Chapter 2
Finite-Difference Discretization of the Advection-Diffusion Equation

2.1 Introduction

Finite-difference methods are numerical methods that find solutions to differential equations using approximate spatial and temporal derivatives that are based on discrete values at spatial grid points and discrete time levels. As the grid spacing and time step are made small, the error due to finite differencing becomes small with correct implementation. In this chapter, we present the fundamentals of the finite-difference discretization using the advection-diffusion equation, which is a simple model for the Navier–Stokes equations.

Before we start the presentation of the finite-difference formulation, let us note that in addition to reducing the error in finite differencing, there are two issues in finite-difference methods which may be more important than the actual value of error itself. First is the properties of the error. For schemes with error that behaves diffusively, the solution becomes smoother than the actual physical solution and for schemes with error that acts dispersively, non-physical oscillation develops in the solution leading to its possible blowup. The second important concept is whether the discrete derivative relations are compatible. For example, differentiation rules for continuous functions f and g

$$\frac{\partial(fg)}{\partial x} = f\frac{\partial g}{\partial x} + \frac{\partial f}{\partial x}g \tag{2.1}$$

$$\frac{\partial^2 f}{\partial x^2} = \frac{\partial}{\partial x}\left(\frac{\partial f}{\partial x}\right) \tag{2.2}$$

$$\frac{\partial^2 f}{\partial x \partial y} = \frac{\partial}{\partial x}\left(\frac{\partial f}{\partial y}\right) = \frac{\partial}{\partial y}\left(\frac{\partial f}{\partial x}\right) \tag{2.3}$$

must hold also in a discrete manner. In this book, we refer to those finite-difference schemes that satisfy the derivative properties discretely as *compatible*. We discuss these topics in further detail in this chapter.

T. Kajishima and K. Taira, *Computational Fluid Dynamics*,
DOI 10.1007/978-3-319-45304-0_2

Finite-difference schemes that have minimal errors and satisfy the derivative relations in the discretized settings should be selected. On the other hand, schemes that do not discretely satisfy the derivative relations should be avoided, because they may produce non-physical solutions despite being supposedly highly accurate.

2.2 Advection-Diffusion Equation

For incompressible flow with constant viscosity, the momentum equation in the Navier–Stokes equations, Eq. (1.39), can be written in Cartesian coordinates as

$$\underbrace{\frac{\partial u_i}{\partial t}}_{\substack{\text{unsteady}\\ \text{term}}} + \underbrace{u_j \frac{\partial u_i}{\partial x_j}}_{\substack{\text{advective}\\ \text{term}}} = \underbrace{-\frac{1}{\rho}\frac{\partial p}{\partial x_i}}_{\substack{\text{pressure}\\ \text{gradient}\\ \text{term}}} + \underbrace{\nu \frac{\partial^2 u_i}{\partial x_j \partial x_j}}_{\substack{\text{diffusive}\\ \text{term}}} + \underbrace{f_i}_{\substack{\text{external}\\ \text{forcing}\\ \text{term}}}. \tag{2.4}$$

The advective term here is nonlinear and requires special care in numerical calculation. The diffusive term smoothes out the velocity profile and provides a stabilizing effect in numerical computations. The pressure gradient term represents forcing on a fluid element induced by the spatial variation in pressure. The pressure variable is solved by coupling the continuity equation, Eq. (1.40), with the above momentum equation. External forcing includes gravitational, electromagnetic, or fictitious forces (due to non-inertial reference frames). Because of the coupled effects from these terms, the examination of the physical and numerical properties of the governing equation, Eq. (2.4) becomes quite complex. Hence we in general consider a simplified model without the pressure gradient and external forcing terms in one dimension,

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2}, \tag{2.5}$$

which is known as the *Burgers' equation*. This equation is often used to validate numerical methods because the exact solution is available. We can further consider the linear approximation of the above equation with constant advective speed c (≥ 0) and diffusivity a (≥ 0) to obtain the linear advection-diffusion equation[1]

$$\frac{\partial f}{\partial t} + c\frac{\partial f}{\partial x} = a \frac{\partial^2 f}{\partial x^2}. \tag{2.6}$$

When each of the terms is removed from the advection-diffusion equation, the resulting equations have different characteristic properties. When we set $a = 0$, we obtain the advection equation

[1]The terms *advection* and *convection* are often used interchangeably. Technically speaking, convection refers to the combined transport by advection and diffusion. Thus, we use advection and convection with careful distinction in this book.

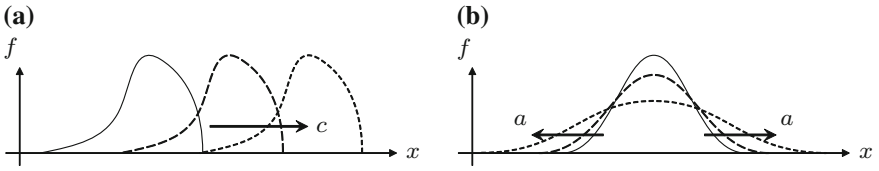

Fig. 2.1 Behavior of the solutions to the advection and diffusion equations. **a** Advection. **b** Diffusion

$$\frac{\partial f}{\partial t} + c\frac{\partial f}{\partial x} = 0, \tag{2.7}$$

which is hyperbolic. When we set $c = 0$, we have the diffusion equation

$$\frac{\partial f}{\partial t} = a\frac{\partial^2 f}{\partial x^2}, \tag{2.8}$$

which is parabolic. Finally, if $\partial f/\partial t = 0$, we find the steady advection-diffusion equation

$$c\frac{\partial f}{\partial x} = a\frac{\partial^2 f}{\partial x^2}, \tag{2.9}$$

which is elliptic. In other words, the advective-diffusion equation as well as the Navier–Stokes equations consist of the combination of these three types of partial differential equations. Properties from one of the models may become more dominant than the others depending on the flow condition.

Solutions for one-dimensional constant coefficient advection and diffusion equations are presented in Fig. 2.1. Since $f(x,t) = f(x - ct, 0)$ satisfies the advection equation, Eq. (2.7), we observe that the solution keeps the initial profile while translating in the x-direction with speed c. For the diffusion equation, Eq. (2.8), the solution decreases ($\partial f/\partial t < 0$) where the solution profile is convex ($\partial^2 f/\partial x^2 < 0$) and increases ($\partial f/\partial t > 0$) where the profile is concave ($\partial^2 f/\partial x^2 > 0$), smoothing out the solution profile over time.

2.3 Finite-Difference Approximation

Let us consider the finite-difference approximation of the spatial derivatives in the advection-diffusion equation, Eq. (2.6). While there are various techniques to derive the finite-difference equations, derivations are commonly based on Taylor series expansions or polynomial approximations.

We note with caution that readers should not use the difference equations that will be presented in Sects. 2.3.1 and 2.3.2 naively. It is important to study the derivations based on Taylor series expansions and the error analyses of the resulting formulas. Similarly, the polynomial approximation to analytically obtain the derivative approximation can provide insights into fundamentals of the numerical methods.

However, the equations presented in Sects. 2.3.1 and 2.3.2 will only be used during the discussion of numerical viscosity and filtering operation for LES in this book. Direct use of these equations in a naive fashion for flow calculations is not recommended.

2.3.1 Taylor Series Expansion

Here, we derive the finite-difference approximations of derivatives using Taylor series expansions and analyze the associated errors. Let us consider the one-dimensional case. Denoting the spatial derivatives of a continuous function $f(x)$ as

$$f'(x) = \frac{\mathrm{d}f(x)}{\mathrm{d}x}, \quad f''(x) = \frac{\mathrm{d}^2 f(x)}{\mathrm{d}x^2}, \quad \cdots, \quad f^{(m)}(x) = \frac{\mathrm{d}^m f(x)}{\mathrm{d}x^m}, \tag{2.10}$$

the Taylor series expansion about point x evaluated at point $x + y$ is

$$f(x+y) = f(x) + \sum_{m=1}^{\infty} \frac{y^m}{m!} f^{(m)}(x). \tag{2.11}$$

Consider setting $x = x_j$ and $y = x_k - x_j$, where x_j represents the jth grid point in space. Denoting the functional values at discrete points as $f_k = f(x_k)$, the Taylor series expansion about x_j can be written as

$$f_k = f_j + \sum_{m=1}^{\infty} \frac{(x_k - x_j)^m}{m!} f_j^{(m)}. \tag{2.12}$$

Using functional values of $f_k = f(x_k)$ at a number of x_k, we will derive approximations for the m-derivative $f_j^{(m)} = f^{(m)}(x_j)$ at point x_k. The geometric arrangement of grid points (x_k) required to construct such finite-difference approximation at a point is called a *stencil*. In this section, we present the derivation of the finite-difference formulas and the discuss the associated order of accuracy.

Big $\mathcal{O}$ Notation

Before we derive the finite-difference formulas, let us briefly introduce a mathematical notation that will be used throughout the book. Below, we will be using what is known as the big $\mathcal{O}$ (Landau notation) [5] to describe the asymptotic behavior of terms and functions. In most of the discussions, this notation will be used to describe the error terms in numerical computation and to study order of magnitude estimates later in examining turbulence.

For example, the big $\mathcal{O}$ is used in the following manner:

$$\begin{aligned} \sin(x) &= x - \tfrac{1}{3!}x^3 + \tfrac{1}{5!}x^5 - \cdots \quad \text{for all } x \\ &= x - \tfrac{1}{3!}x^3 + \mathcal{O}(x^5) \quad \text{as } x \to 0 \end{aligned} \tag{2.13}$$

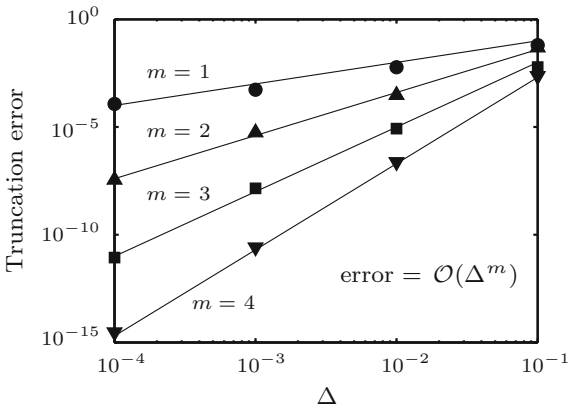

Fig. 2.2 Typical convergence of truncation errors. The slopes of the convergence curves on the log–log plot correspond to the orders of accuracy m

The first line is the Taylor series expansion of $\sin(x)$ about $x = 0$. The second line tells us that one can approximate $\sin(x)$ by $x - \frac{1}{3!}x^3$ in the neighborhood of $x = 0$ and the associated error will decrease faster than a constant times $|x^5|$ as $x \to 0$. With the two-term approximation, we can say $\sin(x) \approx x - \frac{1}{3!}x^3$ with an fifth-order *truncation error* term $\mathcal{O}(x^5)$. Note that it is also possible to use more or less number of terms in the approximation to increase or decrease the accuracy of the computation.

Let us also provide a visual description for the convergence behavior of the truncation error. Consider a truncation error that can be expressed as $\mathcal{O}(\Delta^m)$ for a given numerical approximation with Δ being a small quantity $\ll 1$. If we plot the error over Δ with the use of logarithmic scales for both the x and y-axes, the slope of the error curve (convergence curve) becomes m, as illustrated in Fig. 2.2. The exponent of the leading error term or the slope of the convergence curve m is called the *order of accuracy* of the given numerical method. This term will be used throughout the textbook to assess the accuracies of spatial and temporal numerical discretization schemes.

Central-Difference for Uniform Grid

Taylor series expansion of a function about a point $x_j = j\Delta$ for a uniform mesh, as shown in Fig. 2.3, becomes

$$f_{j+k} = f_j + \sum_{m=1}^{\infty} \frac{(k\Delta)^m}{m!} f_j^{(m)}, \tag{2.14}$$

where $k = \pm 1, \pm 2, \cdots, \pm N$. Using the series expansion of the given function, we can derive numerical approximations for the derivatives.

As an example, let us consider three-point finite-difference schemes that are based on f_{j-1}, f_j, f_{j+1}. Taking the difference of following two Taylor series:

$$f_{j+1} = f_j + \Delta f_j' + \frac{\Delta^2}{2} f_j'' + \frac{\Delta^3}{6} f_j^{(3)} + \frac{\Delta^4}{24} f_j^{(4)} + \cdots \tag{2.15}$$

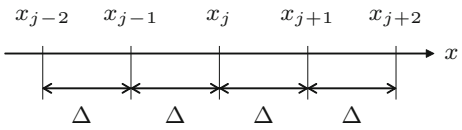

Fig. 2.3 Placement of grid points with uniform spacing (one-dimensional)

$$f_{j-1} = f_j - \Delta f'_j + \frac{\Delta^2}{2} f''_j - \frac{\Delta^3}{6} f^{(3)}_j + \frac{\Delta^4}{24} f^{(4)}_j - \cdots \tag{2.16}$$

and eliminating the even-derivative terms, we find

$$f_{j+1} - f_{j-1} = 2\Delta f'_j + \frac{\Delta^3}{3} f^{(3)}_j + \mathcal{O}(\Delta^5). \tag{2.17}$$

If we had taken the sum of the two Taylor series, we would have found that

$$f_{j+1} + f_{j-1} = 2f_j + \Delta^2 f''_j + \frac{\Delta^4}{12} f^{(4)}_j + \mathcal{O}(\Delta^6). \tag{2.18}$$

From Eqs. (2.17) and (2.18), we can find the three-point finite-difference formula for f'_j and f''_j

$$f'_j = \frac{-f_{j-1} + f_{j+1}}{2\Delta} - \frac{\Delta^2}{6} f^{(3)}_j + \mathcal{O}(\Delta^4) \tag{2.19}$$

$$f''_j = \frac{f_{j-1} - 2f_j + f_{j+1}}{\Delta^2} - \frac{\Delta^2}{12} f^{(4)}_j + \mathcal{O}(\Delta^4). \tag{2.20}$$

With a three-point stencil, we can obtain up to the second-order derivative with finite differencing. The second and third terms on the right-hand side of Eqs. (2.19) and (2.20) represent the truncation errors. As we consider small Δ with bounded high-order derivatives, the truncation error becomes proportional to Δ^2. That means that when the mesh size is reduced by 1/2, the error should become 1/4 in size. When the truncation error is proportional to Δ^n, the finite-difference scheme is said to have nth order of accuracy. The stencil for Eqs. (2.19) and (2.20) is arranged about x_j with symmetry in the positive and negative directions, as well as with symmetry in the corresponding coefficients (their magnitudes). Such schemes are called *central finite-difference schemes*. In particular, Eqs. (2.19) and (2.20) are both second-order accurate central-difference schemes.

If we use a five-point stencil, we can find the finite-difference formulas up to the fourth derivatives

$$f'_j = \frac{f_{j-2} - 8f_{j-1} + 8f_{j+1} - f_{j+2}}{12\Delta} + \frac{\Delta^4}{30} f^{(5)}_j + \mathcal{O}(\Delta^6) \tag{2.21}$$

$$f''_j = \frac{-f_{j-2} + 16f_{j-1} - 30f_j + 16f_{j+1} - f_{j+2}}{12\Delta^2} + \frac{\Delta^4}{90} f^{(6)}_j + \mathcal{O}(\Delta^6) \tag{2.22}$$

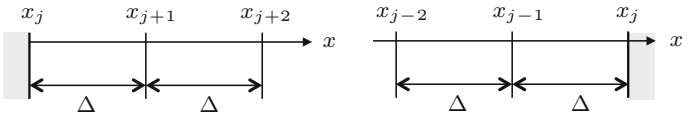

Fig. 2.4 One-sided difference stencils for the *left* and *right edges* of the computational domain

$$f_j^{(3)} = \frac{-f_{j-2} + 2f_{j-1} - 2_{j+1} + f_{j+2}}{2\Delta^3} - \frac{\Delta^2}{4} f_j^{(5)} + \mathcal{O}(\Delta^4) \tag{2.23}$$

$$f_j^{(4)} = \frac{f_{j-2} - 4f_{j-1} + 6f_j - 4_{j+1} + f_{j+2}}{\Delta^4} - \frac{\Delta^2}{6} f_j^{(6)} + \mathcal{O}(\Delta^4). \tag{2.24}$$

The five-point central-difference formulation results in fourth-order accuracy for the first and second derivatives and second-order accuracy for the third and fourth derivatives. Although we do not encounter high-order derivatives in the Navier–Stokes equations, the even derivates such as $f^{(4)}$ with a five-point stencil or $f^{(6)}$ with a seven-point stencil sometimes are used for introducing artificial viscosity or approximating filtering functions, which will be discussed later.

One-Sided Difference for Uniform Grid

When we require the finite-difference approximation for derivatives at the ends of a computational domain, we have to work with one-sided stencils,[2] as shown in Fig. 2.4 when x_j is located at the computational boundary. Given a one-sided n-point stencil, we can find the finite-difference formulas with $(n-1)$-order accuracy for the first derivative and $(n-2)$-order accuracy for the second derivative.

Let us present the derivation for the first derivative. We can approximate f'_j with the following two-point formulas:

$$f'_j = \frac{-f_{j-1} + f_j}{\Delta} + \frac{\Delta}{2} f''_j + \mathcal{O}(\Delta^2) \tag{2.25}$$

$$f'_j = \frac{-f_j + f_{j+1}}{\Delta} - \frac{\Delta}{2} f''_j + \mathcal{O}(\Delta^2) \tag{2.26}$$

based on Eqs. (2.16) and (2.15), respectively. Since the truncation errors are proportional to Δ, these schemes are first-order accurate. If we incorporate another spatial point into the stencil and eliminate the f''_j term, we end up with the second-order accurate three-point stencil finite-difference schemes of

$$f'_j = \frac{f_{j-2} - 4f_{j-1} + 3f_j}{2\Delta} + \frac{\Delta^2}{3} f_j^{(3)} + \mathcal{O}(\Delta^3) \tag{2.27}$$

$$f'_j = \frac{-3f_j + 4f_{j+1} - f_{j+2}}{2\Delta} - \frac{\Delta^2}{3} f_j^{(3)} + \mathcal{O}(\Delta^3). \tag{2.28}$$

[2] Similar idea is used for evaluating temporal derivatives if data is available only from past time.

Next, let us consider cases where the finite-difference error can be problematic. Take a function $f = x^n$ ($n = 2, 3, 4, \cdots$) whose derivative is $f' = nx^{n-1} = 0$ at $x = 0$. Setting $\Delta = 1$ for ease of analysis, we get $f_j = j^n$. For $n = 2$, Eq. (2.26) returns $f'_0 = 1$ which is incorrect, but Eq. (2.28) gives $f'_0 = 0$ which is the correct solution. For $n = 3$, Eq. (2.28) provides $f'_0 = -2$ while in reality $f' \geq 0$ (=0 only at $x = 0$). The numerical solution turns negative, which is opposite in sign for the gradient.

Furthermore, as we will observe in the next section, if grid-stretching is applied such that the adjacent grid is stretched three-times or larger in size near the wall boundary, one-sided three-point finite-difference schemes cannot correctly compute the coordinate transform coefficient [1]. For turbulent flow over a flat plate, turbulent energy $k \propto y^4$ and the Reynolds stress $\overline{u'v'} \propto y^3$. Therefore, it is important to use enough points in the stencil to track changes in the gradient of functions in the vicinity of the wall.

For the second derivatives, the one-sided three-point schemes are

$$f''_j = \frac{f_{j-2} - 2f_{j-1} + f_j}{\Delta^2} + \Delta f^{(3)}_j + \mathcal{O}(\Delta^2) \tag{2.29}$$

$$f''_j = \frac{f_j - 2f_{j+1} + f_{j+2}}{\Delta^2} - \Delta f^{(3)}_j + \mathcal{O}(\Delta^2). \tag{2.30}$$

Observe that these formulas are the same as Eq. (2.20) but just shifted by one grid. That means that $(f_{j-1} - 2f_j + f_{j+1})/\Delta^2$ is an approximation to the second derivative at any of the three points, x_{j-1}, x_j, and x_{j+1}. However, the properties of the truncation error is different for the different points at which the derivative is evaluated. We note in passing that to achieve second-order or higher accuracy for the second derivative with a one-sided stencil, we need at least four points for the finite-difference approximation.

Note that the errors for one-sided finite-difference schemes have the same order of accuracy as the central-difference scheme for the first derivative, while the order of accuracy is reduced by one for the second derivative.

Finite-Difference for Nonuniform Grid

There are two approaches to develop finite-difference schemes for nonuniform grids. The first approach is to employ Taylor series expansion, Eq. (2.12), in physical space, as shown in Fig. 2.5a. The second approach is to introduce a mapping

$$\frac{\partial f}{\partial x} = \frac{\partial \xi}{\partial x}\frac{\partial f}{\partial \xi} \tag{2.31}$$

such that the grid on the transformed variable ξ is spaced uniformly, as illustrated in Fig. 2.5b. In such case, we need to determine the coordinate transform $\partial\xi/\partial x$ and then construct the usual finite-difference scheme on the uniform grid for $\partial f/\partial\xi$. It is convenient to set the grid spacing in ξ to be 1, so that the coefficient $(\partial\xi/\partial x)$ represents the inverse of physical grid spacing.

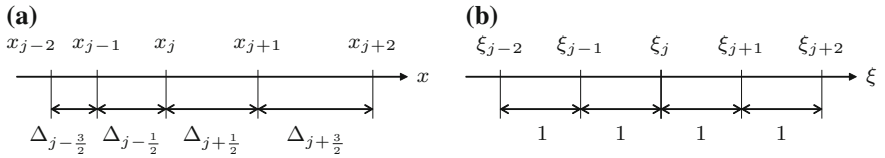

Fig. 2.5 Spatial discretization of a nonuniform grid (one-dimensional). **a** Nonuniform grid spacing in physical space. **b** Uniform grid spacing in transformed computational space

Alternatively, we can derive finite-difference schemes by directly differencing a function over a nonuniform grid in physical space. For a three-point difference method, the Taylor series expansions are

$$f_{j-1} = f_j - \Delta_{j-\frac{1}{2}} f'_j + \frac{\Delta^2_{j-\frac{1}{2}}}{2} f''_j - \frac{\Delta^3_{j-\frac{1}{2}}}{6} f^{(3)}_j + \cdots \tag{2.32}$$

$$f_{j+1} = f_j + \Delta_{j+\frac{1}{2}} f'_j + \frac{\Delta^2_{j+\frac{1}{2}}}{2} f''_j + \frac{\Delta^3_{j+\frac{1}{2}}}{6} f^{(3)}_j + \cdots \tag{2.33}$$

These expansions can be used to eliminate f''_j to derive the finite-difference expression for the first derivative with second-order accuracy

$$f'_j = -\frac{\Delta_{j+\frac{1}{2}} f_{j-1}}{\Delta_{j-\frac{1}{2}}(\Delta_{j-\frac{1}{2}} + \Delta_{j+\frac{1}{2}})} - \frac{(\Delta_{j-\frac{1}{2}} - \Delta_{j+\frac{1}{2}}) f_j}{\Delta_{j-\frac{1}{2}} \Delta_{j+\frac{1}{2}}} + \frac{\Delta_{j-\frac{1}{2}} f_{j+1}}{\Delta_{j+\frac{1}{2}}(\Delta_{j-\frac{1}{2}} + \Delta_{j+\frac{1}{2}})} - \frac{\Delta_{j-\frac{1}{2}} \Delta_{j+\frac{1}{2}}}{6} f^{(3)}_j + \mathcal{O}(\Delta^3). \tag{2.34}$$

If we instead eliminate f'_j, the expression for the second derivative can be found

$$f''_j = -\frac{2 f_{j-1}}{\Delta_{j-\frac{1}{2}}(\Delta_{j-\frac{1}{2}} + \Delta_{j+\frac{1}{2}})} - \frac{f_j}{\Delta_{j-\frac{1}{2}} \Delta_{j+\frac{1}{2}}} + \frac{2 f_{j+1}}{\Delta_{j+\frac{1}{2}}(\Delta_{j-\frac{1}{2}} + \Delta_{j+\frac{1}{2}})} + \frac{\Delta_{j-\frac{1}{2}} - \Delta_{j+\frac{1}{2}}}{3} f^{(3)}_j + \mathcal{O}(\Delta^2). \tag{2.35}$$

Although this approximation (Eq. (2.35)) is first-order accurate, the dominant error term is proportional to $(\Delta_{j-\frac{1}{2}} - \Delta_{j+\frac{1}{2}})$. If the adjacent meshes are of similar size, the resulting first-order error is small and the scheme essentially retains second-order accuracy. For nonuniform grids, the weights derived for the finite-differences become asymmetric. However, if the stencil used is symmetric and if no numerical viscosity is added, some refer to these schemes as central differencing in a broader sense. Based on this argument, some may call Eqs. (2.34) and (2.35) as three-point central-difference schemes with second-order accuracy, even though x_j is not exactly located at the midpoint between $x_{j-\frac{1}{2}}$ and $x_{j+\frac{1}{2}}$. As expected, we recover Eqs. (2.19) and (2.20) by setting $\Delta_{j-\frac{1}{2}} = \Delta_{j+\frac{1}{2}} = \Delta$ in Eqs. (2.34) and (2.35).

Cautionary Note on Taylor Series Expansion

The use of Taylor series expansion can provide us with difference approximation formulas and their leading order errors, which tells us the order of accuracy. The associated error is caused by the truncation of the Taylor series. It should be realized that the order of accuracy is only one measure of performance for differencing schemes. The Taylor series expansion can be very effective for smooth functions. As we have discussed above, the function needs to be smooth over the differencing stencil for the error analysis to be meaningful. It should be noted that Taylor series approximation is not almighty as it can have issues for certain types of functions even if they are smooth.

2.3.2 Polynomial Approximation

In addition to the derivation based on Taylor series expansion, finite-difference methods can also be obtained by analytically differentiating a polynomial approximation $\tilde{f}(x)$ of a function $f(x)$

$$\tilde{f}(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \cdots \tag{2.36}$$

over a stencil. There are two ways we can approximate a function with a polynomial, as illustrated in Fig. 2.6. The first approach is *curve fitting* in which case the polynomial coefficients $\{a_0, a_1, a_2, \ldots\}$ are determined to minimize the overall difference (residual) between the polynomial and the functional values at discrete spatial points as depicted in Fig. 2.6a. The least squares method is one of the common methods. The second approach is to have the polynomial pass through all discrete points, which is called *interpolation* and is illustrated in Fig. 2.6b. There are advantages and disadvantages to use one or the other. Locally speaking, interpolation satisfies $\tilde{f}(x_j) = f_j$ for all x_j. For global approximation of a function and its derivatives, curve fitting can perform better.

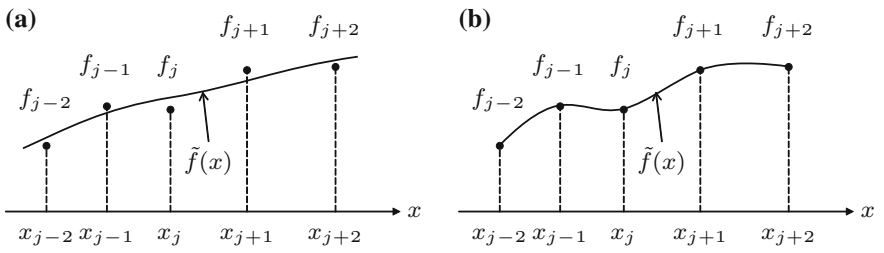

Fig. 2.6 Polynomial approximation using **a** curve fitting and **b** interpolation

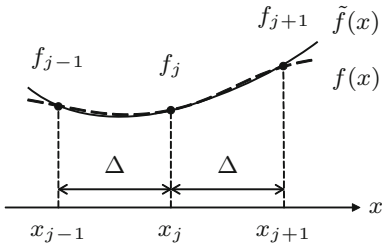

Fig. 2.7 Parabolic approximation using three points over a uniform grid

The finite-difference schemes derived from Taylor series expansion are equivalent to methods based on analytical differentiation of the interpolating polynomials. As an example, let us look at the three-point central-difference formula for a uniform grid, as shown in Fig. 2.7. Since there are three degrees of freedom over this stencil, we can use a quadratic polynomial $\tilde{f}(x) = a_0 + a_1 x + a_2 x^2$ for interpolation. Choosing this quadratic function to pass through the three points of (x_{j-1}, f_{j-1}), (x_j, f_j), and (x_{j+1}, f_{j+1}) to determine the coefficients a_0, a_1, and a_2, we find that

$$\tilde{f}(x) = f_j + \frac{-f_{j-1} + f_{j+1}}{2\Delta}(x - x_j) + \frac{f_{j-1} - 2f_j + f_{j-1}}{2\Delta^2}(x - x_j)^2. \tag{2.37}$$

Using the derivatives of $\tilde{f}(x)$ as approximations to the derivatives of $f(x)$, we now have

$$f'(x) = \frac{-f_{j-1} + f_{j+1}}{2\Delta} + \frac{f_{j-1} - 2f_j + f_{j+1}}{\Delta^2}(x - x_j) \tag{2.38}$$

$$f''(x) = \frac{f_{j-1} - 2f_j + f_{j+1}}{\Delta^2}. \tag{2.39}$$

Setting $x = x_j$, we reproduce Eqs. (2.19) and (2.20), and substituting $x = x_{j\pm1}$, we obtain the formulas for the end points, Eqs. (2.27)–(2.30). Hence we observe that the derivatives of the interpolating polynomial provide the same finite-difference formulas that are derived using Taylor series expansions.

In general, we can consider an interpolating polynomial of degree N and use its derivative to derive the finite-difference formula. The interpolating polynomial can be written as

$$\tilde{f}(x) = \sum_{k=1}^{N} \phi_k(x) f_k \tag{2.40}$$

with

$$\phi_k(x) = \frac{\pi(x)}{(x - x_k)\pi'(x_k)},$$

$$\text{where } \pi(x) = \prod_{m=1}^{N}(x - x_m), \quad \pi'(x) = \sum_{n=1}^{N}\prod_{\substack{m=1 \\ m\neq n}}^{N}(x - x_m). \tag{2.41}$$

This interpolating formulation in Eq. (2.40) is called the Lagrange interpolation. The finite-difference approximation of derivatives can be found by analytically differentiating the Lagrange interpolation function

$$\tilde{f}'(x) = \sum_{k=1}^{N} \phi_k'(x) f_k, \tag{2.42}$$

where

$$\phi_k'(x) = \frac{(x - x_k)\pi'(x) - \pi(x)}{(x - x_k)^2 \pi'(x_k)} \tag{2.43}$$

is the polynomial coefficient. Evaluating these coefficients at discrete spatial points x_j and noticing that $\pi(x_j) = 0$, the coefficients are expressed as

$$\phi_k'(x_j) = \frac{\pi'(x_j)}{(x_j - x_k)\pi'(x_k)}. \tag{2.44}$$

This procedure can be made into a subroutine so that coefficients for any stencil can be determined automatically for a finite-difference scheme with an arbitrary order of accuracy.

We mention briefly that interpolation can suffer from spatial oscillation, known as the *Runge phenomenon*. For high-order Lagrange interpolation, spatial oscillation can be observed, such as the one in the example illustrated in Fig. 2.8. The example here considers a function $f(x) = 1/(x^2 + 1)$ with uniform grid spacing for $x \in [-5, 5]$. We show here the tenth-order interpolating function for 11 discrete grid points. Although the interpolating function goes through all grid points (x_j, f_j), the interpolating function exhibits significant spatial oscillations between the grid points. Higher order interpolating functions should not be used without care because oscillations can develop. We can avoid this problem by using lower-order interpolation, least squares curve fitting, piecewise interpolation, or changing the interpolation points.

The approximation described in this subsection should not be naively used for computational fluid dynamics. Nonetheless, it is important to understand that

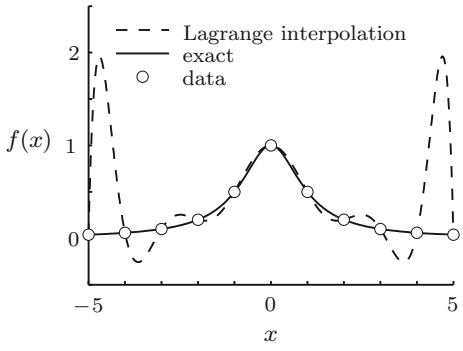

Fig. 2.8 Runge phenomenon observed for the tenth-order Lagrange interpolation of $f(x) = 1/(x^2 + 1)$ with uniform grid

finite-difference approximations can be derived by taking the analytical derivative of approximating polynomials for considering the compatibility of finite difference with analytical differentiation (as it will be mentioned in Sect. 2.3.4).

2.3.3 Central Difference at Midpoints

In this section, we derive finite-difference schemes that are appropriate for the numerical computation of fluid flow (or conservation equations in general). Let us consider a finite-difference formula for a stencil using points of $j \pm \frac{1}{2}$, $j \pm \frac{3}{2}, \cdots$ centered about x_j, as shown in Fig. 2.9. Since the functional values at midpoints are unknown, we use the second-order accurate interpolation and the Taylor series expansion in the derivation of central-difference at midpoints.

Using the values from two adjacent points of $j \pm \frac{1}{2}$, the second-order accurate interpolation and finite-difference can be formulated as

$$f_j = \frac{f_{j-\frac{1}{2}} + f_{j+\frac{1}{2}}}{2} - \frac{\Delta^2}{8} f_j'' + \mathcal{O}(\Delta^4) \tag{2.45}$$

$$f_j' = \frac{-f_{j-\frac{1}{2}} + f_{j+\frac{1}{2}}}{2} - \frac{\Delta^2}{24} f_j^{(3)} + \mathcal{O}(\Delta^4). \tag{2.46}$$

If we widen the stencil to include two additional points of $j \pm \frac{3}{2}$, we obtain the fourth-order accurate interpolation and first-derivative approximation, as well as the second-order accurate second and third-derivative approximations

$$f_j = \frac{-f_{j-\frac{3}{2}} + 9f_{j-\frac{1}{2}} + 9f_{j+\frac{1}{2}} - f_{j+\frac{3}{2}}}{16} + \frac{3\Delta^4}{128} f_j^{(4)} + \mathcal{O}(\Delta^6) \tag{2.47}$$

$$f_j' = \frac{f_{j-\frac{3}{2}} - 27f_{j-\frac{1}{2}} + 27f_{j+\frac{1}{2}} - f_{j+\frac{3}{2}}}{24\Delta} + \frac{3\Delta^4}{640} f_j^{(5)} + \mathcal{O}(\Delta^6) \tag{2.48}$$

$$f_j'' = \frac{f_{j-\frac{3}{2}} - f_{j-\frac{1}{2}} - f_{j+\frac{1}{2}} + f_{j+\frac{3}{2}}}{2\Delta^2} + \frac{5\Delta^2}{24} f_j^{(4)} + \mathcal{O}(\Delta^4) \tag{2.49}$$

$$f_j^{(3)} = \frac{-f_{j-\frac{3}{2}} + 3f_{j-\frac{1}{2}} - 3f_{j+\frac{1}{2}} + f_{j+\frac{3}{2}}}{\Delta^3} - \frac{\Delta^2}{8} f_j^{(5)} + \mathcal{O}(\Delta^4). \tag{2.50}$$

While the third derivative is not generally used in fluid flow simulations since it does not appear in the Navier–Stokes equations, Eq. (2.50) can be used in an

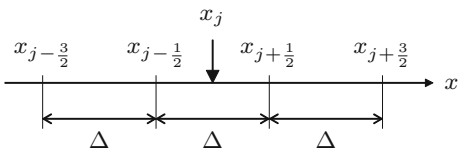

Fig. 2.9 Central difference at midpoint on uniform grid (one-dimensional)

upwinding formulation, which will be discussed later. The use of a second-derivative approximation above in Eq. (2.49) should be avoided.

For the second derivative, we should take the first-derivative finite difference of the first-derivative difference at points $j \pm \frac{1}{2}$ and $j \pm \frac{3}{2}$. By taking the finite difference twice in this manner, we obtain for the second-order accurate formulation

$$f''_j = \frac{-f'_{j-\frac{1}{2}} + f'_{j+\frac{1}{2}}}{\Delta} = \frac{f_{j-1} - 2f_j + f_{j+1}}{\Delta^2}, \tag{2.51}$$

which matches the second-order second-derivative difference formula, Eq. (2.20). For the fourth-order approximation, we obtain

$$\begin{aligned} f''_j &= \frac{f'_{j-\frac{3}{2}} - 27f'_{j-\frac{1}{2}} + 27f'_{j+\frac{1}{2}} - f'_{j+\frac{3}{2}}}{24\Delta} \\ &= \frac{f_{j-3} - 54f_{j-2} + 783f_{j-1} - 1460f_j + 783f_{j+1} - 54f_{j+2} + f_{j+3}}{(24\Delta)^2}, \end{aligned} \tag{2.52}$$

which is based on a wider seven-point stencil and is different from Eqs. (2.22) or (2.49). Although the use of a wider stencil may appear cumbersome, this scheme satisfies Eq. (2.2) or $(f')' = f''$ in a discrete sense and constitutes a compatible differencing scheme. Additional details will be offered next in Sect. 2.3.4.

For nonuniform grids, there are two techniques for deriving the finite-difference schemes as mentioned in Sect. 2.3.1. In the present discussion, we choose to map the nonuniform grid onto a uniform computation grid and construct a central-difference approximation. Further details on the implementation of nonuniform grid treatment are offered in Chaps. 3 and 4.

As one may have noticed, the sum of coefficients for interpolation is one, and the sum of coefficients for finite difference is zero. If round-off errors from interpolation and finite differencing creep into numerical calculations and cause problems, we can take advantage of the properties of the aforementioned sums. To avoid problems related to such round-off errors, we can set one of the coefficients to be 1 − (sum of other interpolation coefficients) or 0 − (sum of other differencing coefficients).

2.3.4 *Compatibility of Finite Differencing*

We mentioned that differentiation rules such as Eqs. (2.1)–(2.3) should be satisfied both in the continuous and discrete settings.[3] For the finite-difference

[3] We restate Eqs. (2.1)–(2.3) for clarity

$$\frac{\partial(fg)}{\partial x} = f\frac{\partial g}{\partial x} + \frac{\partial f}{\partial x}g, \quad \frac{\partial^2 f}{\partial x^2} = \frac{\partial}{\partial x}\left(\frac{\partial f}{\partial x}\right), \quad \text{and} \quad \frac{\partial^2 f}{\partial x \partial y} = \frac{\partial}{\partial x}\left(\frac{\partial f}{\partial y}\right) = \frac{\partial}{\partial y}\left(\frac{\partial f}{\partial x}\right).$$

.

schemes derived above, let us examine whether the analytical derivative relations are satisfied discretely.

Using the second-order differencing from Eq. (2.19) for the differentiation of a product of two functions f and g shown in Eq. (2.1), we find that

$$\frac{-(fg)_{j-1}+(fg)_{j+1}}{2\Delta} \neq f_j \frac{-g_{j-1}+g_{j+1}}{2\Delta} + \frac{-f_{j-1}+f_{j+1}}{2\Delta} g_j, \tag{2.53}$$

which tells us that the product rule does not hold discretely for the chosen differencing scheme. For the same differencing scheme from Eq. (2.19), let us also examine if Eq. (2.2) holds discretely when we apply the first-derivative finite differencing twice. We observe that

$$\begin{aligned} \frac{1}{2\Delta}\left(-\frac{-f_{j-2}+f_j}{2\Delta} + \frac{-f_j+f_{j+2}}{2\Delta}\right) &= \frac{f_{j-2}-2f_j+f_{j+2}}{4\Delta^2} \\ &\neq \frac{f_{j-1}-2f_j+f_{j+1}}{\Delta^2}, \end{aligned} \tag{2.54}$$

which is not equivalent to Eq. (2.20) that directly derived the finite-difference scheme for the second derivative. Thus, it can be said that the first-derivative finite-difference scheme about x_j is not compatible if the stencil is based on $j \pm 1, j \pm 2, \cdots$.

Revisiting this issue with the interpolation and difference operations from Eqs. (2.45) and (2.46), respectively, the finite-difference approximation of the product becomes

$$\begin{aligned} \left[\frac{\partial(fg)}{\partial x}\right]_j &= \frac{1}{2}\left\{\left[\frac{\partial(fg)}{\partial x}\right]_{j-\frac{1}{2}} + \left[\frac{\partial(fg)}{\partial x}\right]_{j+\frac{1}{2}}\right\} \\ &= \frac{1}{2}\left(\frac{-f_{j-1}g_{j-1}+f_jg_j}{\Delta} + \frac{-f_jg_j+f_{j+1}g_{j+1}}{\Delta}\right), \end{aligned} \tag{2.55}$$

which agrees with

$$\begin{aligned} \left[f\frac{\partial g}{\partial x} + \frac{\partial f}{\partial x}g\right]_j &= \frac{1}{2}\left(\left[f\frac{\partial g}{\partial x} + \frac{\partial f}{\partial x}g\right]_{j-\frac{1}{2}} + \left[f\frac{\partial g}{\partial x} + \frac{\partial f}{\partial x}g\right]_{j+\frac{1}{2}}\right) \\ &= \frac{1}{2}\left[\left(\frac{f_{j-1}+f_j}{2}\frac{-g_{j-1}+g_j}{\Delta} + \frac{-f_{j-1}+f_j}{\Delta}\frac{g_{j-1}+g_j}{2}\right)\right. \\ &\quad \left. + \left(\frac{f_j+f_{j+1}}{2}\frac{-g_j+g_{j+1}}{\Delta} + \frac{-f_j+f_{j+1}}{\Delta}\frac{g_j+g_{j+1}}{2}\right)\right]. \end{aligned} \tag{2.56}$$

The above discretization scheme exhibits compatibility for differentiation in discrete sense. We also note that Eq. (2.55) is equivalent to the left-hand side of Eq. (2.53) as shown below

$$\left[\frac{\partial(fg)}{\partial x}\right]_j = \frac{1}{\Delta}\left(-[fg]_{j-\frac{1}{2}} + [fg]_{j+\frac{1}{2}}\right)$$
$$= \frac{1}{\Delta}\left(-\frac{f_{j-1}g_{j-1} + f_j g_j}{2} + \frac{f_j g_j + f_{j+1}g_{j+1}}{2}\right). \tag{2.57}$$

However, it should be observed that, for Eqs. (2.55) and (2.56), we are not directly performing finite difference about x_j but rather interpolating the difference approximations at $x_{j\pm\frac{1}{2}}$ to determine the derivative at x_j. As discussed in Sect. 2.3.3, finite-difference formulas can be derived using polynomial approximations. From that point of view, finite-difference schemes can be thought of as being the analytical derivatives of the polynomial approximation. Therefore, if we employ the same stencil for the same point with polynomial approximation, the differentiation rules should hold discretely. As a general rule, difference schemes that satisfy $(f')' = f''$, such as Eqs. (2.51) and (2.52), should be used in practice. The satisfaction of compatibility for different spatial directions (e.g., Eq. (2.3)) will be examined in Sect. 4.1.1.

2.3.5 *Spatial Resolution*

The spatial resolution of finite-difference methods is determined by the grid spacing and the schemes themselves. Any physical phenomena with structures having length scales smaller than the grid size cannot be captured. For scales larger than the grid spacing, spatial resolution varies for different wavelengths. Waves with larger wavelengths can be represented smoothly compared to waves with shorter wavelengths on the same grid, as shown in Fig. 2.10.

The number of waves between 0 to 2π is called the wave number and has a dimension of 1/length. For a given grid size Δ, the smallest scale of fluctuation (smallest wavelength) that can be resolved is 2Δ as shown in Fig. 2.10a. This means that for a grid spacing with Δ, the maximum wave number that can be analyzed would be $k_c = \pi/\Delta$, which is referred to as the cutoff wave number. Below, we examine the consequence of performing finite-difference operation in wave space through Fourier analysis and study how accurately finite difference can represent the derivative operation. For details on Fourier transform, readers should refer to Appendix B.

Fourier Analysis of Finite Difference

Suppose we have a smooth periodic function $f(x)$ with period 2π in one dimension. Expressing the function $f(x)$ with Fourier series

$$f(x) = \sum_{k=0}^{\infty} A_k \exp(ikx), \tag{2.58}$$

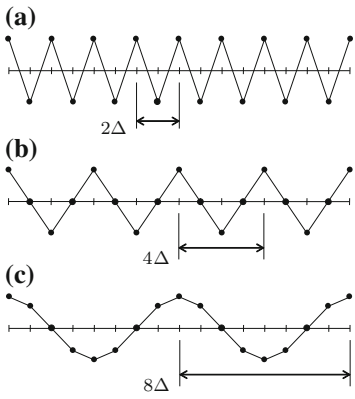

Fig. 2.10 Wavelengths that can be represented for a grid spacing of Δ. **a** Wavelength 2Δ. **b** Wavelength 4Δ. **c** Wavelength 8Δ

the derivative of the function can be expressed as

$$f'(x) = \sum_{k=1}^{\infty} ikA_k \exp(ikx), \tag{2.59}$$

where $i = \sqrt{-1}$. We can hence represent differentiation as

$$\mathcal{F}(f') = ik\mathcal{F}(f), \tag{2.60}$$

where $\mathcal{F}$ denotes the Fourier transform. This indicates that differentiation in wave space is equivalent to the multiplication of the Fourier transformed function and the wave number.

Now, let us examine finite-difference operations in wave space and compare them with the exact expression given by Eq. (2.59). For finite differencing over a uniform grid with spacing $\Delta = 2\pi/N$, we utilize the relations below based on Euler's formula $e^{\pm i\theta} = \cos\theta \pm i \sin\theta$

$$-f_{j-m} + f_{j+m} = \sum_{k=1}^{\infty} 2i \sin(mk\Delta) A_k \exp(ik\Delta j) \tag{2.61}$$

$$f_{j-m} + f_{j+m} = \sum_{k=0}^{\infty} 2\cos(mk\Delta) A_k \exp(ik\Delta j). \tag{2.62}$$

For analyzing the first-derivative finite-difference schemes in wave space, we can use Eq. (2.61) and find that the two- and four-point difference formulas, Eqs. (2.46) and (2.48), respectively, can be expressed as

$$\frac{-f_{j-\frac{1}{2}} + f_{j+\frac{1}{2}}}{\Delta} = \sum_{k=1}^{\infty} \frac{2i}{\Delta} \sin\frac{k\Delta}{2} A_k \exp(ik\Delta j) \tag{2.63}$$

$$\frac{f_{j-\frac{3}{2}} - 27f_{j-\frac{1}{2}} + 27f_{j+\frac{1}{2}} - f_{j+\frac{3}{2}}}{\Delta} = \sum_{k=1}^{\infty} \frac{i}{12\Delta}\left(27\sin\frac{k\Delta}{2} - \sin\frac{3k\Delta}{2}\right) A_k \exp(ik\Delta j). \tag{2.64}$$

Comparing Eqs. (2.63) and (2.64) with Eq. (2.59), we notice that instead of multiplying the wave number k for analytical differentiation in wave space, these finite-difference methods multiply

$$K_{(2)} = \frac{2}{\Delta}\sin\frac{k\Delta}{2} \tag{2.65}$$

$$K_{(4)} = \frac{1}{12\Delta}\left(27\sin\frac{k\Delta}{2} - \sin\frac{3k\Delta}{2}\right) \tag{2.66}$$

to the Fourier coefficients in wave space. For central-difference schemes with sixth or higher order of accuracy, same derivations can be carried out. The variable $K_{(m)}$ is called the *modified wave number* of the m-th order accurate finite-difference scheme.

Following similar procedures, we can further analyze the finite-difference schemes for second derivatives. Analytical differentiation using Fourier series is

$$f''(x) = \sum_{k=1}^{\infty} -k^2 A_k \exp(ikx), \tag{2.67}$$

which means that the second derivative can be computed by multiplying $-k^2$ to each wave number component. In other words, $\mathcal{F}(f'') = -k^2\mathcal{F}(f)$. For the difference schemes in Eqs. (2.51), (2.52), and (2.62) can be utilized to find the modified squared wave number that should correspond to k^2. For the two schemes, the modified squared wave numbers are

$$K_{(2)}^2 = \frac{2(1-\cos k\Delta)}{\Delta^2} \tag{2.68}$$

$$K_{(4)}^2 = \frac{2(730 - 783\cos k\Delta + 54\cos 2k\Delta - \cos 3k\Delta)}{24^2\Delta^2}, \tag{2.69}$$

respectively.

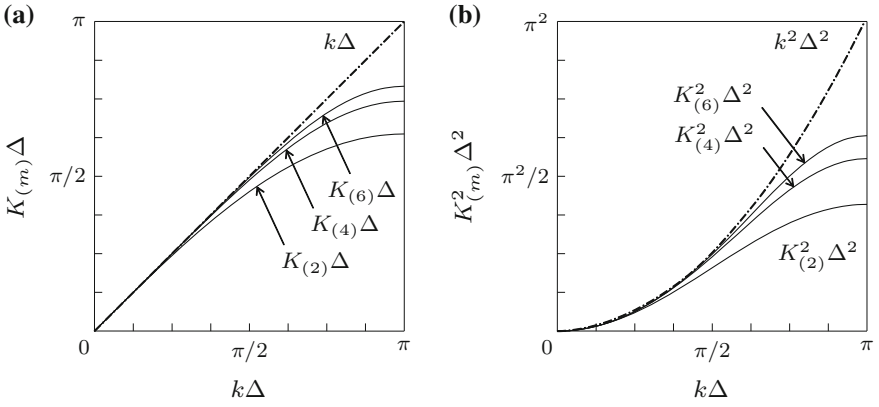

Fig. 2.11 Modified wave numbers for central-difference schemes about x_j using $j \pm 1/2$, $j \pm 3/2$, $\cdots$ ($N = 64$, $\Delta = 2\pi/N$, $k_c = 32$). **a** $K_{(m)}\Delta$. **b** $K^2_{(m)}\Delta^2 (=[K_{(m)}\Delta]^2)$

We compare these modified wave numbers with the wave numbers from the exact analysis in Fig. 2.11. For finite-difference schemes, high wave number components of the derivatives appear as if they have been dissipated (filtered out) compared to the analytically computed derivatives. Such filtering effects to reduce the effective resolution that we can achieve. If the order of accuracy of the central-difference scheme is increased, we can attain enhanced spatial resolution for high-frequency components using the same grid size. Accuracy improvement becomes relatively smaller as we increase the order of accuracy from fourth order to sixth order in comparison to what we achieve from second order to fourth order.

The compatibility of finite-difference schemes discussed in Sect. 2.3.3 can also be examined with Fourier analysis. It should be noted that discretely satisfying $f'' = (f')'$ translates to satisfying $K^2_{(m)} = [K_{(m)}]^2$ in discrete wave space. Such relation is satisfied for example with the Eqs. (2.51) and (2.52).

For comparison, let us also perform the Fourier analysis of the first-derivative finite-difference schemes about point x_j that use stencils of $j \pm 1$, $j \pm 2, \cdots$ presented in Sect. 2.3.1. For the finite-difference schemes, Eqs. (2.19) and (2.21), the modified wave numbers are

$$K_{(2)} = \frac{\sin k\Delta}{\Delta} \tag{2.70}$$

$$K_{(4)} = \frac{8\sin k\Delta - \sin 2k\Delta}{6\Delta}, \tag{2.71}$$

respectively, and are plotted in Fig. 2.12a. The modified wave numbers for the second derivatives with Eqs. (2.20) and (2.22) are

$$K^2_{(2)} = \frac{2(1 - \cos k\Delta)}{\Delta^2} \tag{2.72}$$

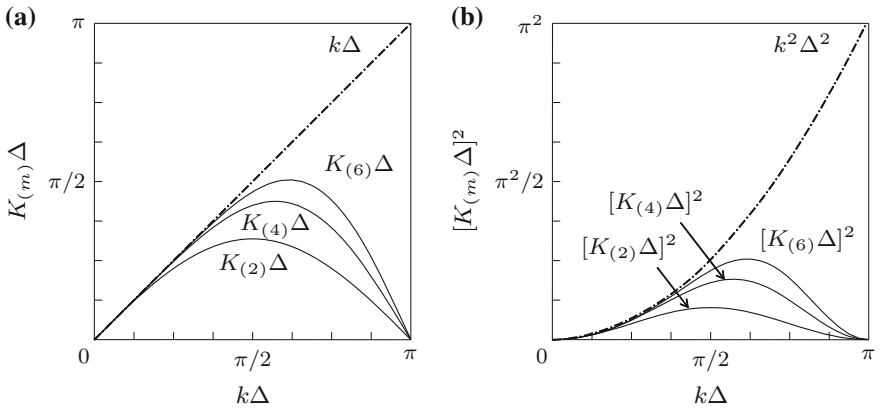

Fig. 2.12 Modified wave numbers for central-difference schemes about x_j using $j \pm 1$, $j \pm 2, \cdots$ ($N = 64$, $\Delta = 2\pi/N$, $k_c = 32$). **a** $K_{(m)}\Delta$. **b** $[K_{(m)}\Delta]^2 (\neq K^2_{(m)}\Delta^2)$

$$K^2_{(4)} = \frac{15 - 16\cos k\Delta + \cos 2k\Delta}{6\Delta^2}, \tag{2.73}$$

respectively, which are not equal to the square of the modified wave numbers shown in Fig. 2.12b. The finite-difference schemes derived in Sect. 2.3.1 result in modified wave numbers of $K^2_{(m)} \neq [K_{(m)}]^2$, implying that $f'' = (f')'$ does not hold discretely.

Based on the above analysis, we can refine the mesh or increase the order of accuracy of the finite-difference schemes to increase the spatial resolution. Widening the finite-difference stencil can increase the amount of memory allocation required on a computer, and increasing the order of accuracy increases the amount of computation at each point. Selecting which approach to follow for the pursuit of better numerical solution is dependent on the problem at hand and on the available computational resource. While high-order accurate schemes in theory can resolve smaller scales, it should be taken with care that such methods are often prone to numerical instabilities.

2.3.6 Behavior of Discretization Error

There are a variety of finite-difference approximations that one can derive as we have seen above. The determination of which finite-difference scheme to select over another is driven by the behavior of their errors and the order of accuracy of the schemes. Here, we examine the error behavior for a few finite-difference methods.

As an example, let us consider the advection equation

$$\frac{\partial f}{\partial t} + c\frac{\partial f}{\partial x} = 0 \tag{2.74}$$

for which the solution translates in the x-direction with velocity c, as illustrated in Fig. 2.1.

Now, let us take a look at the spatial discretization of the advective term. For ease of analysis, we consider a uniform mesh. Using the central-difference approximation

$$c\left.\frac{\partial f}{\partial x}\right|_j = c\frac{-\bar{f}_{j-\frac{1}{2}} + \bar{f}_{j+\frac{1}{2}}}{\Delta} - c\frac{\Delta^2}{24} f_j^{(3)} + c\mathcal{O}(\Delta^3), \tag{2.75}$$

where we will make different choices for evaluating $\bar{f}$ in discussions to follow.

If we compute $\bar{f}_{j-\frac{1}{2}}$ and $\bar{f}_{j+\frac{1}{2}}$ using interpolation with symmetric stencils about $j \pm \frac{1}{2}$:

$$\begin{aligned}
\bar{f}_{j-\frac{1}{2}} &= \frac{f_{j-1} + f_j}{2} - \frac{\Delta^2}{8} f''_{j-\frac{1}{2}} + \mathcal{O}(\Delta^4), \\
\bar{f}_{j+\frac{1}{2}} &= \frac{f_j + f_{j+1}}{2} - \frac{\Delta^2}{8} f''_{j+\frac{1}{2}} + \mathcal{O}(\Delta^4),
\end{aligned} \tag{2.76}$$

we obtain a second-order accurate central-difference scheme for the advective term

$$c\left.\frac{\partial f}{\partial x}\right|_j = c\frac{-f_{j-1} + f_{j+1}}{2\Delta} - c\frac{\Delta^2}{6} f_j^{(3)} + c\mathcal{O}(\Delta^3). \tag{2.77}$$

If we instead consider using upstream values for evaluating $\bar{f}_{j\pm\frac{1}{2}}$,

$$\begin{aligned}
\bar{f}_{j-\frac{1}{2}} &= f_{j-1} + \frac{\Delta}{2} f'_{j-\frac{1}{2}} + \mathcal{O}(\Delta^2), \\
\bar{f}_{j+\frac{1}{2}} &= f_j + \frac{\Delta}{2} f'_{j+\frac{1}{2}} + \mathcal{O}(\Delta^2),
\end{aligned} \tag{2.78}$$

the advective term is expressed as a first-order accurate one-sided discretization

$$c\left.\frac{\partial f}{\partial x}\right|_j = c\frac{-f_{j-1} + f_j}{\Delta} + c\frac{\Delta}{2} f''_j + c\mathcal{O}(\Delta^2). \tag{2.79}$$

It is also possible to use downstream values to derive a first-order accurate one-sided formula of

$$c\left.\frac{\partial f}{\partial x}\right|_j = c\frac{-f_j + f_{j+1}}{\Delta} - c\frac{\Delta}{2} f''_j + c\mathcal{O}(\Delta^2). \tag{2.80}$$

For $c > 0$, Eq. (2.79) uses information only from points upstream of x_j. Such scheme is referred to as the *upstream* or *upwind finite-difference scheme*. On the other hand Eq. (2.80) uses information only from points downstream of x_j and is referred to as the *downwind finite-difference scheme*. Downwind schemes are not generally used in simulations of fluid flows.

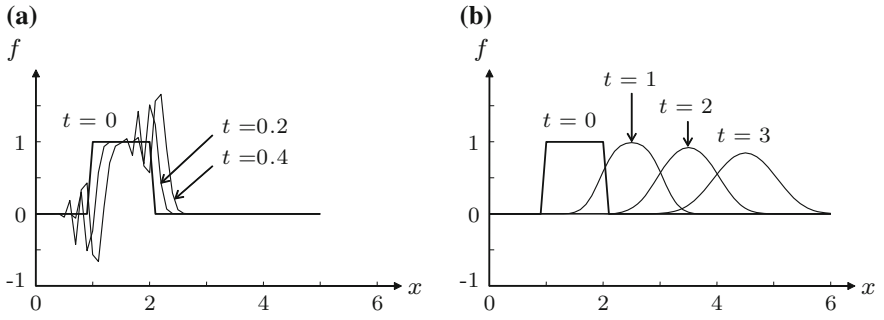

Fig. 2.13 Examples of error appearance due to differencing of the advection equation. **a** Dispersive error from central difference. **b** Diffusive error from upwind difference

For the central-difference formula, Eq. (2.77), the dominant truncation error contains an odd derivative (i.e., third derivative). In such a case, the error generally behaves in a dispersive manner. For the upwind difference formula, Eq. (2.79), the dominant truncation error contains an even derivative (i.e., second derivative), which exhibits a diffusive behavior. When using information from downstream, as it is the case for Eq. (2.80), the error behaves diffusively but with negative diffusivity. In this case, the gradients in the solution becomes steeper as time advances, which eventually causes the solution to blow up from numerical instability. While the exact solution to the advection equation, Eq. (2.74), maintains the solution profile under translation, central differencing introduces spatial oscillations leading to numerical instability and upwind differencing diffuses the solution over time. These error behaviors are illustrated in Fig. 2.13.

Let us further examine the effect of using the upwind finite-difference method

$$\left.\frac{\partial f}{\partial t}\right|_j + c\frac{-f_{j-1} + f_j}{\Delta} = 0 \tag{2.81}$$

by substituting the Taylor series expansion, Eq. (2.15), into f_{j-1}. We accordingly find that Eq. (2.81) becomes

$$\frac{\partial f}{\partial t} + c\frac{\partial f}{\partial x} - \frac{c\Delta}{2}\frac{\partial^2 f}{\partial x^2} + c\mathcal{O}(\Delta^2) = 0. \tag{2.82}$$

Note that we started with a pure advection equation, Eq. (2.74), and did not include any diffusive effects. By employing the upwind difference, Eq. (2.82) now contains a diffusive term $\frac{c\Delta}{2}\frac{\partial^2 f}{\partial x^2}$ with diffusivity of $c\Delta/2$. In contrast to physical diffusivity caused by viscous diffusion, such numerical effect due to truncation error is referred to as *numerical diffusion*. When this effect appears in the momentum equation, numerical diffusivity is called *numerical viscosity*. With the use of the upwinding difference method, the solution profile becomes smoothed out over time due to numerical diffusion, as shown in Fig. 2.13b (see also Sects. 3.5.3 and 4.7.2 for use of upwind difference schemes).

We can also consider the case with viscous diffusion present

$$\frac{\partial f}{\partial t} + c\frac{\partial f}{\partial x} - a\frac{\partial^2 f}{\partial x^2} = 0. \tag{2.83}$$

Discretizing the second term on the left-hand side with first-order accurate upwind difference (and the third term with a second-order or higher scheme), what essentially is solved becomes

$$\frac{\partial f}{\partial t} + c\frac{\partial f}{\partial x} - \left(a + \frac{c\Delta}{2}\right)\frac{\partial^2 f}{\partial x^2} + c\mathcal{O}(\Delta^2) = 0. \tag{2.84}$$

For cases where the advection speed c and the grid size Δ are large in their magnitudes, it is possible to have $c\Delta/2 \gg a$ in which case the numerical diffusion overwhelms physical diffusion. If the contribution from numerical diffusion is large in the momentum equation, this can cause the solution to become insensitive to viscous diffusion.

In the above discussion, we have focused on the influence of the second and third derivatives from the truncation error on the numerical solution of the advection equation. Let us further generalize the analysis and consider the influence of even and odd derivatives on the behavior of errors. For the ease of analysis, we consider the solution to be periodic over $x \in [0, 2\pi]$. First, the solution to original advection equation can be expressed in a separable form as

$$f(x,t) = \sum_{k=0}^{\infty} \widetilde{f}_k(t)\exp(ikx), \tag{2.85}$$

where we denote the wave number as k. Substituting this expression in the advection equation $\frac{\partial f}{\partial t} + c\frac{\partial f}{\partial x} = 0$, we find

$$\frac{\partial \widetilde{f}_k}{\partial t} = -ikc\widetilde{f}_k, \tag{2.86}$$

for each wave number. Solving the above differential equation gives the exact solution of

$$f(x,t) = \sum_{k=0}^{\infty} \widetilde{f}_k(0)\exp[ik(x-ct)]. \tag{2.87}$$

Here, $\widetilde{f}_k(0)$ can be determined from the initial condition.

Next, let us examine the influence of truncation error for the advection equation. In reality, the discretized advection equation results in

$$\frac{\partial f}{\partial t} + c\frac{\partial f}{\partial x} = \alpha_2\frac{\partial^2 f}{\partial x^2} + \alpha_3\frac{\partial^3 f}{\partial x^3} + \alpha_4\frac{\partial^4 f}{\partial x^4} + \cdots, \tag{2.88}$$

where $\alpha_2, \alpha_3, \alpha_4, \ldots$ are dependent on the chosen finite-difference schemes and the grid spacing Δ. Substituting the separable form of f from Eq. (2.85) into Eq. (2.88), for each wave number, we find

$$\frac{\partial \widetilde{f}_k}{\partial t} = \left[-ik(c + k^2\alpha_3 - k^4\alpha_5 + \cdots) + (-k^2\alpha_2 + k^4\alpha_4 - \cdots)\right] \widetilde{f}_k. \tag{2.89}$$

We find that the solution to the above equation is

$$f(x,t) = \sum_{k=0}^{\infty} \widetilde{f}_k(0) \exp\left[ik\left\{x - (c + k^2\alpha_3 - k^4\alpha_5 + \cdots)t\right\}\right] \times \exp\left[(-k^2\alpha_2 + k^4\alpha_4 - \cdots)t\right]. \tag{2.90}$$

Comparing Eqs. (2.87) and (2.90), we notice that extra terms appear in the numerical solution. We observe that the advection speed c is modified to be $(c + k^2\alpha_3 - k^4\alpha_5 + \cdots)$ with truncation error. The error is generated when we have nonzero α_j for odd j. Hence, we see that the phase speed is altered by the inclusion of the odd-derivative terms in the truncation error. The profile of a propagating wave can be influenced by this wave number-dependent phase-speed error, leading to the wave becoming dispersive. Furthermore, $(-k^2\alpha_2 + k^4\alpha_4 - \cdots)$ is added as a component to the solution which alters the growth or decay of the solution. For $\alpha_2 > 0$, we introduce numerical diffusion from the second derivative and likewise for $\alpha_4 < 0$ from the fourth derivative. Numerical diffusion can be added to the governing equation by having even-derivative terms.

When we conduct fundamental research using numerical simulation, the aforementioned numerical errors should be kept as low as possible. Industrial or commercial software often incorporates upwind difference schemes intentionally to allow for simulations of a wide variety of flows in a numerically stable manner. When using these softwares, we must understand that numerical viscosity may influence the outcome of the solution. Simulations of fluid flow cannot be trusted if the discretization error displays unphysical behavior. The question that we should pose is: How do we verify our numerical solver? What needs to be addressed is how the numerical error possibly affects the flow physics. It is difficult to predict how the solution may be influenced by the error a priori, but we can focus on the fact that error is driven by how fine the grid spacing is, and examine how the error responds to the change in grid resolution. To state that a method has been verified (converged), the result must be shown not to change when the grid size is altered. We must at least be able to display the trend of error when the grid is refined.

The error discussed here is based on the truncation error from spatial discretization and does not consider the error from time integration (stepping) of the differential equations. Time stepping methods and the associated temporal discretization errors are discussed below.

2.4 Time Stepping Methods

The advection-diffusion equation is a time evolution equation. The time rate of change of the function f can be determined once we evaluate the advective and diffusive terms. Time derivative of f can be numerically integrated in time to find the solution. In this section, we present the time stepping (integration) methods for an example of the advection-diffusion equation written as

$$\frac{\partial f}{\partial t} = g(t, f). \tag{2.91}$$

Here, we represent the spatial derivative terms by g (advective and diffusive terms), the time step by Δt, and the discrete time levels by $t_n = n\Delta t$. The number of time steps is denoted by n and we use a superscript on f to indicate the time level (i.e., $f^n = f(t_n)$). In what follows, we assume that the flow field up to time t_n is known. In other words, f^n, f^{n-1}, $f^{n-2}, \cdots$ and g^n, g^{n-1}, $g^{n-2}, \cdots$ are taken to be known a priori for determining f^{n+1}. Details on time stepping schemes can also be found in Lambert [2].

2.4.1 Single-Step Methods

First, let us discuss single-step time stepping methods that use only data from t_n to compute the solution at t_{n+1}. Generalizing the trapezoidal rule for integrating Eq. (2.91) in time, we have

$$\frac{f^{n+1} - f^n}{\Delta t} = (1 - \alpha)g^n + \alpha g^{n+1}, \tag{2.92}$$

where $0 \leq \alpha \leq 1$. For $\alpha = 0$, the next solution f^{n+1} is determined from the known f^n and g^n with $f^{n+1} = f^n + \Delta g^n$. Such method that computes the solution at the next time level using information from known time levels is called an *explicit method*. For $0 < \alpha \leq 1$, finite-difference discretization of the differential equation leads to an equation to be solved for f^{n+1}. Such method that computes the solution at the next time level using information from both known and future time levels is called an *implicit method*.

When we set $\alpha = 0$ in Eq. (2.92), we obtain the *explicit Euler's method*

$$f^{n+1} = f^n + \Delta t g^n. \tag{2.93}$$

Using $\alpha = 1$ in Eq. (2.92), we obtain the *implicit Euler's method*

$$f^{n+1} = f^n + \Delta t g^{n+1}. \tag{2.94}$$

Both of these Euler's methods, Eqs. (2.93) and (2.94), are first-order accurate in time. Setting $\alpha = 1/2$, we derive the *Crank–Nicolson method*

$$f^{n+1} = f^n + \Delta t \frac{g^n + g^{n+1}}{2}, \tag{2.95}$$

which is second-order accurate in time. The parameter α is generally set to 0, 1/2, or 1. It is rare to find other values of α being used.

Consider an example of implicitly solving the diffusion equation, Eq. (2.8), with second-order accurate central differencing. We can write

$$\frac{f_j^{n+1}}{\Delta t} - \alpha a \frac{f_{j-1}^{n+1} - 2f_j^{n+1} + f_{j+1}^{n+1}}{\Delta^2} = \frac{f_j^n}{\Delta t} + (1-\alpha) a \frac{f_{j-1}^n - 2f_j^n + f_{j+1}^n}{\Delta^2}, \tag{2.96}$$

which results in an algebraic equation with a tridiagonal coefficient matrix on the left-hand side for f^{n+1}. Implicit methods require solving an equation such as the one above, which can be linear or nonlinear depending on the form of g. Solving such system of equations adds computational time per time step. As discussed later, explicit methods can be prone to numerical instability and often needs the time step to be set small, which in turn increases the number time steps for computations. Whether an explicit or implicit scheme is better suited is dependent on the governing equation and how the mesh is created. For terms found in equations of fluid dynamics, linear terms are suited for implicit schemes. On the other hand, the nonlinear advective term, which is often a source of instability, is not suited for implicit treatment due to the added cost of solving the resulting nonlinear equation.

Predictor-Corrector Methods

There are methods that possess the ease of use of explicit methods and the stability property similar to that of implicit methods. These methods are called *predictor-corrector methods* and achieve the benefits by introducing intermediate steps within a single time advancement to carry out prediction and correction of the time-integrated solution.

For example, we can use the explicit Euler's method as a prediction and use the predicted solution $\tilde{f}$ as a correction to compute $\tilde{g} = g(t_{n+1}, \tilde{f})$ for the right-hand side of Eq. (2.94)

$$\left.\begin{aligned} \tilde{f} &= f^n + \Delta t g^n \\ f^{n+1} &= f^n + \Delta t \tilde{g} \end{aligned}\right\} \tag{2.97}$$

which is known as Matsuno's method in the field of weather forecasting. One can also construct a predictor–corrector method that is similar to the Crank–Nicolson method

$$\left.\begin{aligned} \tilde{f} &= f^n + \Delta t g^n \\ f^{n+1} &= f^n + \Delta t \frac{g^n + \tilde{g}}{2} \end{aligned}\right\} \tag{2.98}$$

which is called Heun's method. Since the form of Eq. (2.98) requires memory allocation for g^n, one can use the equivalent form of

$$f^{n+1} = \frac{1}{2}(f^n + \tilde{f} + \Delta t \tilde{g}) \tag{2.99}$$

that uses less memory allocation.

Runge–Kutta Methods

We can further extend upon the above concept to introduce multiple predictions per time step in an explicit fashion, which is the basis for a class of time stepping schemes, called the *Runge–Kutta methods*. For example, the two-step Runge–Kutta method is

$$\left.\begin{aligned} f^{(1)} &= f^n + \frac{\Delta t}{2} g^n \\ f^{n+1} = f^{(2)} &= f^n + \Delta t g^{(1)} \end{aligned}\right\} \tag{2.100}$$

where m in $f^{(m)}$ denotes the level of the intermediate step. This scheme in Eq. (2.100) has second-order temporal accuracy.

The second-order time integration methods introduced up until now use some form of approximation to the average value between time level t^n and t^{n+1} of the right-hand side of the evolution equation. We can summarize them as

- Crank−Nicolson method $$f^{n+1} = f^n + \Delta t \frac{g^n + g^{n+1}}{2} \tag{2.101}$$
- Heun's method $$f^{n+1} = f^n + \Delta t \frac{g^n + \tilde{g}}{2} \tag{2.102}$$
- 2nd-order Runge−Kutta method $$f^{n+1} = f^n + \Delta t g^{n+1/2} \tag{2.103}$$

where $g^{n+1/2} = g(t_n + \frac{\Delta t}{2}, f_n + \frac{\Delta t}{2} g^n)$.

Another widely used time stepping method is the classic *four-step Runge–Kutta method*

$$\left.\begin{aligned} f^{(1)} &= f^n + \frac{\Delta t}{2} g^{(1)} \\ f^{(2)} &= f^n + \frac{\Delta t}{2} g^{(2)} \\ f^{(3)} &= f^n + \Delta t g^{(2)} \\ f^{n+1} = f^{(4)} &= f^n + \Delta t \frac{g^n + 2g^{(1)} + 2g^{(2)} + g^{(3)}}{6} \end{aligned}\right\} \tag{2.104}$$

which is fourth-order accurate in time. This four-step method uses the Euler prediction, Euler correction, leapfrog prediction, and Milne correction within a single integration step.

The above formulation necessitates f^n, g^n, as well as $g^{(1)}$, $g^{(2)}$, and $g^{(3)}$ to be stored during computation. We note that there are also low-storage Runge–Kutta methods have been proposed by Williamson [6] and have been used widely to achieve high-order temporal accuracy with reduced memory consumption.

2.4.2 Multi-Step Methods

There are time integration methods that use the current state g^n along with the past data g^{n-1}, g^{n-2}, $\cdots$ in an explicit formulation. These methods are referred to as multi-step methods and the *Adams–Bashforth methods* are widely used.

Consider expanding f^{n+1} about f^n with Taylor series and substitute g into $\partial f/\partial t$ to find

$$\begin{aligned} f^{n+1} &= f^n + \Delta t \left.\frac{\partial f}{\partial t}\right|^n + \frac{\Delta^2}{2}\left.\frac{\partial^2 f}{\partial t^2}\right|^n + \frac{\Delta^3}{6}\left.\frac{\partial^3 f}{\partial t^3}\right|^n + \cdots \\ &= f^n + \Delta t g^n + \frac{\Delta^2}{2}\left.\frac{\partial g}{\partial t}\right|^n + \frac{\Delta^3}{6}\left.\frac{\partial^2 g}{\partial t^2}\right|^n + \cdots \end{aligned} \tag{2.105}$$

Truncating the series at the second term yields the first-order Adams–Bashforth method which is the explicit Euler's method, Eq. (2.93). Retaining the expansion up to the third term and inserting

$$\left.\frac{\partial g}{\partial t}\right|^n = \frac{g^n - g^{n-1}}{\Delta t} \tag{2.106}$$

provides the second-order Adams–Bashforth method:

$$f^{n+1} = f^n + \Delta t \frac{3g^n - g^{n-1}}{2}. \tag{2.107}$$

If we keep the expansion up to the fourth-order term in the Taylor series and substitute

$$\left.\frac{\partial g}{\partial t}\right|^n = \frac{3g^n - 4g^{n-1} + g^{n-2}}{2\Delta t}, \quad \left.\frac{\partial^2 g}{\partial t^2}\right|^n = \frac{g^n - 2g^{n-1} + g^{n-2}}{\Delta t^2}, \tag{2.108}$$

we arrive at the third-order Adams–Bashforth method:

$$f^{n+1} = f^n + \Delta t \frac{23g^n - 16g^{n-1} + 5g^{n-2}}{12}. \tag{2.109}$$

Similar derivations can be followed to find higher-order Adams–Basthforth methods.

In order to initiate time integration with Adams–Bashforth methods with second or higher temporal accuracy, one must have the initial condition f_0 and g_0 as well

as past information g^{-1}, g^{-2}, $\cdots$, which may not be available. Thus, an alternate time integration scheme needs to be implemented for the first few steps until all information is gathered for the Adams–Bashforth method to start performing the computation. If the computation is particularly sensitive to the initial condition or transients, it may be desirable to have another high-order accurate method selected initially, instead of using a low-order accurate Adams–Bashforth method.

2.5 Stability Analysis

Once the advection-diffusion equation is discretized in space and time, we can determine the solution over time through numerical integration. However, there can be situations where numerical instability contaminates the solution during the calculation. In some cases, oscillations due to numerical instability can increase the amplitude of the numerical solution, eventually making its value larger than the maximum realizable number on a computer, which is known as overflow. In order to avoid such issues, we examine the stability properties of a few numerical solvers in this section.

The analysis here is motivated by the need to examine the stability of numerical solvers for the advection-diffusion equation

$$\frac{\partial f}{\partial t} + c\frac{\partial f}{\partial x} = a\frac{\partial^2 f}{\partial x^2}. \tag{2.110}$$

By assuming the solution to be of the form

$$f(x,t) = \sum_{k=0}^{\infty} \widetilde{f_k}(t)\exp(ikx), \tag{2.111}$$

we find for each wave number k,

$$\frac{\mathrm{d}\widetilde{f_k}}{\mathrm{d}t} = (-k^2 a - ikc)\widetilde{f_k}. \tag{2.112}$$

We can then express the above equation with a mathematical abstraction of

$$\frac{\mathrm{d}f}{\mathrm{d}t} = \lambda f \tag{2.113}$$

as a linear model problem by setting $\lambda = (-k^2 a - ikc)$. Note that we have removed the subscript k and tilde to generalize the problem formulation. For the advection-diffusion equation, we have

$$\mathrm{Re}(\lambda) = -k^2 a \quad \text{and} \quad \mathrm{Im}(\lambda) = -kc, \tag{2.114}$$

which means that the real part of λ represents the effect of diffusion and the imaginary part constitutes the effect of advection.

In this section, we first analyze the stability characteristics of time stepping methods. We then examine the stability of finite-difference methods (combined effect of temporal and spatial discretization) using the *von Neumann stability analysis*. Strictly speaking, the analysis is only valid for linear problems and assumes that Fourier analysis is applicable. Since the actual governing equations for fluid mechanics are nonlinear, the analysis cannot be directly applied. Nonetheless, the results from the von Neumann analysis provide us with great insight into the numerical stability of finite-difference methods and can be used as a guideline for choosing an appropriate size of time step Δt.

2.5.1 Stability of Time Stepping Methods

For characterizing the stability of the time stepping methods, we consider a linear model problem of

$$\frac{\mathrm{d}f}{\mathrm{d}t} = \lambda f \quad \text{with} \quad f(t_0) = 1, \tag{2.115}$$

where we let λ be complex and f to be a function of time only. This model problem can capture the fundamental behavior of most differential equations, including the advection-diffusion equation, Eq. (2.110), which was briefly described in terms of this model problem in Sect. 2.5. The solution to this model problem is

$$f(t) = e^{\lambda t} = e^{\mathrm{Re}(\lambda)t}[\cos(\mathrm{Im}(\lambda)t) + i\,\sin(\mathrm{Im}(\lambda)t)]. \tag{2.116}$$

This expression tells us that the solution is stable (i.e., $\lim_{t\to 0} |f(t)| = 0$) for $\mathrm{Re}(\lambda) < 0$ and the solution is unstable for $\mathrm{Re}(\lambda) > 0$ (solution is neutrally stable for $\mathrm{Re}(\lambda) = 0$). The imaginary component of λ denotes the frequency at which the solution oscillates. Thus, we can observe that $\mathrm{Re}(\lambda)$ and $\mathrm{Im}(\lambda)$ capture the diffusive and advective behaviors of the flow, respectively.

First, let us consider the stability of the explicit Euler's method. For this model problem, we have

$$f^{n+1} = (1 + \lambda\Delta t) f^n = (1 + \lambda\Delta t)^n f^0. \tag{2.117}$$

For this method to be stable, we need

$$|1 + \lambda\Delta t| < 1 \qquad \text{(Explicit Euler's method)}. \tag{2.118}$$

Hence, for any value of $\lambda\Delta t$ that lies within a unit circle centered at -1 on the complex plane, the method is stable.

We can perform a similar analysis for the implicit Euler's method to find that

$$f^{n+1} = f^n + \Delta t \lambda f^{n+1} \quad \Rightarrow \quad f^{n+1} = (1 - \lambda \Delta t)^{-1} f^n, \tag{2.119}$$

which requires

$$\left|(1 - \lambda \Delta t)^{-1}\right| < 1 \quad \Rightarrow \quad |\lambda \Delta t - 1| > 1 \quad \text{(Implicit Euler's method)} \tag{2.120}$$

which means that for $\lambda \Delta t$ that is outside of a unit circle centered at 1 on the complex plane, the method is stable.

Let us also consider the Crank–Nicolson method where we have

$$f^{n+1} = f^n + \Delta t \frac{1}{2} \lambda (f^n + f^{n+1}) \quad \Rightarrow \quad f^{n+1} = \frac{1 + \frac{1}{2}\lambda \Delta t}{1 - \frac{1}{2}\lambda \Delta t} f^n, \tag{2.121}$$

which necessitates

$$\left|\frac{1 + \frac{1}{2}\lambda \Delta t}{1 - \frac{1}{2}\lambda \Delta t}\right| < 1 \quad \Rightarrow \quad \text{Re}(\lambda \Delta t) < 0 \quad \text{(Crank}-\text{Nicolson method)} \tag{2.122}$$

for stability. This means that the Crank–Nicolson method is stable if $\lambda \Delta t$ is anywhere on the left-half plane of the complex plane.

Note that the analysis can be performed for multistep and predictor–corrector methods also. For example, the second-order Runge–Kutta method gives

$$f^{n+1} = \left[1 + \lambda \Delta t + \frac{(\lambda \Delta t)^2}{2}\right] f^n \tag{2.123}$$

and correspondingly

$$\left|1 + \lambda \Delta t + \frac{(\lambda \Delta t)^2}{2}\right| < 1 \quad \text{(2nd-order Runge}-\text{Kutta method)} \tag{2.124}$$

is needed for stability.

We graphically summarize the stability region in Fig. 2.14 for the explicit and implicit Euler's methods, the Crank–Nicolson method, and the leapfrog method (discussed later in Sect. 2.5.7). The stability contours for predictor-corrector schemes and multistep methods, namely the Runge–Kutta methods and the Adams–Bashforth methods are also shown.

There are several observations we can make about the time stepping methods. First, the Crank–Nicolson method has the entire left-hand side of the complex plane of $\lambda \Delta t$ covered for stability. The leapfrog method is only stable for purely oscillatory (advective) problems. It is stable only along the imaginary axis and would not be able to handle diffusive effects. We can also note that the Runge–Kutta methods can

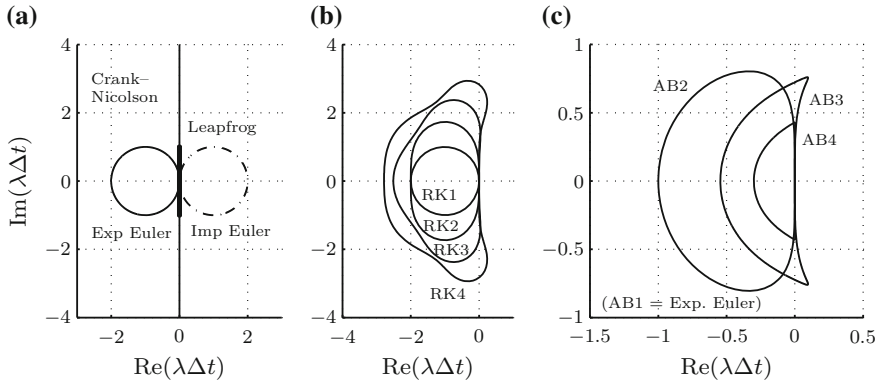

Fig. 2.14 Stability contours of time stepping methods. **a** Stability regions for the explicit Euler's method is inside the circle (*solid*); the implicit Euler is outside the *circle* (*dash-dotted*); the Crank–Nicolson method is the entire *left-hand side* of the complex plane; the leapfrog method is stable along the *thick line*. **b** First to fourth-order Runge–Kutta (RK) methods are stable inside the stability contours. **c** Second to fourth-order Adams–Bashforth (AB) methods are stable inside the contours

enlarge the region of stability by increasing the order of accuracy. This is of course achieved with the added cost of computation required by the use of intermediate steps. For the Adams–Bashforth methods, the scales on the axes used in Fig. 2.14c are different for those in Fig. 2.14a, b. As the temporal accuracy of the Adams–Bashforth method is increased, the region of stability becomes smaller.

If advection is a important feature in the flow, we should choose a time stepping method that covers the imaginary axis. Schemes such as the third-order Adam–Bashforth and Runge–Kutta methods, and their higher order versions cover the imaginary axis as shown in Fig. 2.14b, c. On the other hand, if diffusion is a significant feature in the flow, we should select a scheme that enlarges the stability region in the left-hand side of the complex plane.

2.5.2 *von Neumann Analysis*

Let us now consider $f(x,t)$ to be a function of space and time and expand it with a Fourier series in the spatial direction (assuming that the variables x and t can be separated)

$$f(x,t) = \sum_{k=0}^{\infty} A_k(t) \exp(ikx), \tag{2.125}$$

where k is the wave number. The coefficients $A_k(t)$ are taken to be functions of time. For a scheme to be numerically stable, we require the amplitude $A_k(t)$ to be bounded over time. For linear problems, we can examine the stability of the amplitude for each wave number k, instead of considering the sum of all wave number components.

For a discrete point $x_j = \Delta j$ and time $t_n = n\Delta t$, we let

$$f_j^n = A^n \exp(ik\Delta j). \tag{2.126}$$

Using identities such as Eqs. (2.61) and (2.62) in analyzing finite-difference schemes, we can find the amplitude $A = A^{n+1}/A^n$. Stability of numerical methods are classified as

$$|A| \begin{cases} < 1 & \text{stable} \\ = 1 & \text{neutrally stable} \\ > 1 & \text{unstable} \end{cases} \tag{2.127}$$

If a numerical method is stable for all wave numbers, then that method is *unconditionally stable*. Below we present what is referred to as the *von Neumann stability analysis* of various discretization schemes.

2.5.3 Stability of the Discrete Advection Equation

Here, we consider the forward-in-time central-in-space (FTCS) method for the advection equation, Eq. (2.7), with a constant advection speed of c

$$\frac{f_j^{n+1} - f_j^n}{\Delta t} = -c\frac{-f_{j-1}^n + f_{j+1}^n}{2\Delta}. \tag{2.128}$$

This method discretizes the evolution equation in time with explicit Euler's method (forward in time) and in space with central difference (central in space). Fourier transform of the above equation leads to the evolution equation for the amplitude of the solution

$$\frac{A^{n+1} - A^n}{\Delta t} = -ic\frac{\sin(k\Delta)}{\Delta}A^n. \tag{2.129}$$

Here, we have used Eq. (2.61) for the right-hand side. The amplitude is found to be a complex number of

$$A = 1 - i\frac{c\Delta t}{\Delta}\sin(k\Delta). \tag{2.130}$$

We can observe that unless $\Delta t = 0$, clearly $|A| > 1$. Therefore, the FTCS method for the advection equation is unconditionally unstable.

It is possible to find a stable discretization of the advection equation with an explicit method. Replacing f_j^n in the left-hand side of Eq. (2.128) with a spatial average $\bar{f}_j^n = (f_{j-1}^n + f_{j+1}^n)/2$, we can consider

$$\frac{f_j^{n+1} - \bar{f}_j^n}{\Delta t} = -c\frac{-f_{j-1}^n + f_{j+1}^n}{2\Delta}, \tag{2.131}$$

which is called the *Lax–Friedrichs method*. Using Eqs. (2.61) and (2.62), we find

$$\frac{A^{n+1} - A^n \cos(k\Delta)}{\Delta t} = -ic\frac{\sin(k\Delta)}{\Delta}A^n, \tag{2.132}$$

which yields

$$|A| = \left\{1 + \left[\left(\frac{c\Delta t}{\Delta}\right)^2 - 1\right]\sin^2(k\Delta)\right\}^{1/2}. \tag{2.133}$$

We now notice that

$$C \equiv \frac{c\Delta t}{\Delta} \leq 1 \tag{2.134}$$

must be satisfied for $|A| \leq 1$ so that the scheme is stable for all wave numbers. This condition is known as the *Courant–Friedrichs–Lewy* (*CFL*) *condition*, and tells us that the information traveling at speed c over a time of Δt must not translate farther than Δ. The ratio of the distance traveled by the information on a spatial grid is defined as $C \equiv c\Delta t/\Delta$, the *Courant number* (*CFL number*), and is a very important non-dimensional number in evaluating the numerical stability of solvers for the advection equation.

If we consider the implicit Euler's method

$$\frac{f_j^{n+1} - f_j^n}{\Delta t} = -c\frac{-f_{j-1}^{n+1} + f_{j+1}^{n+1}}{2\Delta} \tag{2.135}$$

then the amplitude for wave number k satisfies

$$\frac{A^{n+1} - A^n}{\Delta t} = -ic\frac{\sin(k\Delta)}{\Delta}A^{n+1} \tag{2.136}$$

resulting in

$$A = \left[1 + i\frac{c\Delta t}{\Delta}\sin(k\Delta)\right]^{-1}. \tag{2.137}$$

Hence, it can be seen that Euler's implicit method always satisfies $|A| < 1$ for all wave numbers and is unconditionally stable. Real fluid flows however have advective velocity that is varying in space and time making the governing equation nonlinear. This makes implementation of implicit methods difficult or computationally expensive, due to the necessity to solve a nonlinear equation at every time step. For instance, the nonlinear Burgers' equation, Eq. (2.5), would be discretized with the implicit Euler's equation as

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = -u_j^{n+1}\frac{-u_{j-1}^{n+1} + u_{j+1}^{n+1}}{2\Delta}, \tag{2.138}$$

which requires a nonlinear solver. Such formulation can become computationally expensive for large-scale problems. On the other hand, we can consider a semi-implicit method, such as

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = -u_j^n \frac{-u_{j-1}^{n+1} + u_{j+1}^{n+1}}{2\Delta}. \tag{2.139}$$

From a practical point of view, it is often appropriate to select an explicit method for the advective term that has stability and high order of accuracy.

2.5.4 Stability of the Discrete Diffusion Equation

Let us consider the FTCS method for the diffusion equation, Eq. (2.8), with diffusivity $a > 0$,

$$\frac{f_j^{n+1} - f_j^n}{\Delta t} = a \frac{f_{j-1}^n - 2f_j^n + f_{j+1}^n}{\Delta^2}. \tag{2.140}$$

If we analyze the time evolution of the amplitude with Fourier transform, we observe that

$$\frac{A^{n+1} - A^n}{\Delta t} = 2a \frac{\cos(k\Delta) - 1}{\Delta^2} A^n \tag{2.141}$$

and the amplitude A is found to be

$$A = 1 - \frac{4a\Delta t}{\Delta^2} \sin^2 \frac{k\Delta}{2}. \tag{2.142}$$

For this scheme to be stable ($|A| \leq 1$), we have

$$\frac{4a\Delta t}{\Delta^2} \sin^2 \frac{k\Delta}{2} \leq 2. \tag{2.143}$$

This inequality holds for all wave numbers if

$$\Delta t \leq \frac{\Delta^2}{2a} \tag{2.144}$$

because $0 \leq \sin^2(k\Delta/2) \leq 1$. We can further draw an analogy to the CFL condition by considering a/Δ as the characteristic diffusion speed and express

$$\frac{(a/\Delta)\Delta t}{\Delta} \leq \frac{1}{2}. \tag{2.145}$$

The above analysis means that the diffusion equation can be integrated stably with FTCS by choosing the time step according to Eq. (2.144). However, it should be noted that the time step Δt needs to be selected such that it is proportional to Δ^2. That is, if we choose the spatial discretization to be reduced by half, the time step must be reduced by a quarter. This discretization scheme is thus at a disadvantage for finer grids because of the larger number of steps required for time advancement.

We can switch the time integration scheme to be implicit and show that the diffusion equation can be discretized in a stable manner for all wave numbers. Utilizing the implicit Euler's method, the diffusion equation can be discretized as

$$\frac{f_j^{n+1} - f_j^n}{\Delta t} = a \frac{f_{j-1}^{n+1} - 2f_j^{n+1} + f_{j+1}^{n+1}}{\Delta^2} \tag{2.146}$$

with the corresponding equation for the amplitude being

$$\frac{A^{n+1} - A^n}{\Delta t} = 2a \frac{\cos(k\Delta) - 1}{\Delta^2} A^{n+1}. \tag{2.147}$$

In this case, the amplitude becomes

$$A = \left(1 + \frac{4a\Delta t}{\Delta^2} \sin^2 \frac{k\Delta}{2}\right)^{-1}, \tag{2.148}$$

which satisfies $|A| \leq 1$ for all wave numbers.

If we choose the Crank–Nicolson method, the diffusion equation becomes

$$\frac{f_j^{n+1} - f_j^n}{\Delta t} = \frac{a}{2} \frac{f_{j-1}^n - 2f_j^n + f_{j+1}^n}{\Delta^2} + \frac{a}{2} \frac{f_{j-1}^{n+1} - 2f_j^{n+1} + f_{j+1}^{n+1}}{\Delta^2} \tag{2.149}$$

with

$$A = \left(1 - \frac{2a\Delta t}{\Delta^2} \sin^2 \frac{k\Delta}{2}\right) \left(1 + \frac{2a\Delta t}{\Delta^2} \sin^2 \frac{k\Delta}{2}\right)^{-1}, \tag{2.150}$$

which also satisfies $|A| \leq 1$ for all wave numbers. Hence, the implicit Euler's and Crank–Nicolson methods are unconditionally stable for the diffusion equation.

For the analysis of viscous flows, the grids are generally refined near the wall to resolve the thin boundary layers. In those regions, the diffusive term becomes the dominant term in the governing equation. For fine grids, computations based on explicit methods become inefficient due to the restrictive time stepping constraint ($\Delta t \propto \Delta^2$). In comparison to the nonlinear advective term, the diffusive term is linear which makes the implementation of implicit method straightforward. For these reasons, it is beneficial to treat the diffusive term implicitly. Even for cases where the viscosity is nonconstant, we can implicitly treat the diffusive terms about the baseline viscosity value and explicitly treat diffusion term related to the variation in viscosity from the baseline value.

2.5.5 Stability of the Discrete Advection-Diffusion Equation

We have discussed that explicit treatment is suitable for the advective term and implicit treatment is desirable for the diffusive term. Let us now consider the advection-diffusion equation, Eq. (2.6), where both of these terms appear. Here, we integrate the advective and diffusive terms using different time stepping schemes. Such approach is called *splitting* and is used to apply the appropriate method for each term in the equation.[4]

Let us consider three splitting methods [4]. First, we consider a comparison of cases where the advective term is integrated in time with the explicit Euler but the diffusive term is integrated in time with the explicit Euler's method (FTCS) or with the implicit Euler's method. Next, we consider second-order accurate time integration with the Crank–Nicolson method for the diffusive term and the second-order Adams–Bashforth method for the advective term. Comparisons are made and the stability characteristics are discussed.

First, consider utilizing the explicit Euler's method for both the advective and diffusive terms (FTCS). We find that the discretization results in

$$\frac{f_j^{n+1} - f_j^n}{\Delta t} = -c\frac{-f_{j-1}^n + f_{j+1}^n}{2\Delta} + a\frac{f_{j-1}^n - 2f_j^n + f_{j+1}^n}{\Delta^2}. \tag{2.151}$$

Representing the solution to be of the form $f_j^n = A^n \exp(ik\Delta j)$, we can rewrite the above equation as

$$\frac{A^{n+1}e^{ik\Delta j} - A^n e^{ik\Delta j}}{\Delta t} = -c\frac{-A^n e^{ik\Delta(j-1)} + A^n e^{ik\Delta(j+1)}}{2\Delta} + a\frac{A^n e^{ik\Delta(j-1)} - 2A^n e^{ik\Delta j} + A^n e^{ik\Delta(j+1)}}{\Delta^2}. \tag{2.152}$$

Solving the equation for the amplitude A, we find

$$A = 1 - iC\sin k\Delta - 2\frac{C}{R}(1 - \cos k\Delta), \tag{2.153}$$

where C is the Courant number that we saw in the discussion of the advection equation (Sect. 2.5.4) and R is the non-dimensional number that compares the viscous and inertial effects over a grid

$$R \equiv \frac{c\Delta}{a}, \tag{2.154}$$

[4] Splitting methods are not limited to time domains. One can treat upwind and downwind fluxes in a different fashion, which results in *flux splitting methods*.

which is called the *cell Reynolds number*. The magnitude of A can then be expressed as

$$|A|^2 = \left[1 - 2\frac{C}{R}(1 - \cos k\Delta)\right]^2 + C^2 \sin^2 k\Delta. \tag{2.155}$$

When $k\Delta \approx 0$ and π, the stability of scheme becomes critical ($|A| \approx 1$). Expanding Eq. (2.155) using Taylor series about these critical points in terms of $k\Delta$, we observe that

$$|A|^2 \to 1 + C^2(k\Delta)^2 - 2\frac{C}{R}(k\Delta)^2 + \mathcal{O}((k\Delta)^4) \quad \text{for} \quad k\Delta \to 0 \tag{2.156}$$

$$|A|^2 \to 1 - 8\frac{C}{R} + 16\left(\frac{C}{R}\right)^2 + \mathcal{O}((k\Delta - \pi)^2) \quad \text{for} \quad k\Delta \to \pi \tag{2.157}$$

which we need them to satisfy $|A|^2 \leq 1$ for stability. Therefore, we find that

$$C \leq \frac{2}{R} \quad \text{and} \quad C \leq \frac{R}{2} \tag{2.158}$$

must be satisfied for the stable numerically integration based on the explicit Euler's method for both advective and diffusive terms. The shaded region in Fig. 2.15a illustrates the combination of the Courant number and cell Reynolds number for which the method is stable. With $C \leq R/2$, the stability region is cut off significantly for $R < 2$ at higher values of Courant number.

Next, let us consider integrating the diffusive term implicitly as we have recommended in Sect. 2.5.4. We choose the implicit Euler's method to demonstrate the change in stability characteristics. The finite-difference formulation becomes

$$\frac{f_j^{n+1} - f_j^n}{\Delta t} = -c\frac{-f_{j-1}^n + f_{j+1}^n}{2\Delta} + a\frac{f_{j-1}^{n+1} - 2f_j^{n+1} + f_{j+1}^{n+1}}{\Delta^2}. \tag{2.159}$$

Utilizing the Fourier representation in the above equation, we can find

$$A = \frac{1 - iC\sin k\Delta}{1 + 2\frac{C}{R}(1 - \cos k\Delta)}. \tag{2.160}$$

For a stable calculation, we require that

$$|A| = \frac{\sqrt{1 + C^2 \sin^2 k\Delta}}{1 + 4\frac{C}{R}\sin^2 \frac{k\Delta}{2}} \leq 1. \tag{2.161}$$

This inequality is critical near $k\Delta \approx 0$ (or 2π) and upon expanding the above relations with respect to $k\Delta$ near 0, we find

$$1 + C^2(k\Delta)^2 \leq 1 + 2\frac{C}{R}(k\Delta)^2 + \mathcal{O}((k\Delta)^4). \tag{2.162}$$

Hence we observe that for numerical stability, we require

$$C \leq \frac{2}{R} \tag{2.163}$$

for solving the advection-diffusion equation in a stable manner using the explicit Euler's method and the implicit Euler's method for the advective and diffusive terms, respectively. The stability region is illustrated in Fig. 2.15b. Comparing this scheme with the previous scheme that used explicit Euler's method for both the advective and diffusive terms, the restriction of $C \leq R/2$ is absent. Thus, we achieve a larger region of stability for $R < 2$.

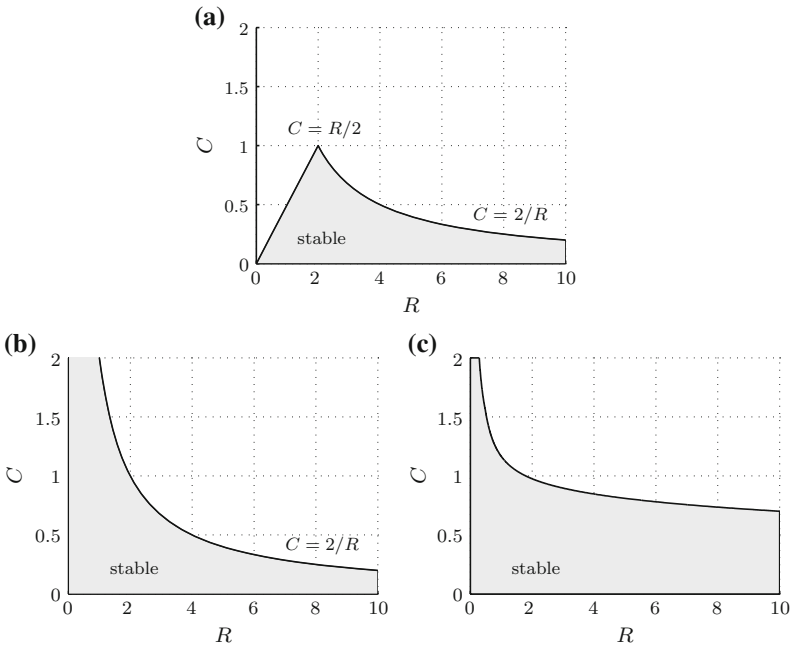

Fig. 2.15 Stability of splitting methods using **a** the explicit Euler's method for advective and diffusive terms, **b** the explicit Euler's method for advective term and the implicit Euler's method for diffusive term, and **c** the second-order Adams–Bashforth method for advective term and the Crank–Nicolson method for diffusive term. *Shaded regions* represent where the methods are stable

In the above analysis, we utilized the first-order accurate explicit and implicit methods to assess the stability properties. We further consider the use of second-order accurate methods that are commonly used to integrate the incompressible Navier–Stokes equations. Below, let us examine the use of the second-order Adams–Bashforth method and the Crank–Nicolson method for advective and diffusive terms, respectively. Based on these schemes, the finite-difference formulation can be written as

$$\frac{f_j^{n+1}-f_j^n}{\Delta t} = -c\left(\frac{3}{2}\frac{-f_{j-1}^n+f_{j+1}^n}{2\Delta} - \frac{1}{2}\frac{-f_{j-1}^{n-1}+f_{j+1}^{n-1}}{2\Delta}\right) + a\left(\frac{1}{2}\frac{f_{j-1}^{n+1}-2f_j^{n+1}+f_{j+1}^{n+1}}{\Delta^2} + \frac{1}{2}\frac{f_{j-1}^{n}-2f_j^{n}+f_{j+1}^{n}}{\Delta^2}\right). \tag{2.164}$$

Again, solving for the amplitude using the Fourier analysis, we find that

$$A = \frac{-\beta \pm \sqrt{\beta^2-4\alpha\gamma}}{2\alpha}, \quad \text{where} \quad \alpha = 1+\frac{C}{R}(1-\cos k\Delta),$$
$$\beta = \frac{C}{R}(1-\cos k\Delta) - 1 + i\frac{3}{2}C\sin k\Delta, \quad \text{and} \quad \gamma = -i\frac{C}{2}\sin k\Delta. \tag{2.165}$$

We can numerically determine the stability boundary for this case as shown in Fig. 2.15c. Notice that the stability boundary is significantly enlarged for $R \gtrsim 2$. While the stability region becomes narrow for small R, we are still able to retain the region where the Courant number C is large, similar to Fig. 2.15b.

With the above three examples of the splitting methods for the advection-diffusion equation, we have found that the implicit treatment of the diffusive term can increase stability for small cell Reynolds numbers and the use of a second-order method can enlarge the region of stability. These assessments become important when we consider the treatment of the inertial and viscous terms in the Navier–Stokes equations.

2.5.6 Time Step Constraints for Advection and Diffusion

Let us discuss an alternative approach for graphically choosing the maximum stable time step for a selected combination of spatial and temporal discretization schemes. There are two constraints that we face from time integration of advective and diffusive terms. First, let us revisit the Courant number for the advective term. The Courant number can be related to $\lambda\Delta t$ along the imaginary axis (see Sect. 2.5.1) and the maximum modified wave number:

$$C = \frac{c\Delta t}{\Delta} < \frac{\max\limits_{\mathrm{Re}(\lambda)=0} \mathrm{Im}(\lambda\Delta t)}{\max(K\Delta)} = \frac{|\lambda_A|\Delta t}{\max(K\Delta)}, \tag{2.166}$$

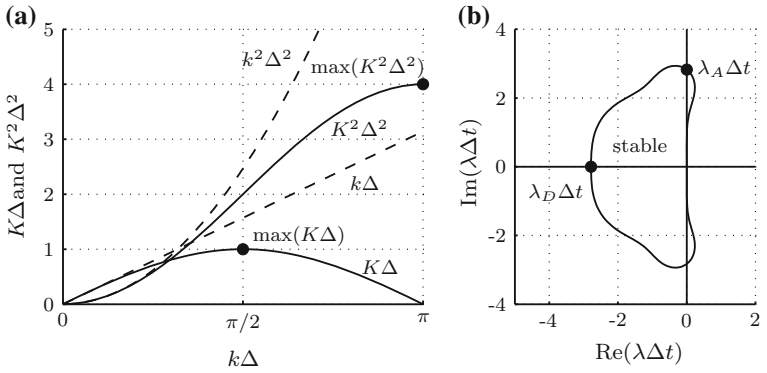

Fig. 2.16 Graphical representation of advective and diffusion time step constraints for the **a** spatial and **b** temporal discretization schemes

where λ_A is determined at the intersect of the stability boundary of the time stepping method and the imaginary axis as depicted in Fig. 2.16.

For the diffusive term, we noticed that the time step is restricted by the diffusive time scale of

$$C/R = \frac{a\Delta t}{\Delta^2} < \frac{-\min\limits_{\mathrm{Im}(\lambda)=0} \mathrm{Re}(\lambda\Delta t)}{\max(K^2\Delta^2)} = \frac{|\lambda_D|\Delta t}{\max(K^2\Delta^2)}, \tag{2.167}$$

by noticing that the decay of the solution is represented by $\lambda\Delta t$ along the real axis. Here, the value of λ_D can be found from the intersect of the stability boundary and the real axis, as shown in Fig. 2.16.

If we choose to discretize the advection-diffusion equation with second-order central-difference schemes of

$$f'_j = \frac{-f_{j-1} + f_{j+1}}{2\Delta} \tag{2.168}$$

$$f''_j = \frac{f_{j-1} - 2f_j + f_{j+1}}{\Delta^2} \tag{2.169}$$

with the fourth-order Runge–Kutta method for time stepping, we can find graphically from Fig. 2.16 that

$$|\lambda_A|\Delta t = 2.83, \quad |\lambda_D|\Delta t = 2.79, \quad \max(K\Delta) = 1, \quad \max(K^2\Delta^2) = 4, \tag{2.170}$$

providing us with

$$C = \frac{c\Delta t}{\Delta} < 2.83 \quad \text{and} \quad C/R = \frac{a\Delta t}{\Delta^2} < 0.698 \tag{2.171}$$

as restrictions on the choice of time step to perform the simulation in a stable manner.

2.5.7 Amplitude and Phase Errors

Up until now, we have focused on the amplitude error during time integration. Although unstable schemes are not useful, it does not automatically mean that stable schemes are superior. If the solution scheme is stable but over-damped, the solution becomes inaccurate. We should also analyze whether neutrally stable schemes can compute the solutions correctly.

Let us consider the next example, known as the *leapfrog method*,[5] based on the second-order accurate central-difference discretization in time and space

$$\frac{f_j^{n+1} - f_j^{n-1}}{2\Delta t} = -c\frac{-f_{j-1}^n + f_{j+1}^n}{2\Delta}. \tag{2.172}$$

Utilizing the von Neumann analysis and realizing that $A = A^{n+1}/A^n = A^n/A^{n-1}$, we obtain a second-order equation

$$A^2 + 2iCA\sin k\Delta - 1 = 0. \tag{2.173}$$

The solution to this equation is

$$A = -iC\sin k\Delta \pm \sqrt{1 - C^2\sin^2 k\Delta}. \tag{2.174}$$

Now let us write $A = |A|e^{i\varphi}$, where φ represents the phase difference. From Eq. (2.174), we observe that the leapfrog method for $C \leq 1$ leads to

$$|A| = 1, \quad \varphi = -\tan^{-1}\frac{C\sin k\Delta}{(1 - C^2\sin^2 k\Delta)^{1/2}}. \tag{2.175}$$

Since the exact solution to the advection equation for $f_j^n = A^n\exp(ik\Delta j)$ is $f_j^{n+1} = A^n\exp[ik(\Delta j - c\Delta t)]$, we know that $|A|_{\text{exact}} = 1$ and $\varphi_{\text{exact}} = -Ck\Delta$. The ratio between Eq. (2.175) and the exact solutions, $|A|/|A|_{\text{exact}}$ and $\varphi/\varphi_{\text{exact}}$ correspond to the relative amplitude error and the relative phase error, respectively [1, 4].

For the leapfrog scheme with $C \leq 1$, the method is neutrally stable with no amplitude error for all wave numbers k. However, for $C < 1$, $\varphi/\varphi_{\text{exact}}$ becomes small in the neighborhood of $k\Delta = \pi$ (near the cutoff frequency $k_c = \pi/\Delta$), resulting in delay of high-frequency components being advected. Such numerical schemes having different wave speeds for different wavelengths are called *dispersive* and the associated error is referred to as *dispersive error*. The central-difference leapfrog method suffers from oscillations due to the phase error and leads to eventual blowup of the solution, even though the method is analytically speaking neutrally stable

[5] The stability of the leapfrog method is depicted in Fig. 2.14.

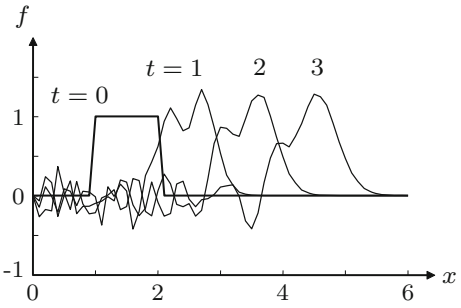

Fig. 2.17 Numerical solution to the advection equation using the leapfrog method (Courant number of 0.5)

(i.e., $|A| = 1$). As shown in Fig. 2.17, we can observe oscillations for short wavelengths do not get transported as fast as they should in the x-direction. This tells us that the phase delay is appearing in high-frequency components.

Even for numerical schemes with growth rates of less than 1, dispersive methods introduce numerical oscillations into the solution. In general, oscillations due to dispersive errors appear behind the wave packet if the scheme is phase-delayed and in front of the wave packet if the scheme is phase-advanced.

2.6 Higher-Order Finite Difference

In this section, we present an alternative approach for deriving higher-order finite-difference schemes. The approach below considers an implicit formulation of derivative approximation to achieve spectral-like accuracy using a compact stencil. Such schemes have been proven useful to study turbulence and aeroacoustics. High-order accuracy is particularly important since pressure disturbance associated with acoustics waves are orders of magnitude smaller than the hydrodynamic pressure fluctuations. Therefore, solvers need to be designed with high-order accuracy to correctly predict the overall flow physics. The method in this section is not discussed further in other chapters of this book, as we focus on conservation properties of the discrete equations. Here, we briefly present the main idea behind deriving higher-order accurate methods as they continue to be a subject of active research.

We have thus far derived finite-difference formulas in an explicit manner such that the discrete derivative can be computed by summing discrete functional values with appropriate weights. It is also possible to implicitly solve for the derivate values by constructing a sparse matrix equation.[6] Following the work of Lele [3], let us present an example of implicit finite-difference method on a five-point stencil

[6]Implicitly solving for the derivative may increase the computational cost. However, this translates to the differencing stencil being all coupled, which shares similarity with spectral methods.

$$\beta f'_{j-2} + \alpha f'_{j-1} + f'_j + \alpha f'_{j+1} + \beta f'_{j+2} = c\frac{f_{j+3} - f_{j-3}}{6\Delta} + b\frac{f_{j+2} - f_{j-2}}{4\Delta} + a\frac{f_{j+1} - f_{j-1}}{2\Delta}. \tag{2.176}$$

For the above equation, we must meet the following condition to cancel truncation error and achieve the noted order of accuracy:

$$a + b + c = 1 + 2\alpha + 2\beta \qquad \text{(second order)} \tag{2.177}$$

$$a + 2^2 b + 3^2 c = 2\frac{3!}{2!}(\alpha + 2^2\beta) \qquad \text{(fourth order)} \tag{2.178}$$

$$a + 2^4 b + 3^4 c = 2\frac{5!}{4!}(\alpha + 2^4\beta) \qquad \text{(sixth order)} \tag{2.179}$$

$$a + 2^6 b + 3^6 c = 2\frac{7!}{6!}(\alpha + 2^6\beta) \qquad \text{(eighth order)} \tag{2.180}$$

$$a + 2^8 b + 3^8 c = 2\frac{9!}{8!}(\alpha + 2^8\beta) \qquad \text{(tenth order)} \tag{2.181}$$

Finite-difference schemes based on this formulation is called the *compact finite-difference schemes* and is a generalization of the *Padé approximation*. The corresponding modified wave number for Eq. (2.176) becomes

$$K\Delta = \frac{a\sin(k\Delta) + \frac{b}{2}\sin(2k\Delta) + \frac{c}{3}\sin(3k\Delta)}{1 + 2\alpha\cos(k\Delta) + 2\beta\cos(2k\Delta)}. \tag{2.182}$$

For illustration of the accuracy of the compact scheme, let us consider a tridiagonal system on the left-hand side of Eq. (2.176) with $\beta = 0$ and $c = 0$. We can then find a family of fourth-order accurate finite-difference schemes with

$$a = \frac{2}{3}(\alpha + 2), \quad b = \frac{1}{3}(4\alpha - 1), \quad c = 0, \quad \beta = 0 \tag{2.183}$$

Note that for $\alpha = 0$, we recover the fourth-order central-difference scheme, Eq. (2.21), and for $\alpha = 1/4$, we find the classical Padé approximation. If we choose $\alpha = 1/3$, the truncation errors cancel and provide us with the sixth-order compact finite-difference scheme, where

$$a = \frac{14}{9}, \quad b = \frac{1}{9}, \quad c = 0, \quad \alpha = \frac{1}{3}, \quad \beta = 0. \tag{2.184}$$

This scheme requires us to solve a tridiagonal system but provides us with high order of accuracy. For illustration of the scheme, we provide a comparison of the modified wave number in Fig. 2.18 for higher order schemes and the classical Taylor series-based formulations. It can be observed that the compact finite-difference schemes can closely position the modified wave number to the exact wave number even for high wave number.

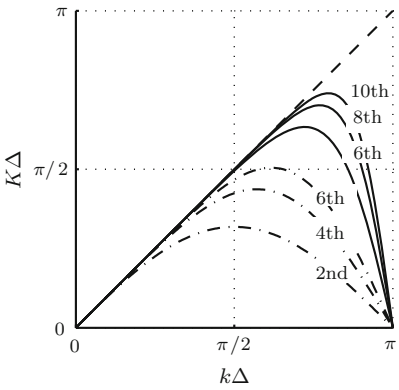

Fig. 2.18 Comparison of modified wave numbers for the compact finite-difference methods with sixth, eighth, and tenth-order accuracy (*solid*) and Taylor series-based central-difference schemes of second, fourth, and sixth-order accuracy (*dash-dot*)

In general, the above approach can be taken to derive higher derivative approximations and higher-order accurate finite-difference methods. For non-periodic boundaries, special care must be taken in the construction of the compact finite-difference methods. Additional details can be found in Lele [3].

2.7 Consistency of Finite-Difference Methods

Finite-difference methods are called *consistent* when the discretized differential equation converges to the original continuous differential equation in the limit of $\Delta \to 0$ and $\Delta t \to 0$. In other words, the truncation error from the finite-difference method should converge to zero for the method to be consistent.

To illustrate how an *inconsistent* finite-difference method can produce incorrect results, we consider a method called the *DuFort–Frankel method* for the diffusion equation

$$\frac{f_j^{n+1} - f_j^{n-1}}{2\Delta t} = a \frac{f_{j-1}^n - 2\frac{f_j^{n+1} + f_j^{n-1}}{2} + f_{j+1}^n}{\Delta^2}. \tag{2.185}$$

The spatial discretization is similar to the second-order central-difference scheme but replaces the middle value with time average value. For, the time derivative, the past and next time step values are employed. While this scheme is unconditionally stable, the Taylor series analysis reveals that in the limit of Δt and Δ approaching 0 the DuFort–Frankel method leads to

$$\frac{\partial f}{\partial t} = a \frac{\partial^2 f}{\partial x^2} - a \frac{\Delta t^2}{\Delta^2} \frac{\partial^2 f}{\partial t^2}. \tag{2.186}$$

This implies that for the method to produce the correct solution, the time step Δt must approach 0 at a faster rate than Δ approaching 0. Otherwise, we would have a hyperbolic PDE instead of a parabolic PDE, which completely changes the behavior of the solution. Although the Dufort–Frankel method is unconditionally stable, the inconsistency seen above prohibits us from achieving convergence. To achieve convergence, this method requires us to satisfy $\Delta t/\Delta \rightarrow 0$ in addition to Δt and $\Delta \rightarrow 0$, which is restrictive.

We briefly note that both stability and consistency of finite-difference schemes are required for convergence of the numerical solution to exact solution of a linear PDE. This is known as the *Lax equivalence theorem*. The theorem does not hold for nonlinear equations, such as the Navier–Stokes equations. Nonetheless, it does provide us with the insight that satisfying stability and consistency is important for developing convergent finite-difference methods to solve fluid flow problems.

2.8 Remarks

Every finite-difference method has some level numerical error from the discretization process. While one may wonder whether we can obtain a reliable solution from flow simulations, in reality there are stable and accurate computational schemes that allow us to confidently simulate flow physics. It is important that we understand how the results from different spatial and temporal discretization schemes behave. Such insights enable us to develop numerical solvers that can arrive at the incompressible flow solution in a stable and accurate manner, as we will see in the next chapter.

Before we end this chapter, we should discuss how we can validate simulation results. Numerical validation can be performed relatively easily for those simulations where (1) experiments are possible but only limited data can be obtained and (2) the time and cost required by experiments are so large that simulations are preferred to experiments. For the first case, numerical results can provide deeper insight into the flow physics. In these cases, close comparison of numerical and experimental data can provide confidence in using the simulation.

If we do not limit the use of flow simulations to reproduce experiments or theoretical results and use simulations to gain new knowledge of flow physics, it is not enough to simply check the correctness of our solution against experiments or theory. In these cases, the validity of a simulation must be confirmed with extra care because numerical simulation can be the only means to obtain information about the flow. The simplest yet realistic way to check the solution is to examine how the numerical solution may change for different choices of spatial discretization, as mentioned in Sect. 2.3.6. Specifically, we can change the number of grid points in the domain, how the grids are distributed (nonuniformly), or the size of the computational domain. In order to analyze the results, one would need to use insights from fluid mechanics and mathematics to examine how the solution could be affected. What we discussed in this chapter focused only on whether the governing equation has been discretized correctly. Whether the equation that we solve reproduces the flow physics correctly

requires insights from fluid mechanics and is a different problem, which we will discuss in the following chapters.

2.9 Exercises

2.1 Burgers' equation captures nonlinear advection and diffusion

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = a\frac{\partial^2 u}{\partial x^2},$$

where a is a positive constant. Show that using a transform (Cole–Hopf transform) of

$$u = -2a\frac{1}{\phi}\frac{\partial \phi}{\partial x},$$

the nonlinear Burgers' equation can be transformed to a linear PDE.

2.2 For real x and integers m, n with $m > n$, prove the following identities:

1. $\mathcal{O}(x^m) \pm \mathcal{O}(x^n) = \mathcal{O}(x^m)$
2. $\mathcal{O}(x^m) \cdot \mathcal{O}(x^n) = \mathcal{O}(x^{m+n})$
3. $\mathcal{O}(x^m)/\mathcal{O}(x^n) = \mathcal{O}(x^{m-n})$

2.3 Given the values of a function $f_j = f(x_j)$ at equally space points $x_j = j\Delta x$, consider the finite-difference scheme for the first derivative below

$$f'(x_j) = \frac{1}{\Delta x}(af_{j-2} + bf_{j-1} + cf_j).$$

1. Perform Taylor series expansions of f_{j-2} and f_{j-1} about x_j and find the relationships that the coefficients a, b, and c need to satisfy for the above scheme. Show the first four equations for the coefficients starting from relationship for the lowest order terms.
2. Determine the coefficients a, b, and c such that the above scheme achieves the highest possible order of accuracy.
3. Find the coefficients a, b, and c such that the third derivative f''' does not influence the error.

2.4 Given the values of a function $f_j^n = f(x_j, t_n)$ at equally space points $x_j = j\Delta x$ and time levels $t_n = n\Delta t$, consider the linear advection equation with constant advection velocity u

$$\frac{\partial f}{\partial t} + u\frac{\partial f}{\partial x} = 0$$

and derive an explicit time stepping method of the following form:

$$f_j^{n+1} = af_{j-2}^n + bf_{j-1}^n + cf_j^n.$$

1. Perform Taylor series expansions of f_j^{n+1}, f_{j-2}^n, and f_{j-1}^n about (x_j, t_n) and find the relationships that the coefficients a, b, and c need to satisfy for the above time stepping scheme. Show the first four equations for the coefficients starting from relationship for the lowest order terms.
2. Determine the coefficients a, b, and c such that the above scheme achieves the highest possible order of accuracy.
3. Find the coefficients a, b, and c such that the second (spatial) derivative f'' does not influence the error.

2.5 Consider a function $f_j = f(x_j)$ given at equally spaced points $x_j = j\Delta x$.

1. Determine the coefficients (a_0, a_1, a_2, and a_3) and the order of accuracy for

$$\frac{\mathrm{d}^2 f}{\mathrm{d}x^2}(x_j) = a_0 f_j + a_1 f_{j+1} + a_2 f_{j+2} + a_3 f_{j+3}.$$

2. For a periodic function $f(x) = \exp(ikx)$ with wave number k, the second derivative is given by $\frac{\mathrm{d}^2 f}{\mathrm{d}x^2} = -k^2 \exp(ikx)$. Find the expression K_2 that the above finite-difference scheme yields as an approximation (i.e., $-K_2 \exp(ikx)$). Moreover, show that K_2 approaches k^2 as $\Delta x \to 0$.

2.6 Let us consider a function F evaluated at uniform grid points $x_j = j\Delta$ and write $F_j = F(x_j)$. We denote the derivatives of F as $F^{(n)} = \mathrm{d}^n F/\mathrm{d}x^n$.

1. Express $F_{j\pm1}$ using the Taylor series expansion up to the fourth derivative about $x = x_j$.
2. Denoting $F^{(1)} = U$ and $F^{(2)} = V$, find the finite-difference approximations of $F^{(3)}$ and $F^{(4)}$ as the second derivatives of $F^{(1)}$ and $F^{(2)}$, respectively, using U_j and V_j.
3. Find the fourth-order finite-difference formulas for $F^{(1)}$ and $F^{(2)}$ using U_j and V_j.

2.7 Consider the ordinary differential equation given by Eq. (2.115) $\frac{\mathrm{d}f}{\mathrm{d}t} = \lambda f$ with $f(t_0) = 1$.

1. Show that the Crack–Nicolson method is stable for $Re(\lambda\Delta t) < 0$.
2. Find the stability criteria of first, second, third, and fourth-order Runge–Kutta methods in terms of $\lambda\Delta t$. Plot the stability contours.

2.8 Consider discretizing the two-dimensional linear advection equation

$$\frac{\partial f}{\partial t} + u\frac{\partial f}{\partial x} + v\frac{\partial f}{\partial y} = 0,$$

for constant $u > 0$ and $v > 0$ using the first-order upwind finite differencing in space and forward Euler integration in time to yield

$$\frac{f_{i,j}^{n+1} - f_{i,j}^{n}}{\Delta t} = -u\frac{-f_{i-1,j}^{n} + f_{i,j}^{n}}{\Delta x} - v\frac{-f_{i,j-1}^{n} + f_{i,j}^{n}}{\Delta y}.$$

Based on the von Neumann analysis, find the condition needed to achieve numerical stability using Courant numbers, $C_x \equiv u\Delta t/\Delta x$ and $C_y \equiv v\Delta t/\Delta y$. Assume uniform grid for spatial discretization.

2.9 Given the linear advection-diffusion equation

$$\frac{\partial f}{\partial t} + u\frac{\partial f}{\partial x} = a\frac{\partial^2 f}{\partial x^2},$$

let us consider the forward-in-time central-in-space (FTCS) method with uniform spacing to yield

$$f_j^{n+1} = f_j^n - \frac{C}{2}(-f_{j-1}^n + f_{j+1}^n) + D(f_{j-1}^n - 2f_j^n + f_{j+1}^n),$$

where $C \equiv u\Delta t/\Delta$ and $D \equiv a\Delta t/\Delta^2$. Assuming solution of the form

$$f_j^n = A^n \exp(ik\Delta j),$$

answer the following questions:

1. Find the expression for the amplitude coefficient A and show that it represents an ellipse on the complex plane. Determine its axes.
2. Using the result from part 1, derive the condition for the above FTCS scheme to be stable.
3. What can the cell Reynolds number, $R \equiv C/D = c\Delta/a$, reveal in terms of the analysis here.

2.10 For the advection-diffusion equation, consider the use of

1. The explicit Euler's method for the advection term and the implicit Euler's method for the diffusion term (see: Eq. (2.159)).
2. The second-order Adam–Bashforth method for the advection term and the Crank–Nicolson method for the diffusion term (see: Eq. (2.164)).

Find the expressions for the amplitude A for the above two cases. Hint: final answers are given in Sect. 2.5.5.

References

1. Fujii, K.: Numerical Methods for Computational Fluid Dynamics. Univ. Tokyo Press, Tokyo (1994)
2. Lambert, J.D.: Computational Methods in Ordinary Differential Equations. Wiley, London (1973)
3. Lele, S.K.: Compact finite difference schemes with spectral-like resolution. J. Comput. Phys. **103**, 16–42 (1992)
4. Lomax, H., Pulliam, T.H., Zingg, D.W.: Fundamentals of Computational Fluid Dynamics. Springer, New York (2001)
5. Wade, W.R.: Introduction to Analysis, 4th edn. Pearson (2009)
6. Williamson, J.H.: Low-storage Runge-Kutta schemes. J. Comput. Phys. **35**(1), 48–56 (1980)

http://www.springer.com/978-3-319-45302-6

Computational Fluid Dynamics
Incompressible Turbulent Flows
Kajishima, T.; Taira, K.
2017, XV, 358 p. 107 illus., Hardcover
ISBN: 978-3-319-45302-6