(\frac{2\pi m}{M} \right) + 4\kappa_4 \left[1 - \cos \left(\frac{2\pi m}{M} \right) \right]^2 \right\}, \quad m = 0 \dots M-1, \quad (5.33)$$

where $C_n = ah/\Delta x$ is the Courant number.

The λh values given by (5.33) are plotted in Fig. 5.4 for $M = 40$, $\kappa_4 = 1/32$, and $C_n = 2.5$ together with the $|\sigma|$ contours arising from the five-stage scheme (5.25) with $\alpha_1 = 1/5$, $\alpha_2 = 1/4$, and $\alpha_3 = 1/3$. Figure 5.5 plots $|\sigma(\lambda_m h)|$ vs. $\kappa \Delta x$ for $0 \leq \kappa \Delta x \leq \pi$, where $\kappa \Delta x = 2\pi m/M$. This plot shows poor damping for

Fig. 5.4 Plot of λh values given by (5.33) for $M = 40$, $\kappa_4 = 1/32$, and $C_n = 2.5$ with contours of $|\sigma|$ for the five-stage time-marching method with $\alpha_1 = 1/5$, $\alpha_2 = 1/4$, and $\alpha_3 = 1/3$. Contours shown have $|\sigma|$ equal to 1, 0.8, 0.6, 0.4, and 0.2

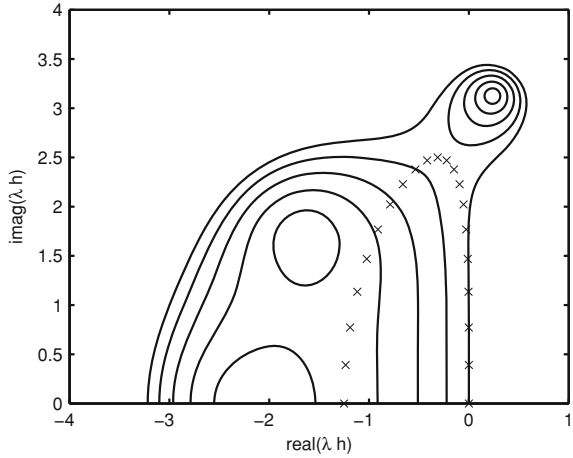
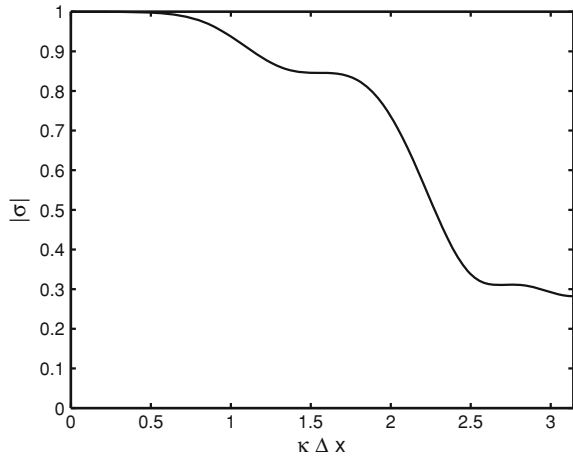


Fig. 5.5 Plot of $|\sigma|$ values vs. $\kappa \Delta x$ for the spatial operator given by (5.31) with $C_n = 2.5$, $\kappa_4 = 1/32$, and the five-stage time-marching method with $\alpha_1 = 1/5$, $\alpha_2 = 1/4$, and $\alpha_3 = 1/3$



low wavenumbers and good damping at high wavenumbers. As we shall see later, this provides a smoothing property suitable for use with the multigrid method. It is important to recognize that this model problem includes only the mechanism of damping within the domain. With periodic boundary conditions, the error cannot convect out of the domain, so this mechanism is not represented. Therefore, the Courant number is also an important quantity to be aware of. Although the effect is not seen in the present analysis, a higher stable Courant number translates into a larger time step, which enables the error to convect out through the outer boundary of the domain in fewer time steps.

Through careful selection of the free parameters, α_1 , α_2 , and α_3 , a multi-stage method can be designed for fast convergence when used in conjunction with a specific spatial discretization. For example, consider the choice $\alpha_1 = 1/4$, $\alpha_2 = 1/6$, and $\alpha_3 = 3/8$, which maximizes the stable region on the imaginary axis (see Van der

Fig. 5.6 Plot of λh values given by (5.33) for $M = 40$, $\kappa_4 = 1/32$, and $C_n = 3$ with contours of $|\sigma|$ for the five-stage time-marching method with $\alpha_1 = 1/4$, $\alpha_2 = 1/6$, and $\alpha_3 = 3/8$

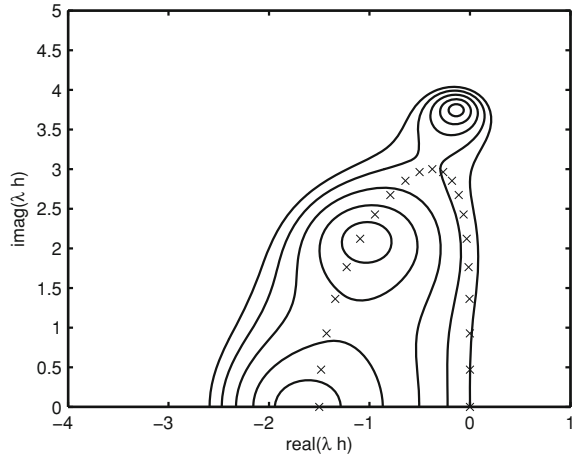
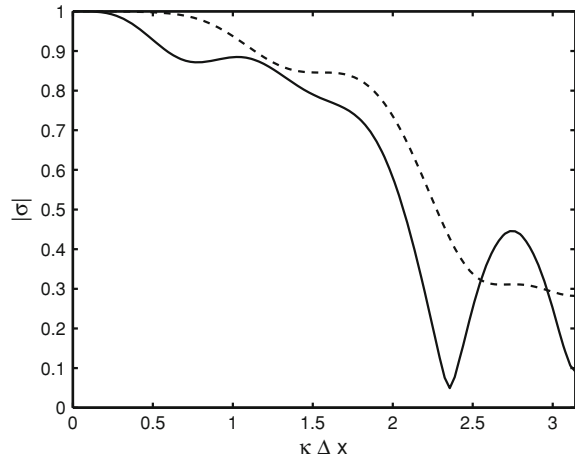


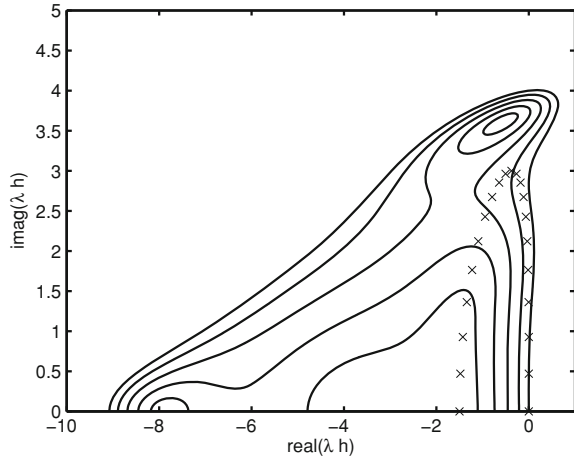
Fig. 5.7 Plot of $|\sigma|$ values vs. $\kappa \Delta x$ for the spatial operator given by (5.31) with $C_n = 3$, $\kappa_4 = 1/32$, and the five-stage time-marching method with $\alpha_1 = 1/4$, $\alpha_2 = 1/6$, and $\alpha_3 = 3/8$ (solid line). The dashed line shows the results with $C_n = 2.5$ and $\alpha_1 = 1/5$, $\alpha_2 = 1/4$, and $\alpha_3 = 1/3$



Houwen [9]). The associated plots are shown in Figs. 5.6 and 5.7 with a Courant number of 3. The improvement in damping properties is small, but the higher Courant number enables the error to be propagated to the outer boundary more rapidly. This particular choice of α coefficients is intended for use with a spatial discretization that combines centered differencing (or an equivalent finite-volume method) with artificial dissipation. One can also design multi-stage schemes specifically for upwind schemes.

One must be aware of the limitations of such scalar Fourier analysis in this context. It provides a useful guide for the design of multi-stage schemes, but, since it does not account for systems of PDEs, multidimensionality, or the effect of boundaries, the performance of such schemes when applied to the Euler equations must be assessed through more sophisticated theory or numerical experiment.

Fig. 5.8 Plot of λh values given by (5.33) for $M = 40$, $\kappa_4 = 1/32$, and $C_n = 3$ with contours of $|\sigma|$ for the five-stage time-marching method with $\alpha_1 = 1/4$, $\alpha_2 = 1/6$, and $\alpha_3 = 3/8$ with the artificial dissipation computed only on stages 1, 3, and 5



A further generalization of (5.25) can be introduced if the distinct components of $R(Q)$, for example $\mathcal{L}_i Q$ and $\mathcal{L}_{ad} Q$, are handled differently by the multi-stage method. Consider a scheme where at stage m the residual term $R(Q_{j,k}^{(m-1)})$ in (5.25) is replaced by

$$R^{(m-1)} = \frac{1}{A} \left(\mathcal{L}_i Q^{(m-1)} + \sum_{p=0}^{m-1} \gamma_{mp} \mathcal{L}_{ad} Q^{(p)} \right). \tag{5.34}$$

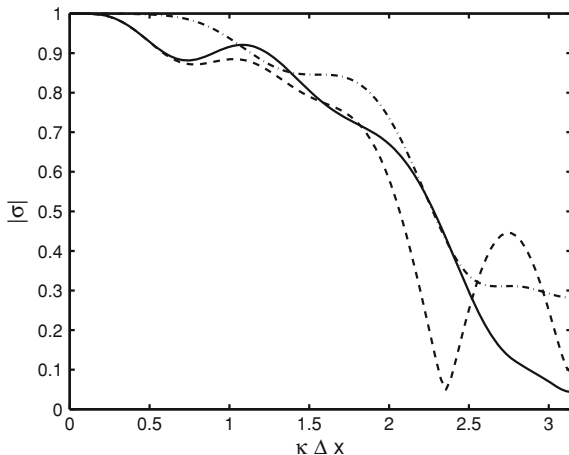
The γ_{mp} coefficients can be chosen such that the artificial dissipation operator is evaluated only at certain stages, thus reducing the computational effort per time step. The following values lead to a method in which the artificial dissipation is evaluated at the first, third, and fifth stages:

$$\begin{aligned} \gamma_{10} &= 1 \\ \gamma_{20} &= 1, \quad \gamma_{21} = 0 \\ \gamma_{30} &= 1 - \Gamma_3, \quad \gamma_{31} = 0, \quad \gamma_{32} = \Gamma_3 \\ \gamma_{40} &= 1 - \Gamma_3, \quad \gamma_{41} = 0, \quad \gamma_{42} = \Gamma_3, \quad \gamma_{43} = 0 \\ \gamma_{50} &= (1 - \Gamma_3)(1 - \Gamma_5), \quad \gamma_{51} = 0, \quad \gamma_{52} = \Gamma_3(1 - \Gamma_5), \quad \gamma_{53} = 0, \quad \gamma_{54} = \Gamma_5. \end{aligned} \tag{5.35}$$

Note that the coefficients sum to unity at each stage. With $\Gamma_3 = 0.56$ and $\Gamma_5 = 0.44$, the results shown in Figs. 5.8 and 5.9 are obtained for the linear convection equation. This method retains the favourable damping properties of the previous method while reducing the computational cost per time step, thereby reducing the overall cost to achieve a converged solution.

The above multi-stage method is also appropriate for the numerical solution of the Navier-Stokes equations. In this case, the residual includes the contribution from

Fig. 5.9 Plot of $|\sigma|$ values vs. $\kappa\Delta x$ for the spatial operator given by (5.31) with $C_n = 3$, $\kappa_4 = 1/32$, and the five-stage time-marching method with $\alpha_1 = 1/4$, $\alpha_2 = 1/6$, and $\alpha_3 = 3/8$ with the artificial dissipation computed only on stages 1, 3, and 5 (solid line). The dashed line shows the results with the artificial dissipation computed at every stage, and the dash-dot line shows the results with $C_n = 2.5$ and $\alpha_1 = 1/5$, $\alpha_2 = 1/4$, and $\alpha_3 = 1/3$



the viscous and heat conduction terms, \mathcal{L}_v . The residual can be computed at each stage as follows:

$$R^{(m-1)} = \frac{1}{A} \left(\mathcal{L}_i Q^{(m-1)} - \mathcal{L}_v Q^{(0)} + \sum_{p=0}^{m-1} \gamma_{mp} \mathcal{L}_{ad} Q^{(p)} \right). \quad (5.36)$$

The viscous terms are evaluated at the first stage only, thereby minimizing the additional cost per time step.

Local Time Stepping. Use of a local time step specific to each grid cell is important to improve the convergence rate of an explicit algorithm for steady flows. In order to understand why, consider first the use of a constant time step. For example, for the one-dimensional Euler equations we have

$$\Delta t \leq \frac{\Delta x}{|u| + a} (C_n)_{\max}, \quad (5.37)$$

where $|u| + a$ is the largest eigenvalue of the flux Jacobian, and $(C_n)_{\max}$ is the maximum Courant number for stability of the particular combination of spatial discretization and time-marching method, as determined by Fourier analysis, for example (bearing in mind that Fourier analysis provides a necessary condition for stability but not a sufficient one). The stability requirement resulting from the conditional stability associated with explicit schemes will dictate that the time step be determined based on the grid cell with the smallest value of $\Delta x/(|u| + a)$. Typically the variation in mesh spacing far exceeds the variation in the maximum wave speed; hence the time step is often limited by the size of the smallest cells in the mesh. If the smallest cells are several orders of magnitude smaller than the largest cells, then this time step will be much smaller than the optimal time step for the larger cells.

We can assign a physical meaning to the Courant number. It is the distance travelled by the fastest wave in one time step expressed in terms of the mesh spacing. For example, with a Courant number of 3, the fastest wave travels a distance $3\Delta x$ in one time step. However, if the time step is determined by a very small cell, then the effective Courant number at a large cell is very small, and it will take many time steps for a disturbance to propagate through the large cell.

On a mesh with a wide variation in mesh spacing, much faster convergence to steady state can be achieved by using a time step at each cell that gives the desired value of the Courant number for that cell. For example, in our one-dimensional example the local time step is computed from

$$(\Delta t)_j = \frac{(\Delta x)_j}{(|u| + a)_j} C_n, \quad (5.38)$$

where C_n is the desired (optimal) Courant number. The use of such a local time step destroys time accuracy but has no impact on the converged steady solution.

For the one-dimensional Euler equations, the definition of the local time step (5.38) is a relatively straightforward matter. Extension to multidimensions and to the Navier-Stokes equations is not straightforward, and a number of approximations are typically made. In order to present some of the issues, we will consider the convection-diffusion equation as a model problem:

$$\frac{\partial u}{\partial t} + a \frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2}. \quad (5.39)$$

With periodic boundary conditions and second-order centered-difference approximations to both the first and the second spatial derivatives on a mesh with M nodes, the eigenvalues of the semi-discrete operator matrix are, from Fourier analysis:

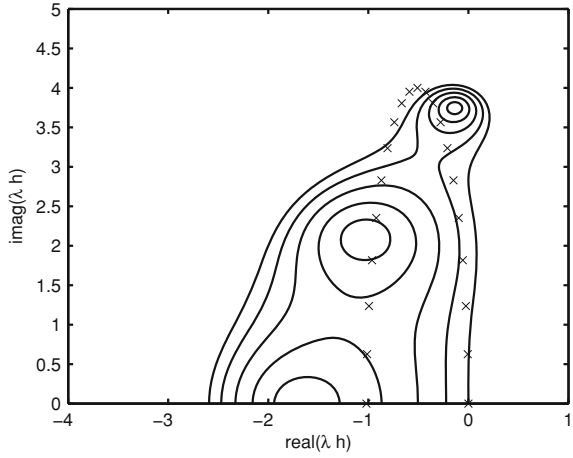
$$\lambda_m = -\frac{a}{\Delta x} i \sin\left(\frac{2\pi m}{M}\right) - \frac{4\nu}{\Delta x^2} \sin^2\left(\frac{\pi m}{M}\right), \quad m = 0, \dots, M-1, \quad (5.40)$$

where $\Delta x = 2\pi/M$. The imaginary part of the eigenvalue is associated with the convective term, the real part with the diffusive term.

Let us consider the solution of this semi-discrete system using the five-stage time-marching method described previously with $\alpha_1 = 1/4$, $\alpha_2 = 1/6$, and $\alpha_3 = 3/8$. From Fig. 5.6 we see that this method is stable for imaginary eigenvalues up to 4 and for negative real eigenvalues up to -2.59 . We will attempt to define a local time step based solely on this information about the time-marching method. Multiplying the above eigenvalues by h gives

$$\lambda_m h = -C_n i \sin\left(\frac{2\pi m}{M}\right) - 4V_n \sin^2\left(\frac{\pi m}{M}\right), \quad m = 0, \dots, M-1, \quad (5.41)$$

Fig. 5.10 Plot of λh values given by (5.33) for $M = 40$, $\kappa_4 = 1/32$, and $C_n = 3$ with contours of $|\sigma|$ for the five-stage time-marching method with $\alpha_1 = 1/4$, $\alpha_2 = 1/6$, and $\alpha_3 = 3/8$. Time step based on minimum of h_c and h_d



where $V_n = \nu h / \Delta x^2$ is sometimes referred to as the von Neumann number. Based on the above properties of the time-marching method, we require for stability:

$$\begin{aligned} C_n &= \frac{ah}{\Delta x} \leq 4 \\ V_n &= \frac{\nu h}{\Delta x^2} \leq \frac{2.59}{4}. \end{aligned} \quad (5.42)$$

Based on the first criterion, one can define a convective time step limit as

$$h_c \leq \frac{4\Delta x}{a}, \quad (5.43)$$

while the second criterion gives the diffusive time step limit as

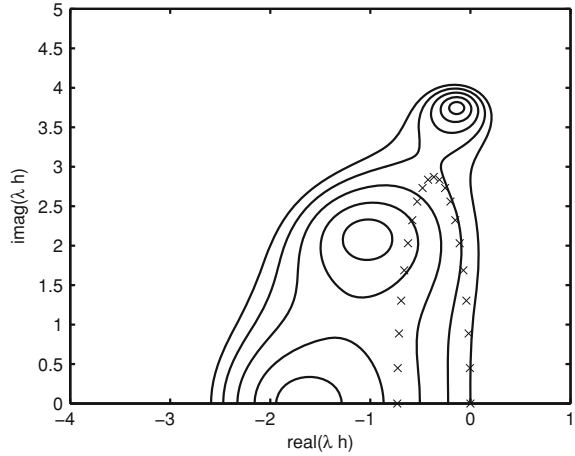
$$h_d \leq \frac{2.59\Delta x^2}{4\nu}. \quad (5.44)$$

It is tempting, therefore, to choose the time step as the minimum of h_c and h_d , which ensures that the imaginary part of all eigenvalues is less than 4 and the negative real part is less than 2.5. However, consider an example with $a = 1$, $\nu = 0.01$ and $M = 40$. The resulting spectrum is displayed in Fig. 5.10 along with the $|\sigma|$ contours of the time-marching method. Some eigenvalues lie outside the stable region; hence this time step definition is not adequate to ensure stability.

A more conservative time step definition is obtained from

$$\frac{1}{h} = \frac{1}{h_c} + \frac{1}{h_d}. \quad (5.45)$$

Fig. 5.11 Plot of λh values given by (5.33) for $M = 40$, $\kappa_4 = 1/32$, and $C_n = 3$ with contours of $|\sigma|$ for the five-stage time-marching method with $\alpha_1 = 1/4$, $\alpha_2 = 1/6$, and $\alpha_3 = 3/8$. Time step based on (5.45)



With this choice, the time step is less than the minimum of h_c and h_d . For the above example, the λh values plotted in Fig. 5.11 are obtained. All of the eigenvalues lie well within the stable region of the time-marching method.

Based on approximations such as this, various local time stepping strategies have been developed for explicit multi-stage time-marching methods with the goal of providing robust and rapid convergence. One such approach, which is based on (5.45), is given by Swanson and Turkel [7] as follows:

$$h = \frac{N_i A}{\lambda_C + \lambda_D}, \tag{5.46}$$

where

$$\begin{aligned} \lambda_C &= \lambda_\xi + \lambda_\eta \\ \lambda_D &= (\lambda_D)_\xi + (\lambda_D)_\eta + (\lambda_D)_{\xi\eta}, \end{aligned} \tag{5.47}$$

with

$$\begin{aligned} (\lambda_D)_\xi &= \frac{\gamma\mu}{Re\rho Pr} A^{-1} (x_\eta^2 + y_\eta^2) \\ (\lambda_D)_\eta &= \frac{\gamma\mu}{Re\rho Pr} A^{-1} (x_\xi^2 + y_\xi^2) \\ (\lambda_D)_{\xi\eta} &= \frac{\mu}{Re\rho} A^{-1} \left[-\frac{7}{3} (y_\eta y_\xi + x_\xi x_\eta) + \frac{1}{3} \sqrt{(x_\eta^2 + y_\eta^2)(x_\xi^2 + y_\xi^2)} \right]. \end{aligned} \tag{5.48}$$

The quantity N_i is the stability bound on pure imaginary eigenvalues associated with the time-marching method used. The assumption made is that the maximum negative real eigenvalue is of a similar magnitude. The cell area is denoted by A ,

and the terms λ_ξ and λ_η are defined as in (5.19). For the cell in question, λ_ξ is obtained by averaging the values obtained from the two edges of constant ξ , while λ_η is obtained by averaging the values obtained from the two edges of constant η . The diffusive terms $(\lambda_D)_\xi$, $(\lambda_D)_\eta$, and $(\lambda_D)_{\xi\eta}$ are approximations to the spectral radii of the respective viscous flux Jacobians. The metric terms appearing in these terms are also calculated based on undivided differences for the appropriate edges and then averaging to get a value for the cell. For example, y_η is obtained by averaging Δy for opposing edges of constant ξ , and the other terms are obtained similarly.

Given the various approximations made in determining the local time step for the Navier-Stokes equations in multidimensions, it is typical to include a factor in the time step definition that is determined to be effective, i.e. both reliable and efficient, through numerical experimentation. The use of a local time step enables fast convergence of an explicit method on a mesh with a large variation in mesh spacing. However, it does not address the slow convergence of explicit methods resulting from grid cells with high aspect ratios.

5.3.2 Implicit Residual Smoothing

Implicit residual smoothing is a convergence acceleration technique that enables a substantial increase in the Courant number, thus speeding up the propagation of disturbances to the outer boundary. First we define a residual that incorporates the local time step:

$$\tilde{R}_{j,k}^{(m-1)} = \frac{(\Delta t)_{j,k}}{A_{j,k}} \left(\mathcal{L}_i Q_{j,k}^{(m-1)} - \mathcal{L}_v Q_{j,k}^{(0)} + \sum_{p=0}^{m-1} \gamma_{mp} \mathcal{L}_{ad} Q_{j,k}^{(p)} \right). \quad (5.49)$$

A smoothed residual $\bar{R}_{j,k}^{(m-1)}$ is found from the following:

$$(1 - \beta_\xi \nabla_\xi \Delta_\xi)(1 - \beta_\eta \nabla_\eta \Delta_\eta) \bar{R}_{j,k}^{(m-1)} = \tilde{R}_{j,k}^{(m-1)} \quad (5.50)$$

and replaces the term $hR(Q_{j,k}^{(m-1)})$ in (5.25). As in Sect. 5.2.2, Δ_ξ and ∇_ξ represent undivided differences in the ξ direction, and Δ_η and ∇_η are the corresponding operators in the η direction. The smoothing coefficients β_ξ and β_η are discussed below. The operator in the ξ direction can be rewritten as

$$(1 - \beta_\xi \nabla_\xi \Delta_\xi) \bar{R}_{j,k}^{(m-1)} = \left[-\beta_\xi \bar{R}_{j-1,k}^{(m-1)} + (1 + 2\beta_\xi) \bar{R}_{j,k}^{(m-1)} - \beta_\xi \bar{R}_{j+1,k}^{(m-1)} \right]. \quad (5.51)$$

The residuals of the individual equations, i.e. mass, x and y -momentum, and energy, are smoothed separately. Hence in two dimensions implicit residual smoothing

requires the solution of two scalar tridiagonal systems per stage of the multi-stage time-stepping scheme. This adds considerably to the computational cost per time step.

In order to understand and analyze implicit residual smoothing, we return to the linear convection equation with periodic boundary conditions discretized using the operator given in (5.31). In a one-dimensional scalar problem, the implicit residual smoothing operator is given by

$$B_p(M : -\beta, 1 + 2\beta, -\beta)\bar{R} = R, \tag{5.52}$$

or

$$\bar{R} = [B_p(M : -\beta, 1 + 2\beta, -\beta)]^{-1}R, \tag{5.53}$$

where we use the notation for the banded periodic matrix $B_p(M : a, b, c)$ given in (2.33). Hence we can obtain the eigenvalues of the system with implicit residual smoothing by dividing those given in (5.33) by the eigenvalues of $B_p(M : -\beta, 1 + 2\beta, -\beta)$,¹ leading to

$$\lambda_m h = -C_n \frac{i \sin\left(\frac{2\pi m}{M}\right) + 4\kappa_4 \left[1 - \cos\left(\frac{2\pi m}{M}\right)\right]^2}{1 + 4\beta \sin^2\left(\frac{\pi m}{M}\right)}, \quad m = 0 \dots M - 1. \tag{5.54}$$

For the problem studied previously, with $M = 40$, $C_n = 3$, and $\kappa = 1/32$ coupled with a smoothing coefficient of $\beta = 0.6$, the eigenvalues λh are displayed in Fig. 5.12. There are two primary observations to be made. First, the magnitude of the eigenvalues has generally been reduced as a result of the implicit residual smoothing. This means that a larger Courant number can be used while remaining within the stability bounds of a given time-marching method. Second, the eigenvalues associated with small m , which are those at the origin and just above and below, are affected the least by the residual smoothing. These eigenvalues correspond to well resolved modes, i.e. low frequency modes, which are those that convect out through the boundary. Hence the residual smoothing has little effect on the manner in which these modes are propagated.

Figure 5.13 shows these eigenvalues superimposed on the $|\sigma|$ contours of the five-stage method with dissipation evaluated on the first, third, and fifth stages. As a result of the implicit residual smoothing, a Courant number of 7 can be used while remaining in the stable region. Consequently, disturbances will propagate to the outer boundary in fewer time steps than without residual smoothing (where the Courant number is 3). Figure 5.14 shows that the damping properties are similar to those obtained without residual smoothing, so the primary benefit is the higher Courant number. It is important to recognize that the use of implicit residual smoothing entails

¹ as a result of the properties of circulant matrices

Fig. 5.12 Plot of λh values given by (5.33) for $M = 40$, $\kappa_4 = 1/32$, and $C_n = 3$ without implicit residual smoothing (x) and with implicit residual smoothing (+) with $\beta = 0.6$

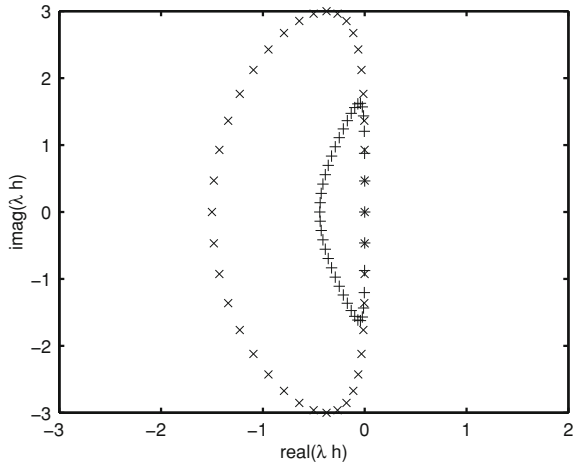
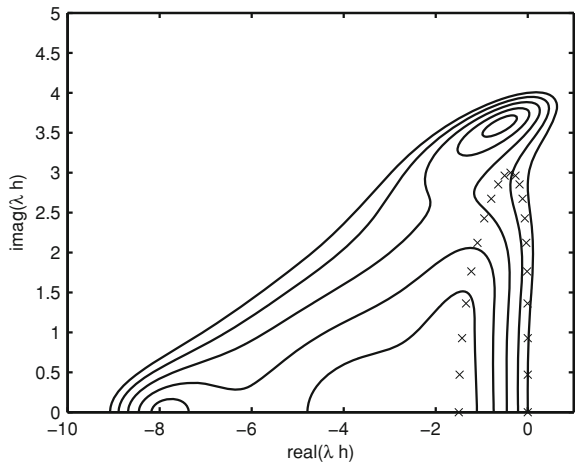


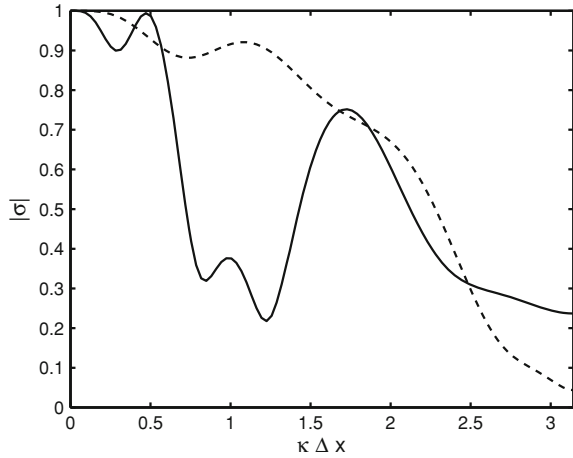
Fig. 5.13 Plot of λh values given by (5.33) for $M = 40$, $\kappa_4 = 1/32$, and $C_n = 7$ with implicit residual smoothing with $\beta = 0.6$ and contours of $|\sigma|$ for the five-stage time-marching method with $\alpha_1 = 1/4$, $\alpha_2 = 1/6$, and $\alpha_3 = 3/8$ with the artificial dissipation computed only on stages 1, 3, and 5



a significant computational expense per time step that must be weighed against the reduced number of time steps to steady state associated with the increased Courant number.

The main purpose of implicit residual smoothing is to enable the use of a larger time step, or Courant number. Typically the Courant number limit is increased by a factor of two to three. This enables disturbances to propagate more rapidly to the domain boundary without compromising damping properties, as shown in Fig. 5.14. The maximum stable Courant number continues to increase as β is increased. However, at some point this does not lead to faster convergence, and there is an optimum value of β . The reason is that the implicit residual smoothing eliminates time accuracy and therefore interferes with the physics of convection and hence the propagation of error to the outer boundary. In Fig. 5.12 we saw that the smaller eigenvalues are not

Fig. 5.14 Plot of $|\sigma|$ values vs. $\kappa\Delta x$ for the spatial operator given by 5.31 with $C_n = 7$ with implicit residual smoothing ($\beta = 0.6$), $\kappa_4 = 1/32$, and the five-stage time-marching method with $\alpha_1 = 1/4$, $\alpha_2 = 1/6$, and $\alpha_3 = 3/8$ with the artificial dissipation computed only on stages 1, 3, and 5 (solid line). The dashed line shows the results without implicit residual smoothing with $C_n = 3$



greatly affected by the smoothing with $\beta = 0.6$. As β is increased, these eigenvalues begin to deviate more from their values without implicit residual smoothing. Hence there is a compromise between a large Courant number and accurate representation of the convection process for low frequency modes.

Based on one- and two-dimensional stability analysis as well as numerical experiments, Swanson and Turkel [7] developed the following formulas for β_ξ and β_η :

$$\beta_\xi = \max \left\{ \frac{1}{4} \left[\left(\frac{N}{N^*} \frac{1}{1 + \psi r_{\eta\xi}} \right)^2 - 1 \right], 0 \right\}$$

$$\beta_\eta = \max \left\{ \frac{1}{4} \left[\left(\frac{N}{N^*} \frac{1}{1 + \psi r_{\eta\xi}^{-1}} \right)^2 - 1 \right], 0 \right\}. \tag{5.55}$$

Here N^* is the Courant number for the unsmoothed scheme, while N is the Courant number for the smoothed scheme, so N/N^* typically takes a value between 2 and 3. The ratio of inviscid spectral radii was defined in (5.22), and ψ is a user-defined parameter generally between 0.125 and 0.25.

5.3.3 The Multigrid Method

The multigrid method systematically uses sets of coarser grids to accelerate the convergence of iterative schemes. It can be applied to any iterative method that displays a smoothing property, i.e. it preferentially damps high-frequency error modes. For explicit iterative methods, multigrid is critical to obtaining fast convergence to steady-state for stiff problems.

Multigrid theory is well developed for elliptic problems, such as the steady diffusion equation. For such problems, there is a correlation between the eigenvalues and the spatial frequencies of the associated eigenvectors. For example, for the diffusion equation, the eigenvalues of the semi-discrete operator matrix resulting from a second-order centered-difference discretization are all real and negative (see Sect. 2.3.4). The eigenvectors associated with the eigenvalues with small magnitudes have low spatial frequencies, while those corresponding to eigenvalues with large magnitudes have high frequencies. This means that in the exact solution of the semi-discrete ODE system (see Sect. 2.3.3) the high frequency components in the transient solution are rapidly damped, while the low frequency components are slowly damped. This is a fundamental property of a diffusive system that is retained after discretizing in space.

Given this correlation between eigenvalues with large magnitudes and high space frequencies, it is a natural property of several iterative methods (such as the Gauss-Seidel relaxation method) to reduce error components corresponding to high spatial frequencies more effectively than those corresponding to low spatial frequencies. Moreover, iterative methods can be specifically designed to have this property, such as the Richardson method described in Lomax et al. [10]. The multigrid method exploits this property by systematically using coarser grids to target the removal of specific components of the error. For example, high frequency error components are rapidly damped on the initial grid, whose density is determined by accuracy considerations. Hence the error is *smoothed* on that mesh. The low frequency error components can be represented on a coarser mesh on which some of them appear as high frequencies, where the frequency is relative to the mesh spacing, and are thus more rapidly damped.

To make this clearer, consider the range of wavenumbers that are representable on a mesh with spacing Δx_f , which are given by $0 \leq \kappa \Delta x_f \leq \pi$. If the mesh spacing is increased by a factor of two ($\Delta x_c = 2\Delta x_f$), then the wavenumber range $\pi/2 \leq \kappa \Delta x_f \leq \pi$ on the original mesh cannot be represented on the coarse mesh. However, the range of error modes with $0 \leq \kappa \Delta x_f \leq \pi/2$ have their value of $\kappa \Delta x$ doubled. Those error modes in the wavenumber range $\pi/4 \leq \kappa \Delta x_f \leq \pi/2$ on the fine mesh, which are poorly damped compared to those in the high wavenumber range, appear in the wavenumber range $\pi/2 \leq \kappa \Delta x_c \leq \pi$, which are well damped on the coarse mesh. This can be repeated with successively coarser meshes until the mesh is so coarse that the problem can be affordably solved directly rather than iteratively, such that on that mesh all error modes are damped. This is essentially how the multigrid method works for a linear diffusion problem. See Chap. 10 of [10] for a more detailed description.

Here we are interested in the application of the multigrid method to the discretized Euler and Navier-Stokes equations, which introduces two important differences in comparison to the diffusion equation. First, the Euler and Navier-Stokes equations are nonlinear, which means that the *full approximation storage* approach in which both the residual and the solution must be transferred from the fine to the coarse mesh must be used. Second, in the diffusion problem with Dirichlet boundary conditions, the only mechanism available to remove error modes is diffusion within the domain.

When the Euler and Navier-Stokes equations are solved, error is also propagated through the outer boundary of the domain. This mechanism is primarily associated with low frequency error modes, for which the spatial discretization is relatively accurate. Since such modes are typically poorly damped, this is an important mechanism for their removal. For example, referring to Figs. 5.9 and 5.14, we see that our discretization of the linear convection equation, which includes artificial dissipation, shows preferential damping of high frequencies, i.e. a smoothing property, with the particular time-marching method used.

The analysis reflected in Figs. 5.9 and 5.14 does not include the mechanism of error removal by convection through the boundary. In Sects. 5.3.1 and 5.3.2, we accounted for this by designing schemes to permit as large a Courant number as possible. The multigrid method also exploits this mechanism of error removal. The low frequency error modes for which propagation through the boundary is important are well represented on the coarser mesh. Since the mesh spacing is doubled on the coarse mesh, maintaining a constant Courant number will lead to a doubling of the time step, enabling disturbances to propagate to the outer boundary in roughly half as many time steps. Therefore, when applied to the Euler and Navier-Stokes equations, the multigrid method enhances the convergence rate both through accelerating the damping of error modes within the domain and through accelerating the removal of error through the outer boundary.

We now present the implementation of the multigrid method in conjunction with the cell-centered finite-volume scheme and multi-stage time-marching method described in this chapter. The system of ODEs resulting from the spatial discretization is

$$\frac{d}{dt} Q_{j,k} = -\frac{1}{A_{j,k}} \mathcal{L} Q_{j,k} = -R_{j,k}. \quad (5.56)$$

A sequence of grids can be created by successively removing every second grid line in each coordinate direction from the finest grid. The coarse grid cell is then the agglomeration of four fine grid cells sharing a common grid node. If the number of fine grid cells in each coordinate direction is even, then all cells can be merged. Typically sequences of three to five meshes are used. For a five-mesh sequence, the finest mesh should have a number of cells in each direction that is a multiple of 16 in order that the second coarsest mesh have an even number of cells in each direction.

We will now describe a two-grid process that can readily be extended to an arbitrary number of grids, since the process is recursive. We first complete one or more iterations of the five-stage time-marching method with implicit residual smoothing described previously to obtain Q_h . This is followed by an additional computation of the full residual based on the updated solution, including the convective, viscous, and artificial dissipation contributions.

The next step is to transfer the residual and the solution from the fine to the coarse mesh, a process known as *restriction*. Consider the residual first. The term $\mathcal{L} Q_{j,k}$ is the net flux out of cell j, k . In order to transfer the residual to the coarse mesh in a conservative manner, the net flux out of the coarse grid cell should be equal to the

net flux out of the four fine grid cells that were merged to form the coarse grid cell. This is achieved simply by summing the flux of each of the four fine grid cells, since internal fluxes will cancel, giving

$$I_h^{2h} R_h = \frac{1}{A_{2h}} \sum_{p=1}^4 A_h R_h, \quad (5.57)$$

where the subscripts h and $2h$ denote the fine and coarse grids, respectively, and I_h^{2h} is the restriction operator.

An analogous conservative approach is taken to restrict the solution Q . The amount of a conserved quantity, such as mass, momentum, or energy, in the coarse grid cell should be equal to the sum of the amount of that conserved quantity in the constituent fine grid cells. Since Q represents the conserved quantities per unit volume in a given cell, it must be multiplied by the cell area to give the total amount of the conserved quantity in the cell (noting that in two dimensions the conserved quantities are per unit depth). Hence the formula for restricting the solution to the coarse mesh is

$$Q_{2h}^{(0)} = I_h^{2h} Q_h = \frac{1}{A_{2h}} \sum_{p=1}^4 A_h Q_h, \quad (5.58)$$

where $Q_{2h}^{(0)}$ is the solution used to initiate the multi-stage method on the coarse mesh (see (5.25)).

Now we are ready to solve a problem on the coarse mesh. It is important to recognize that it is not our goal to find the solution to the governing equations on the coarse mesh. The purpose of solving on the coarse mesh is to provide a correction to the solution that will reduce the residual on the fine mesh. To this end, a forcing term P_{2h} is introduced into the ODE solved on the coarse mesh as follows [7]:

$$\frac{d}{dt} Q_{2h} = -[R_{2h}(Q_{2h}) + P_{2h}], \quad (5.59)$$

where R_{2h} is the residual computed by applying the spatial discretization on the coarse mesh. The forcing term is

$$P_{2h} = I_h^{2h} R_h - R_{2h}(Q_{2h}^{(0)}), \quad (5.60)$$

which is the difference between the restricted residual and the coarse grid residual computed based on the restricted solution. If we were to drive the coarse mesh problem (5.59) to convergence, we would drive to zero

$$R_{2h}(Q_{2h}) + P_{2h} = R_{2h}(Q_{2h}) - R_{2h}(Q_{2h}^{(0)}) + I_h^{2h} R_h. \quad (5.61)$$

Thus we would obtain the change in the solution on the coarse mesh ($Q_{2h} - Q_{2h}^{(0)}$) that produces a change in the coarse mesh residual ($R_{2h}(Q_{2h}) - R_{2h}(Q_{2h}^{(0)})$) that offsets the residual restricted from the fine mesh ($I_h^{2h} R_h$), which is the purpose of the coarse grid correction.

Let us examine the forcing term in more detail. At the first stage of the multi-stage method on the coarse mesh the residual is

$$-[R_{2h}(Q_{2h}^{(0)}) + P_{2h}] = -[R_{2h}(Q_{2h}^{(0)}) + I_h^{2h} R_h - R_{2h}(Q_{2h}^{(0)})] = -I_h^{2h} R_h, \quad (5.62)$$

which is simply the residual restricted from the fine mesh. This means that once the solution on the fine mesh has converged, the coarse mesh calculation will produce no correction, which is appropriate. This provides a useful test when debugging a multigrid algorithm. One can compute the converged solution on the fine mesh using the basic algorithm without multigrid and use this as the initial condition for the multigrid algorithm. Quite a few possible errors can reveal themselves if the coarse mesh correction is nonzero. For example, it is important to enforce the boundary conditions on the coarse mesh before computing the term $R_{2h}(Q_{2h}^{(0)})$ in the forcing function P_{2h} . Otherwise, when they are enforced during the first stage of the multi-stage method, the value of $R_{2h}(Q_{2h}^{(0)})$ will not cancel with the same term in P_{2h} , and a nonzero correction will be produced.

When the multi-stage method is applied to (5.59), the m th stage becomes

$$Q_{2h}^{(m)} = Q_{2h}^{(0)} - \alpha_m h [R(Q_{2h}^{(m-1)}) + P_{2h}], \quad (5.63)$$

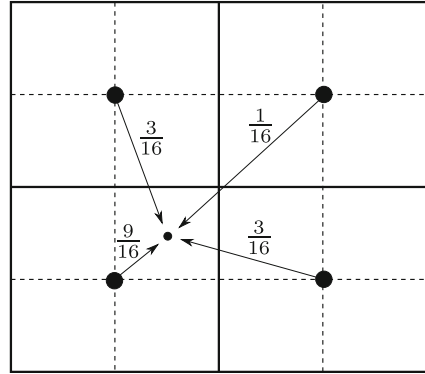
where $R(Q_{2h}^{(m-1)})$ is computed as in (5.36). Note that P_{2h} does not depend on m and remains fixed during the stages. If the present coarse mesh is not the coarsest mesh in the sequence, then one or more iterations of the multi-stage method are performed and the problem is transferred to the next coarser mesh after an additional computation of the residual. The residual and solution are restricted using the operators in (5.57) and (5.58), respectively. When continuing to a coarser mesh, the residual that is restricted must include the forcing term, i.e. $R_{2h}(Q_{2h}) + P_{2h}$.

Once the coarsest grid level is reached, the correction to the solution must be transferred, or *prolonged*, back to the next finer grid. There is an important condition that the transfer operators must satisfy in order to achieve mesh-size independent rates of convergence of the multigrid algorithm, which can be written as [11]:

$$p_R + p_P + 2 > p_{PDE}, \quad (5.64)$$

where p_R and p_P are the highest degree polynomials interpolated exactly by the restriction and prolongation operators, respectively, and p_{PDE} is the order of the PDE. For the restriction operator given in (5.57), $p_R = 0$. Therefore, a prolongation based on a piecewise constant interpolation ($p_P = 0$) is adequate for the Euler equations, but

Fig. 5.15 Bilinear prolongation operator for cell-centered scheme in two dimensions



a piecewise linear interpolation ($p_P = 1$) is needed for the Navier-Stokes equations, for which $p_{\text{PDE}} = 2$.

The prolongation operation for a cell-centered algorithm in two dimensions is depicted in Fig. 5.15. With bilinear interpolation, the value of the correction ΔQ in each fine mesh cell is calculated based on ΔQ in four coarse mesh cells. The resulting prolongation operator is

$$I_{2h}^h \Delta Q = \frac{1}{16} (9\Delta Q_1 + 3\Delta Q_2 + 3\Delta Q_3 + \Delta Q_4), \quad (5.65)$$

where ΔQ_1 is the value in the coarse mesh cell containing the fine mesh cell, ΔQ_2 and ΔQ_3 are the values in the coarse mesh cells that share an edge with the coarse mesh cell containing the fine mesh cell, and ΔQ_4 is the value in the coarse mesh cell that shares only a vertex with the fine mesh cell.

The ΔQ to be prolonged to the fine mesh is the difference between Q_{2h} after completing the iteration or iterations on the coarse mesh and the original $Q_{2h}^{(0)}$ that was restricted to the coarse mesh based on (5.58). Hence we obtain for the corrected Q_h on the fine mesh:

$$Q_h^{(\text{corrected})} = Q_h + I_{2h}^h (Q_{2h} - Q_{2h}^{(0)}), \quad (5.66)$$

where Q_h is the value originally computed on the fine mesh (see (5.58)), and I_{2h}^h is the prolongation operator given in (5.65).

This basic two-grid framework provides the basis for many variations, known as multigrid cycles, that depend on the number of grids in the sequence and the manner in which they are visited. Figure 5.16 displays two popular cycles, the *V cycle* and the *W cycle*, based on four grids. Downward pointing arrows indicate restriction to a coarser mesh, while upward pointing arrows indicate prolongation to a finer mesh. There are trade-offs between the two cycles, and typically experimentation is needed to determine which is more efficient and robust for a given problem class. In the *W cycle*, relatively more computations are performed on the coarser grid levels; since

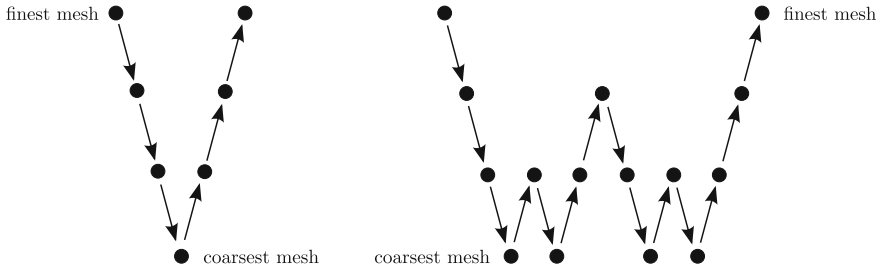
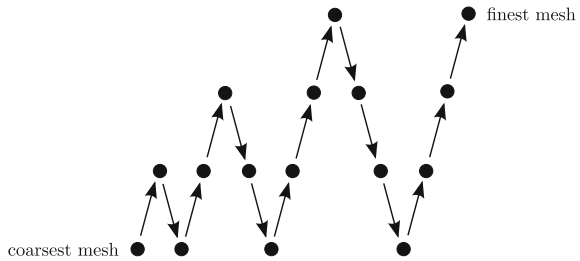


Fig. 5.16 Four-grid V and W multigrid cycles

Fig. 5.17 Full multigrid with four grids



these are inexpensive, the W cycle is often more efficient than the V cycle. Unlike the classical approach to multigrid for linear problems, where the problem is solved exactly on the coarsest mesh, in the present context one simply applies one or more iterations of the multi-stage method on the coarsest mesh. Experiments show that there is typically no benefit to converging further on the coarsest mesh. Similarly, it is rare to see an overall benefit in terms of computational expense in going beyond four or five grids. Within a given cycle, there are also several possible variants. For example, one can apply the multi-stage scheme at each grid level when transferring from the coarse grid levels back to the fine levels, or one can simply add the correction and prolong the result to the next finer grid. Some authors apply an implicit smoothing to the corrections. It is also common to apply various simplifications, such as a lower-order spatial discretization, on the coarser grids. This reduces the computational expense without affecting the converged solution.

Finally, the *full multigrid method* combines the concept of mesh sequencing presented in Sect. 4.5.6 with the multigrid method. Since a sequence of meshes exists as well as a transfer operator from coarse to fine meshes, this is a natural approach. The computation begins on the coarsest mesh in the sequence, on which a number of multi-stage iterations are performed. The solution is transferred to the next finer mesh, and a number of two-grid multigrid cycles are carried out. This solution is transferred to the next finer grid, and a number of three-grid cycles are performed. This process continues until the full cycle is reached, as depicted in Fig. 5.17.

5.4 One-Dimensional Examples

As in Chap. 4, we present examples of the application of the algorithm described in this chapter to the quasi-one-dimensional Euler equations. These examples coincide with the exercises listed at the end of the chapter, giving the reader a benchmark for their results. In the context of a one-dimensional uniform mesh, the implementation of the second-order finite-volume method described in this chapter is very similar to that of the second-order finite-difference method of the previous chapter. Consequently, we will use the same spatial discretization as in Sect. 4.8, but coupled with the explicit multi-stage multigrid algorithm presented in this chapter. Our focus here is on steady flows.

The spatial discretization used to illustrate the performance of the multi-stage multigrid algorithm is node centered. Therefore, the grid transfer operators described in this chapter cannot be used, and we introduce suitable operators for a node-centered scheme in one dimension. The coarse grid is formed by removing every other grid node from the fine mesh. An odd number of nodes should be used to ensure that the boundary nodes are preserved in the coarse mesh. For a sequence of p grids, the finest mesh should have a number of interior nodes equal to some multiple of 2^{p-1} minus one.

The simplest restriction operator is simple injection, where the coarse grid node is assigned the value at the corresponding fine grid node. In a linear weighted restriction, the coarse grid node is assigned a value equal to one-half that at the corresponding fine grid node plus one-quarter of that at each of the neighbours of the fine grid node, which do not exist on the coarse grid. The reader should experiment with these two approaches in order to examine their effect on multigrid convergence. After restricting the solution to the coarse mesh, the boundary values should be reset to satisfy the boundary conditions on the coarse mesh.

For prolongation, linear interpolation gives the following transfer operator. Each fine grid node for which there is a corresponding coarse grid node is assigned the value at that coarse grid node. For fine grid nodes that do not exist on the coarse grid, they receive one-half of the value from each neighbouring coarse grid node. After prolonging the correction to a finer mesh, the boundary values should be reset to satisfy the boundary conditions on the fine mesh.

For the methods presented in this chapter and the previous one, the converged steady solution is independent of the details of the iterative method such as the time step. Since we apply the same spatial discretization as for the results presented in Sect. 4.8, except for a different value of κ_4 , the solutions will be nearly identical to those presented previously, as long as the residual is reduced sufficiently. Therefore, we concentrate here only on convergence histories.

The example results presented here are based on the five-stage time-marching method with $\alpha_1 = 1/4$, $\alpha_2 = 1/6$, and $\alpha_3 = 3/8$ and the artificial dissipation computed only on stages 1, 3, and 5. Without residual smoothing, $C_n = 3$. Residual smoothing is applied with $\beta = 0.6$ and $C_n = 7$. The multigrid method is based on the multi-stage method with implicit residual smoothing and the same parameter values.

Fig. 5.18 Residual convergence histories for the subsonic channel flow problem with 103 interior nodes using the explicit algorithm with $C_n = 3$ (-), $C_n = 7$ and implicit residual smoothing with $\beta = 0.6$ (- -), and a four-level W multigrid cycle with $C_n = 7$ and implicit residual smoothing with $\beta = 0.6$ (-.)

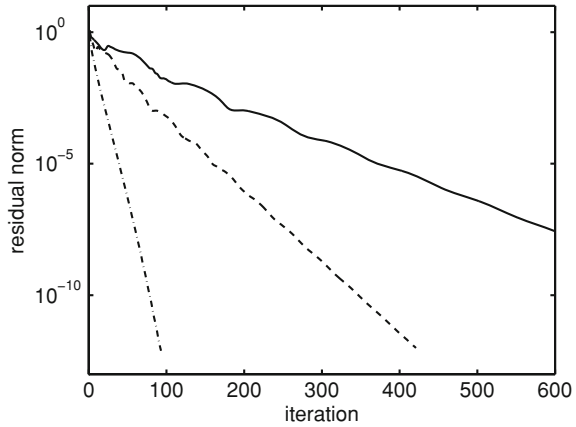
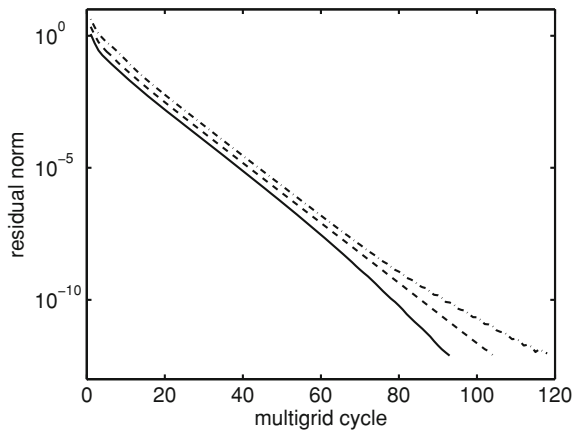


Fig. 5.19 Residual convergence histories for the subsonic channel flow problem with 103 interior nodes (-), 207 interior nodes (- -), and 415 interior nodes (-.) using the explicit algorithm with a W multigrid cycle with $C_n = 7$ and implicit residual smoothing with $\beta = 0.6$. Four grid levels are used on the coarsest mesh, five on the intermediate mesh, and six on the finest mesh



The solution is restricted through simple injection, while linear weighted restriction is used for the residual. For both W and V cycles, the time-marching method is not applied after prolongation except on the finest mesh when the cycle is repeated. The artificial dissipation coefficients are $\kappa_4 = 1/32$ and $\kappa_2 = 0.5$ in all cases.

Figure 5.18 compares the convergence of the explicit algorithm on a single grid without implicit residual smoothing, with implicit residual smoothing, and with a four-level W multigrid cycle for the subsonic channel on a mesh with 103 interior nodes. The norm of the residual of the conservation of mass equation is shown. With the multigrid algorithm, the residual is reduced to below 10^{-12} in 93 multigrid cycles. Figure 5.19 displays the performance of the multigrid algorithm W cycle for varying numbers of grid nodes. Four grid levels are used with 103 interior nodes, five with 207 interior nodes, and six with 413 interior nodes. Thus the coarsest mesh, which has 12 interior nodes, is the same in each case. With this approach, the number of multigrid cycles needed for convergence is nearly independent of the mesh size, as shown in the figure. Figure 5.20 shows that the V cycle does not converge as quickly for this case and requires more cycles as the mesh is refined.

Fig. 5.20 Residual convergence histories for the subsonic channel flow problem with 103 interior nodes (-), 207 interior nodes (- -), and 415 interior nodes (-.) using the explicit algorithm with a V multigrid cycle with $C_n = 7$ and implicit residual smoothing with $\beta = 0.6$. Four grid levels are used on the coarsest mesh, five on the intermediate mesh, and six on the finest mesh

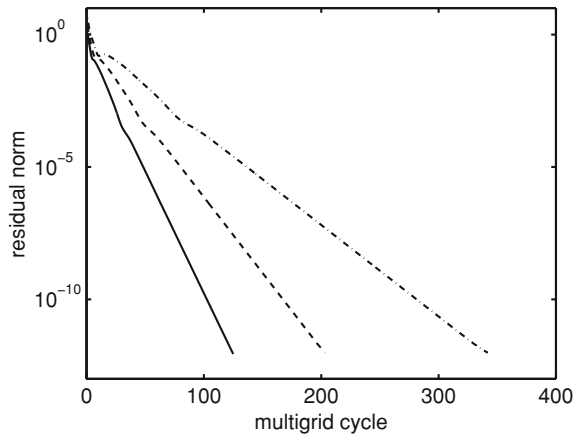
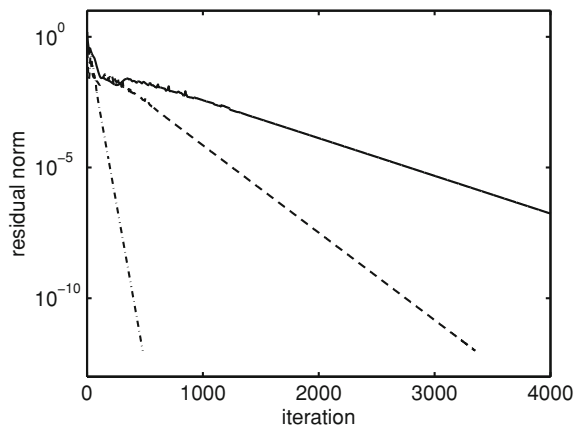


Fig. 5.21 Residual convergence histories for the transonic channel flow problem with 103 interior nodes using the explicit algorithm with $C_n = 3$ (-), $C_n = 7$ and implicit residual smoothing with $\beta = 0.6$ (- -), and a four-level W multigrid cycle with $C_n = 7$ and implicit residual smoothing with $\beta = 0.6$ (-.)



Figures 5.21, 5.22, and 5.23 show the same comparisons for the transonic channel problem. Although the number of iterations or multigrid cycles required for convergence is much higher in this case, the trends are very similar. Implicit residual smoothing improves the convergence rate by a factor close to two. Multigrid is very effective in reducing the number of iterations needed, and the W cycle converges in fewer cycles than the V cycle.

5.5 Summary

The algorithm described in this chapter has the following key features:

- The discretization of the spatial derivatives is accomplished through a second-order cell-centered finite-volume method applied on a structured grid. This approach

Fig. 5.22 Residual convergence histories for the transonic channel flow problem with 103 interior nodes (-), 207 interior nodes (- -), and 415 interior nodes (-.) using the explicit algorithm with a W multigrid cycle with $C_n = 7$ and implicit residual smoothing with $\beta = 0.6$. Four grid levels are used on the coarsest mesh, five on the intermediate mesh, and six on the finest mesh

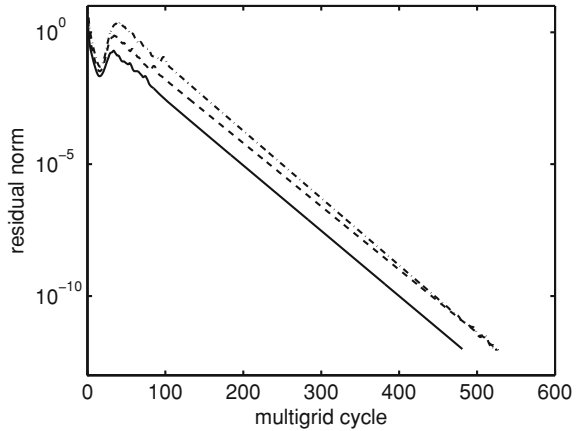
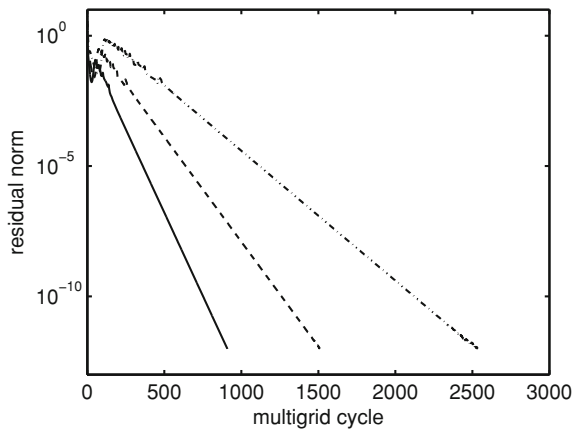


Fig. 5.23 Residual convergence histories for the transonic channel flow problem with 103 interior nodes (-), 207 interior nodes (- -), and 415 interior nodes (-.) using the explicit algorithm with a V multigrid cycle with $C_n = 7$ and implicit residual smoothing with $\beta = 0.6$. Four grid levels are used on the coarsest mesh, five on the intermediate mesh, and six on the finest mesh



can be extended to unstructured grids. Numerical dissipation is added through a nonlinear artificial dissipation scheme that combines a third-order dissipative term in smooth regions of the flow with a first-order term near shock waves. A pressure-based term is used as a shock sensor.

- After discretization in space, the original PDEs are converted to a large system of ODEs. For computations of steady flows, a five-stage explicit method is used in which the artificial dissipation is computed only on stages one, three, and five, and the viscous flux operator is applied only on the first stage. At each stage, the residual is smoothed by application of a scalar tridiagonal implicit operator in each coordinate direction. The multigrid method is applied in order to accelerate convergence to steady state. For computations of unsteady flows, this algorithm can be used within the context of an implicit dual-time-stepping approach.

5.6 Exercises

For related discussion, see Sect. 5.4.

5.1 Write a computer program to apply the explicit multigrid algorithm presented in this chapter to the quasi-one-dimensional Euler equations for the following subsonic problem. $S(x)$ is given by

$$S(x) = \begin{cases} 1 + 1.5 \left(1 - \frac{x}{5}\right)^2 & 0 \leq x \leq 5 \\ 1 + 0.5 \left(1 - \frac{x}{5}\right)^2 & 5 \leq x \leq 10 \end{cases} \quad (5.67)$$

where $S(x)$ and x are in meters. The fluid is air, which is considered to be a perfect gas with $R = 287 \text{ N} \cdot \text{m} \cdot \text{kg}^{-1} \cdot \text{K}^{-1}$, and $\gamma = 1.4$, the total temperature is $T_0 = 300 \text{ K}$, and the total pressure at the inlet is $p_{01} = 100 \text{ kPa}$. The flow is subsonic throughout the channel, with $S^* = 0.8$. Use the spatial discretization described in Chap. 4 with the nonlinear scalar artificial dissipation model, since, on a uniform mesh in one dimension, it is essentially the same as that presented in this chapter. Compare your solution with the exact solution computed in Exercise 3.1. Show the convergence history for each case. Experiment with parameters, such as the multigrid cycle (e.g. W and V), the number of grid levels, the Courant number, and the implicit residual smoothing coefficient, to examine their effect on convergence. Find optimal values of the implicit residual smoothing coefficient and the Courant number for rapid and reliable convergence.

5.2 Repeat Exercise 5.1 for a transonic flow in the same channel. The flow is subsonic at the inlet, there is a shock at $x = 7$, and $S^* = 1$. Compare your solution with that calculated in Exercise 3.2.

References

1. Jameson, A., Schmidt, W., Turkel, E.: Numerical solutions of the Euler equations by finite volume methods using Runge-Kutta time-stepping schemes. AIAA Paper 81-1259, 1981
2. Baker, T.J., Jameson, A., Schmidt, W.: A family of fast and robust Euler codes. Princeton University Report MAE 1652, 1984
3. Jameson, A., Baker, T.J.: Multigrid solution of the Euler equations for aircraft configurations. AIAA Paper 84-0093, 1984
4. Jameson, A.: Solution of the Euler equations for two-dimensional transonic flow by a multigrid method. *Appl. Math. Comput.* **13**(3–4), 327–355 (1983)
5. Jameson, A.: Multigrid algorithms for compressible flow calculations. In: Proceedings of the 2nd European Conference on Multigrid Methods, Lecture Notes in Mathematics 1228, Springer-Verlag, 1986
6. Swanson, R.C., Turkel, E.: On central-difference and upwind schemes. *J. Comput. Phys.* **101**(2), 292–306 (1992)
7. Swanson, R.C., Turkel, E.: Multistage schemes with multigrid for Euler and Navier-Stokes equations. NASA TP 3631, 1997
8. Warming, R.F., Beam, R.M., Hyett, B.J.: Diagonalization and simultaneous symmetrization of the gas-dynamic matrices. *Math. Comput.* **29**(132), 1037–1045 (1975)

9. Van der Houwen, P.J.: Construction of Integration Formulas for Initial Value Problems. North Holland, Amsterdam (1977)
10. Lomax, H., Pulliam, T.H., Zingg, D.W.: Fundamentals of Computational Fluid Dynamics. Springer, Berlin (2001)
11. Hemker, P.W.: On the order of prolongations and restrictions in multigrid procedures. *J. Comput. Appl. Math.* **32**(3), 423–429 (1990)