Mesoscale Dynamics

By Yuh-Lang Lin

[Lin, Y.-L., 2007: *Mesoscale Dynamics*. Cambridge University Press, 630pp.]

Chapter 12 Basic numerical methods

12.1 Introduction

In Chapter 2, we derived a set of nonlinear partial differential equations governing mesoscale atmospheric motions. One way to study the dynamics associated with these equations is to make the small-amplitude approximation and solve the linearized equations analytically, as demonstrated in earlier chapters. However, this approach limits us to study only mesoscale systems with small-amplitude perturbations. In addition, the number of available analytical methods at hand to solve these complicated equations is limited. As mentioned in earlier chapters, an alternative solution is to use numerical methods where the equations are discretized and solved numerically in space and time. The advantage of applying the numerical methods is that they are able to solve completely the nonlinear set of equations. Numerical methods also provide a powerful framework for sensitivity tests or experiments with forcing or physical processes. In these experiments, physical parameterizations or external forcing can be easily altered or completely deactivated. Examples of the parameterizations include those for planetary boundary layer processes, moist processes, and radiative processes, while external forcing can come from orography. In this regard, numerical simulations are more flexible than physical experiments, such as the experiments conducted in a water tank, gas chamber or wind tunnel, and field experiments conducted in the real atmosphere.

When numerical methods are adopted to solve mathematically intractable governing equations, one needs to address the following important questions: (1) Does the solution

of the approximate equations converge to that of the original differential equations when the time and grid intervals approach zero? (2) Is the numerical solution well-behaved in time, or more precisely, is the numerical scheme stable? (3) If the numerical scheme is stable, how well do the amplitudes and phases of the approximated waves or disturbances represent those of the exact solution? We will try to answer these questions in this chapter.

The major numerical methods that are used to solve partial differential equations can be categorized as (1) *finite difference methods*, (2) *Galerkin methods*, and (3) *Lagrangian methods*. Combinations of these methods for solving a set of time-dependent equations have also been developed. In the finite difference methods, dependent variables are defined at specific grid points in space and time, and the derivatives in the equations are approximated by Taylor series expansion or other approaches. For mesoscale numerical weather prediction models, the governing equations are solved in a finite region of the atmosphere. Thus, mesoscale models are often referred to as *limited-area* or *regional models*. In order to integrate the governing equations numerically, boundary conditions of the variables at the boundary of the integration domain are required. In addition, initial conditions are also required for integrating time-dependent partial differential equations so as to arrive at a future solution, which we call *prediction*. Finite difference methods are the most popular numerical methods adopted for mesoscale numerical weather prediction models.

In *Galerkin methods*, dependent variables are represented by a sum of functions that have prescribed spatial structures. The coefficient associated with each function is normally a function of time for a time dependent problem, which transforms a partial

differential equation into a set of ordinary differential equations (in time) for the coefficients. These equations are usually solved using finite difference approximations in time. The Galerkin methods can be divided into two major categories: the *spectral method* and the *finite element method*. In the spectral method, dependent variables are represented by orthogonal, global basis functions, such as a sinusoidal function. The spectral method is less popular with mesoscale models due to the difficulties posed by nonperiodic lateral boundary conditions with limited area models. The spectral method is, however, much more convenient with global models due to the periodic nature of their zonal boundary conditions. Techniques for treating nonperiodic boundary conditions have been developed in the last two decades; thus the spectral method has also been used in mesoscale models. Finite element method is similar to spectral method except that it uses local instead of global (in terms of the integration domain) basis functions. The local basis functions include chapeau and tent functions. There is a growing interest in adopting the finite element methods for mesoscale models due to their accuracy and flexibility in treating the irregular geometry of internal or external boundary. One of the disadvantages of finite element methods is that they usually require significantly more computing time to invert a normally large matrix every time step. Although finite element methods are more accurate compared to finite difference methods with the same order of accuracy, finite difference methods can achieve similar solution accuracy by using a higher-order scheme with less computing time. In addition, the advantage of finite element methods in treating irregular lower boundaries is significantly diminished when finite difference numerical models employ the so-called *terrain-following*

coordinates, in which the irregular lower boundary becomes 'flat' or regular within the transformed computational domain.

In the *Lagrangian methods*, the equations governing the fluid motion are solved by following a fixed set of particles throughout the period of integration. The advantage of a Lagrangian method is that it treats the total derivative at once, instead of treating the local rate of change and advection terms individually. However, in general, a set of fluid particles, which are initially distributed regularly, will soon become greatly deformed, and are thus rendered unsuitable for numerical integration. In order to avoid this problem, the *semi-Lagrangian method* is employed. Thus, the fluid variables at the predicted time step can be defined at the regular grid points and those at the previous time steps (which are often not located at the regular grid points) are interpolated from the known values at the regular grid points from the previous time step. The semi-Lagrangian method has become popular in recent years, especially with large-scale models, since a relatively large time interval for integration can be used due to its unconditional stability characteristics.

Other methods, such as the *upstream interpolation* (e.g., Pielke 2002) and *finite-volume* (e.g., Durran 1998) methods, have also been used in mesoscale numerical models. In an *interpolation method*, dependent variables at grid points are used to derive interpolation formulae that are then used to calculate spatial derivatives. Unlike finite-difference methods, *finite-volume methods* generate approximations to the grid-interval or grid-cell average. In a finite-volume method, the grid-point value f_i represents the average of a function, $f(x)$, over the interval (or grid cell) $[(i-\frac{1}{2})\Delta x, (i+\frac{1}{2})\Delta x]$, taking

a one-dimensional problem as an example. Finite-volume methods are very useful for approximating solutions that contain discontinuities (e.g., Colella and Woodward 1984).

12.2 Finite difference approximations of derivatives

Before discussing various finite difference methods, let us consider a simple example of a finite difference approximation to help us understand the approximation of derivatives of a given function. For example, one may use the Taylor series to approximate $f(x)$ at $x+\Delta x$,

$$f(x+\Delta x) = f(x) + f'(x)\Delta x + f''(x)\frac{\Delta x^2}{2!} + f'''(x)\frac{\Delta x^3}{3!} + ..., \qquad (12.2.1)$$

where Δx is the *grid interval*. For convenience, Δx is assumed to be greater than 0 in the following discussions. The derivative of $f(x)$ can be calculated from,

$$f'(x) = \frac{f(x+\Delta x) - f(x)}{\Delta x} + R(x,\Delta x), \qquad (12.2.2)$$

where

$$R(x,\Delta x) \equiv \Delta x\left\{-\frac{f''(x)}{2!} - \frac{f'''(x)\Delta x}{3!} - ....\right\}$$

is called the *remainder*, which has a magnitude of $O(\Delta x)$. If the remainder term is much smaller than the first term on the right side of (12.2.2), then the above equation can be approximated by

$$f_D'(x) = \frac{f(x+\Delta x) - f(x)}{\Delta x}. \qquad (12.2.3)$$

A differential equation becomes a *finite difference equation* when the derivatives are approximated by their finite difference forms, known as a *forward difference scheme* for

(12.2.3), which has a first order of accuracy. The actual derivative, $f'(x)$, is approximated by the slope $f_D'(x)$ (Fig. 12.1). The distance between $f(x)+f'(x)\Delta x$ and $f(x+\Delta x)$ is $-R\Delta x$. When Δx is reduced, the approximated derivative $f'(x)$, i.e. the approximated slope of $f(x)$, is closer to the real derivative. As seen from (12.2.2) and Fig. 12.1, there are two ways to reduce the truncation errors: (a) reducing the space interval (Δx) and (b) using a higher order approximation.

Similarly, the Taylor series expansion can also be expanded in a backward manner,

$$f(x-\Delta x)=f(x)-f'(x)\Delta x+f''(x)\frac{\Delta x^2}{2!}-f'''(x)\frac{\Delta x^3}{3!}+\ldots., \tag{12.2.4}$$

which can be rearranged in the following form:

$$f'(x)=\frac{f(x)-f(x-\Delta x)}{\Delta x}+\Delta x\left\{\frac{f''(x)}{2!}-\frac{f'''(x)\Delta x}{3!}+\ldots.\right\}. \tag{12.2.5}$$

Again, if the remainder term of the above equation is much smaller than the first term on the right side, then (12.2.5) can be approximated using the *backward difference scheme*,

$$f_D'(x)=\frac{f(x)-f(x-\Delta x)}{\Delta x}. \tag{12.2.6}$$

The meaning of (12.2.6) can be easily understood by replacing $x+\Delta x$ and x with x and $x-\Delta x$, respectively, in Fig. 12.1, where the approximated slope, $f'(x)$, is replaced by the straight line connecting $f(x-\Delta x)$ and $f(x)$. Like the forward difference scheme, the backward scheme has the first order of accuracy.

An alternative way to approximate the derivative is to subtract (12.2.4) from (12.2.1):

$$f(x+\Delta x)-f(x-\Delta x)=2f'(x)\Delta x+2f'''(x)\frac{\Delta x^3}{3!}+\ldots. . \tag{12.2.7}$$

The derivative, $f'(x)$, can then be calculated from

$$f'(x) = \frac{f(x+\Delta x) - f(x-\Delta x)}{2\Delta x} + R, \tag{12.2.8}$$

where the remainder term is defined as the following,

$$R = \Delta x^2 \left\{ -\frac{f'''(x)}{3!} - \frac{f^{(5)}(x)}{5!}\Delta x - .... \right\}. \tag{12.2.9}$$

Neglecting the remainder term leads to the *centered difference scheme*,

$$f_D'(x) = \frac{f(x+\Delta x) - f(x-\Delta x)}{2\Delta x}. \tag{12.2.10}$$

Based on (12.2.9), the centered difference scheme has an accuracy on the order of Δx^2, which is the second order of accuracy. The mathematical meaning of the centered difference scheme is depicted in Fig. 12.2. When compared with Fig. 12.1, it is apparent that this scheme is more accurate than the forward finite difference scheme.

Sometimes an *approximation of the second-order derivative* is needed in solving the governing equations, such as that appearing in the diffusion terms. One way to approximate the second-order derivative, $f''(x)$, is to add (12.2.4) to (12.2.1) and neglect the remainder term,

$$f''(x) = \frac{f(x+\Delta x) - 2f(x) + f(x-\Delta x)}{\Delta x^2}. \tag{12.2.11}$$

When an approximation is made, it is important to know the *accuracy* of the scheme obtained as a result of the approximation. To determine the accuracy of finite difference methods, we may consider the centered difference approximation to the first derivative of the sine function,

$$f(x) = A\sin\frac{2\pi x}{L}. \tag{12.2.12}$$

The first-order derivative can be easily obtained through analytical methods,

$$f'(x) = \frac{2\pi A}{L}\cos\frac{2\pi x}{L}. \tag{12.2.13}$$

Now, we can apply the centered difference scheme, (12.2.10), to $f'(x)$,

$$f_D'(x) = \frac{A\sin\left(2\pi(x+\Delta x)/L\right) - A\sin\left(2\pi(x-\Delta x)/L\right)}{2\Delta x}, \tag{12.2.14}$$

which can be rearranged as

$$f_D'(x) = \frac{A\ \cos\left(2\pi x/L\right)\sin\left(2\pi\Delta x/L\right)}{\Delta x}. \tag{12.2.15}$$

Dividing the above approximation by $f'(x)$ yields

$$\frac{f_D'(x)}{f'(x)} = \frac{\sin(2\pi\Delta x/L)}{2\pi\Delta x/L}. \tag{12.2.16}$$

The relationship between $f_D'(x)$ and $f'(x)$ is also sketched in Fig. 12.2. From the above expression, we obtain

$$\frac{f_D'(x)}{f'(x)} \to 1 \quad \text{as} \quad \frac{2\pi\Delta x}{L} \to 0 \tag{12.2.17}$$

because $\sin\theta \to \theta$ when $\theta \to 0$. In other words, the truncation error of the center difference scheme approaches 0 when $\Delta x << L$. In order to obtain a good approximation, the grid interval chosen should therefore be much smaller than the wavelength.

Now let us consider a special wave with $L = 2\Delta x$, which implies that one wavelength is exactly equal to two grid intervals. Such a wave is often called $2\Delta x$ *wave*. Substituting $L = 2\Delta x$ on the right side of (12.2.16) leads to

$$\frac{f_D'(x)}{f'(x)} = \frac{\sin\pi}{\pi} = 0. \qquad (12.2.18)$$

The above equation implies that the centered difference scheme fails to represent accurately a $2\Delta x$ wave. For a fixed grid interval, shorter waves are much more poorly represented by difference schemes than longer waves are. In fact, long waves can be very accurately represented.

Examples of finite difference approximations of derivatives with various orders of accuracy can be found in Appendix 12.1.

12.3 Finite difference approximations of the advection equation

One of the simplest finite difference time integration equations is the one-dimensional advection equation with a constant advection velocity (*c*), which composes of only one dependent variable, one time derivative and one spatial derivative,

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0, \qquad (12.3.1)$$

where *u* represents a quantity, such as horizontal velocity or temperature, being advected in the *x* direction at a speed *c*. If the advection speed *c* is replaced by *u*, and *u* is the advective velocity in the above equation, it leads to one of the simplest nonlinear equations, the inviscid *Burger equation*. An analytical solution of (12.3.1) is

$$u(x,t) = f(x-ct), \qquad (12.3.2)$$

where f is an arbitrary function, whose functional form is determined by u at $t=0$, or the *initial condition*, $f^0(x)$. For example, if

$$f^0(x) = \frac{u_o a^2}{x^2 + a^2}, \tag{12.3.3}$$

then

$$u(x,t) = f^0(x - ct) = \frac{u_o a^2}{(x - ct)^2 + a^2}. \tag{12.3.4}$$

If the advection velocity is positive (negative), then the wave propagates to the right (left). Note that (12.3.3), as mentioned in earlier chapters, is called the *bell-shaped function*, which has an amplitude u_o and a half-width a. The physical meaning of the above solution is that $u(x,t)$ constantly maintains its initial shape along the *phase line*, i.e. *x-ct* = constant. This type of wave is also called nondispersive. Such a wave or disturbance propagation is illustrated in Fig. 12.3.

In the following, we will discuss the characteristics of several popular numerical approximation of the advection equation that have been adopted in mesoscale numerical models. Based on the number of time levels involved, these methods can be categorized as *two-time-level schemes* and *three-time-level schemes*.

12.3.1 Two-time-level schemes

The finite difference schemes used to approximate $f'(x)$, as discussed in Section 12.2, can also be applied to the time derivative. One can choose to adopt the forward, backward, or centered difference approximations. The methods of forward or backward finite difference in time belong to the so-called *two-time-level schemes* because only two time levels are involved in each step of time integration. On the other hand, using the second-order centered difference in time (leapfrog scheme) would be to use the so-called

three-time-level scheme since there are three times involved at each time step of integration.

(a) Forward-in-time and centered-in-space scheme

We use this scheme to demonstrate that not every finite difference method can be used to obtain a usable numerical solution and also to prove that *not every numerical method is numerically stable*. A natural choice in approximating the advection equation, (12.3.1), isby a combination of the forward difference of the time derivative and center difference of the spatial derivative on a time-spatial grid system shown in Fig. 12.4:

$$\frac{u_i^{\tau+1} - u_i^{\tau}}{\Delta t} + c\left(\frac{u_{i+1}^{\tau} - u_{i-1}^{\tau}}{2\Delta x}\right) = 0, \tag{12.3.5}$$

where the superscript τ and subscript i denote the time step and the grid point in space, respectively. Of particular interest is predicting u at time step $\tau+1$ and grid point i , i.e. $u_i^{\tau+1}$, which can be obtained from the above equation,

$$u_i^{\tau+1} = u_i^{\tau} - \left(\frac{c\Delta t}{2\Delta x}\right)\left(u_{i+1}^{\tau} - u_{i-1}^{\tau}\right). \tag{12.3.6}$$

Equation (12.3.6) is called the difference equation of the advection equation obtained using the *forward-in-time and centered-in-space scheme*. The scheme's algorithm is sketched in Fig. 12.4. Initial conditions are needed in order to obtain values at time step 2, while boundary conditions are needed at both the left and right boundary points, $i=1$ and $i=n$. The interior points u_i, $i=2,3,...,n-1$ at time step $\tau+1$ are predicted by

(12.3.6), using the values of $u_{i-1,}$ u_i, u_{i+1} at time step τ. The values at the boundary points, i.e. $u_1^{\tau+1}$ and $u_n^{\tau+1}$, are determined by the boundary conditions.

Although many schemes to approximate a differential equation exist, there is no guarantee that every numerical solution is well-behaved or stable. A finite difference scheme is stable only if the solution at a fixed time $t = \tau\Delta t$ remains bounded as $\Delta t \to 0$. When this occurs, the scheme is *numerically stable*. Otherwise, it is *numerically unstable*. To examine the *numerical stability* of the forward-in-time and centered-in-space scheme, we consider the following sinusoidal wave in both time *t* and space *x*:

$$u(x,t) = \hat{u}(k,\omega)\ e^{i(kx-\omega t)}, \tag{12.3.7}$$

where $\hat{u}$ is the wave amplitude, k the wave number and ω the wave frequency. All three variables, $\hat{u}$, k, ω, are complex numbers. Both x and t can be represented by the grid and time intervals, respectively, $x = n\Delta x$, and $t = \tau\,\Delta t$, where n and τ represent the grid and time intervals, respectively, from the origin $(x,t) = (0,0)$. Using these expressions, (12.3.7) can be rewritten as

$$u(x,t) = u(n\Delta x, \tau\Delta t) = \hat{u}(k,\omega)\, e^{i(kn\Delta x-\omega\tau\Delta t)}. \tag{12.3.8}$$

Substituting the above equation into the finite difference equation, (12.3.5), yields

$$\hat{u}(k,\omega)\left(e^{i(kn\Delta x-\omega(\tau+1)\Delta t)} - e^{i(kn\Delta x-\omega\tau\Delta t)}\right)$$

$$+\frac{c\Delta t}{2\Delta x}\hat{u}(k,\omega)\left(e^{i(k(n+1)\Delta x-\omega\tau\Delta t)} - e^{i(k(n-1)\Delta x-\omega\tau\Delta t)}\right) = 0, \tag{12.3.9}$$

or

$$e^{-i\omega\Delta t} = 1 - iC\sin k\Delta x, \tag{12.3.10}$$

where

$$C = \frac{c\Delta t}{\Delta x}, \tag{12.3.11}$$

is called the *Courant number*. Substituting $\omega = \omega_r + i\omega_i$, where both ω_r and ω_i are real numbers, on the left side of (12.3.10) yields

$$e^{-i\omega\Delta t} = e^{\omega_i \Delta t} e^{-i\omega_r \Delta t}. \tag{12.3.12}$$

The first term on the right side of the above equation represents the wave amplitude change in one time step Δt, while the second term represents the phase change per time step since the first term is a real number and the second term is an imaginary number. Let $\lambda = e^{\omega_i \Delta t}$, (12.3.10) becomes

$$\lambda e^{-i\omega_r \Delta t} = 1 - i\,C \sin k\Delta x. \tag{12.3.13}$$

Equating the real and imaginary parts yields

$\lambda \cos\omega_r \Delta t = 1$, and

$$\lambda \sin \omega_r \Delta t = C \sin k\Delta x. \tag{12.3.14}$$

Summing the squares of the above two equations gives

$$\lambda = \pm\sqrt{1 + C^2 \sin^2 k\Delta x}. \tag{12.3.15}$$

Combining (12.3.8), (12.3.12) and (12.3.13) leads to

$$u(x,t) = \hat{u}(k,\omega)\left(e^{ikn\Delta x} e^{-i\omega_r \tau \Delta t}\right)\lambda^{\tau}. \tag{12.3.16}$$

In the above equation, only the last term on the right side, λ^{τ}, may change the amplitude as time proceeds. The other two terms inside the bracket, $e^{ikn\Delta x}$ and $e^{-i\omega_r \tau \Delta t}$, can only change the phase of these waves. In order for the *numerical stability* to occur for (12.3.16), $|\lambda| < 1$ is required. However, (12.3.15) implies that the absolute value of λ is

always greater than 1. Thus, the amplitude will grow with time and the *scheme of forward-in-time and centered-in-space is unconditionally unstable* since any small perturbations will grow indefinitely given enough time, and this type of *stability analysis* should be conducted prior to adopting a numerical scheme to approximate a differential equation.

(b) Forward-in-time and upstream-in-space scheme

Another two-time-level scheme that has been adopted in mesoscale models is the *forward-in-time and upstream-in-space scheme*. Under this scheme, the advection equation, (12.3.1), is approximated by

$$\frac{u_i^{\tau+1}-u_i^{\tau}}{\Delta t}=\begin{cases}-c\dfrac{u_i^{\tau}-u_{i-1}^{\tau}}{\Delta x}, & \text{if } c>0 \qquad (12.3.17a)\\ -c\dfrac{u_{i+1}^{\tau}-u_i^{\tau}}{\Delta x}, & \text{if } c\le 0 \qquad (12.3.17b)\end{cases}.$$

To check the stability of this scheme, we consider a positive constant advection velocity, *c>0*, without loss of generality and substituting (12.3.8) into (12.3.17a)

$$e^{-i(\omega_r+i\omega_i)\Delta t}=1-C\ \left(1-e^{-ik\Delta x}\right), \qquad (12.3.18)$$

or

$$\lambda\cos\omega_r\Delta t=1-C\left[1-\cos k\Delta x\right], \qquad (12.3.19a)$$

$$\lambda\sin\omega_r\Delta t=C\sin k\Delta x\,. \qquad (12.3.19b)$$

Summing the squares of the above equations yields

$$\lambda\ =\pm\sqrt{1+2C(\cos k\Delta x-1)(1-C)}\,. \qquad (12.3.20)$$

To insure the numerical stability ($|\lambda|<1$), the above equation requires

$$2C(\cos k\Delta x - 1)(1 - C) \le 0 . \qquad (12.3.21)$$

It holds if $C \le 1$ since $(\cos k\Delta x - 1)$ is always negative. Thus, (12.3.21) requires $\Delta t \le \Delta x / c$ for the scheme to be stable, called the *CFL* (Courant-Friedrichs-Lewy) *stability criterion*. Thus, *the numerical scheme of forward-in-time and upstream-in-space is conditionally stable.*

The phase of a wave can also be altered by the application of numerical methods to solve differential equations. We can investigate the phase characteristics of the forward-in-time and upstream-in-space scheme by dividing (12.3.19b) by (12.3.19a) to obtain

$$\tan \omega_r \Delta t = \frac{C \sin k\Delta x}{1 + C(\cos k\Delta x - 1)} . \qquad (12.3.22)$$

Based on the above equation, the *numerical phase speed* is

$$\tilde{c}_p = \frac{\omega_r}{k} = \frac{1}{k\Delta t} \tan^{-1}\left[\frac{C \sin k\Delta x}{1 + C(\cos k\Delta x - 1)} \right]. \qquad (12.3.23)$$

Equation (12.3.23) indicates that the finite difference scheme of forward-in-time and upstream-in-space is dispersive because the numerical phase speed is a function of the wave number. Similar to what occurs in the physical dispersion, here waves with different wavelengths propagate at different speeds. The wave therefore cannot preserve its original wave pattern, making it a *dispersive wave* (Ch. 3). Interestingly enough, wave dispersion can be induced numerically as well as physically.

Based on the advection equation, (12.3.1), and (12.3.7), the physical phase speed for the advection equation can be obtained,

$$c_p = \frac{\omega}{k} = c . \qquad (12.3.24)$$

According to the above equation, the wave is physically nondispersive since its physical phase speed is independent of the wave number. However, the numerical method applied here does introduce a numerical wave mode and makes the wave dispersive artificially. The ratio of the *numerical phase speed* ($\tilde{c}_p$) to the *physical phase speed* (c_p) is

$$\frac{\tilde{c}_p}{c_p} = \frac{1}{kc\Delta t}\tan^{-1}\left[\frac{C\sin k\Delta x}{1+C(\cos k\Delta x - 1)}\right], \tag{12.3.25}$$

which indicates that

$\tilde{c}_p > c_p$ when $0.5 < C < 1.0$ and,

$\tilde{c}_p < c_p$ when $0 < C < 0.5$. (12.3.26)

A numerical method may introduce, in addition to numerical instability and numerical dispersion, *numerical damping*. For example, when C = 0, 1, or $k \approx 0$ (very long waves), we have λ = 1. The aforementioned case means that the amplitude will be kept the same, which indicates that no damping exists under these special conditions. However, this scheme tends to damp waves in general, especially at C = 0.5. To demonstrate the damping characteristics, we use a truncated Taylor series approximation to the advection equation, (12.3.1),

$$u_i^{\tau+1} \cong u_i^{\tau} + \frac{\partial u}{\partial t}\Delta t + \frac{1}{2!}\frac{\partial^2 u}{\partial t^2}\Delta t^2, \text{ and} \tag{12.3.27}$$

$$u_{i-1}^{\tau} \cong u_i^{\tau} - \frac{\partial u}{\partial x}\Delta x + \frac{1}{2!}\frac{\partial^2 u}{\partial x^2}\Delta x^2. \tag{12.3.28}$$

Again, we may assume $c > 0$ without loss of generality and substitute the above approximations into (12.3.17a),

$$\frac{\left(u_i^\tau + \frac{\partial u}{\partial t}\Delta t + \frac{1}{2!}\frac{\partial^2 u}{\partial t^2}\Delta t^2\right) - u_i^\tau}{\Delta t} = -c\left[\frac{u_i^\tau - \left(u_i^\tau - \frac{\partial u}{\partial x}\Delta x + \frac{1}{2!}\frac{\partial^2 u}{\partial x^2}\Delta x^2\right)}{\Delta x}\right],$$

which can be rearranged to be

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} + \frac{1}{2}\frac{\partial^2 u}{\partial t^2}\Delta t - \frac{1}{2}\frac{\partial^2 u}{\partial x^2}c\Delta x = 0\,. \tag{12.3.29}$$

In addition to the original advection equation, this particular numerical scheme artificially introduces two additional terms that are caused by truncation errors. If both Δt and Δx approach 0, then the above equation reduces to the original differential equation. Under normal conditions, however, they have non-zero values that introduce errors to the numerical solution.

One can prove that the last two terms are related to each other because

$$\frac{\partial^2 u}{\partial t^2} = c^2\frac{\partial^2 u}{\partial x^2}\,. \tag{12.3.30}$$

Substituting the above equation into (12.3.29) leads to

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = \nu_c\frac{\partial^2 u}{\partial x^2}, \quad \nu_c \equiv \frac{1}{2}c\Delta x(1-C), \qquad \mathrm{C} = \frac{c\Delta t}{\Delta x}\,. \tag{12.3.31}$$

The solution obtained from the finite difference equation is similar to that of the above differential equation, which contains a spurious diffusion term. The solution obtained, therefore, tends to be damped. The term ν_c is called the *numerical diffusion coefficient*. In the early days of numerical model development, the forward-in-time and upstream-in-space scheme was used extensively due to its two-time-level simplicity and its low memory storage requirement. However, its strong numerical damping characteristics and

failure to preserve the proper phase have generated serious criticism. The technique is acceptable if advection or wave propagation is not dominant for a particular mesoscale phenomenon. However, if the subgrid mixing is important, ν_c must be smaller than the corresponding physically relevant turbulent exchange coefficient in order to avoid excess damping. Thus, the development of increasingly accurate three-time-level schemes and the advancement of computing facility make this scheme less attractive to mesoscale modelers.

(c) Lax-Wendroff scheme

This scheme was originally proposed by Lax and Wendroff (1960). The procedure for computation is based on the grid stencil shown in Fig. 12.5 and is described as follows. First, provisional values of *u* at provisional time step $\tau+1/2$ and grid points $i-1/2$ and $i+1/2$ are calculated at the points denoted by the cross symbol by applying the forward in time and centered in space scheme:

$$\frac{u_{i+1/2}^{\tau+1/2}-(u_{i+1}^{\tau}+u_i^{\tau})/2}{\Delta t/2}=-c\frac{u_{i+1}^{\tau}-u_i^{\tau}}{\Delta x}, \tag{12.3.32a}$$

$$\frac{u_{i-1/2}^{\tau+1/2}-(u_i^{\tau}+u_{i-1}^{\tau})/2}{\Delta t/2}=-c\frac{u_i^{\tau}-u_{i-1}^{\tau}}{\Delta x}. \tag{12.3.32b}$$

Then, applying the second-order centered difference scheme in both time and space to values at grid points $u_{i+1/2}^{\tau+1/2}, u_{i-1/2}^{\tau+1/2}$, and u_i^{τ} gives,

$$\frac{u_i^{\tau+1}-u_i^{\tau}}{\Delta t}=-c\frac{u_{i+1/2}^{\tau+1/2}-u_{i-1/2}^{\tau+1/2}}{\Delta x}. \tag{12.3.33}$$

Finally, substituting the above provisional values of $u_{i+1/2}^{\tau+1/2}$ and $u_{i-1/2}^{\tau+1/2}$ from (12.3.32) into (12.3.33) leads to

$$u_i^{\tau+1} = u_i^{\tau} - \frac{C}{2}\left(u_{i+1}^{\tau} - u_{i-1}^{\tau}\right) + \frac{C^2}{2}\left(u_{i+1}^{\tau} - 2u_i^{\tau} + u_{i-1}^{\tau}\right). \tag{12.3.34}$$

From computational point of view, $u_{i+1/2}^{\tau+1/2}$ and $u_{i-1/2}^{\tau+1/2}$ are provisional since they do not show up in (12.3.34); if (12.3.34) is used directly, there is no need to calculate or store them at all. The scheme as given by (12.3.34) is a two-time-level scheme, just like the forward-in-time upstream-in-space scheme discussed earlier, and has therefore the same low storage requirement.

The Lax-Wendroff scheme has a truncation error of $O[\Delta x^2] + O[\Delta t^2]$, meaning it has second-order accuracy in space and time, which is an improvement over the previous two-time-level scheme. It can be derived that

$$|\lambda| = \left[1 - 4C^2\left(1 - C^2\right)\sin^4\frac{k\Delta x}{2}\right]^{1/2}. \tag{12.3.35}$$

Therefore, the Lax-Wendroff scheme is stable if

$$C^2 \le 1 \quad or \quad \frac{|c|\Delta t}{\Delta x} \le 1. \tag{12.3.36}$$

That is, (12.3.36) satisfies the CFL stability criterion, and it can be proven that the last term of (12.3.34) serves as a damping term. In fact, the Lax-Wendroff scheme can be viewed as a modification of the forward-in-time and centered-in-space scheme with damping. For the shortest resolvable wavelength $2\Delta x$, we have $k = \pi / \Delta x$. Substituting *k* into (12.3.35) yields

$$|\lambda| = \left|1 - 2C^2\right|. \tag{12.3.37}$$

For the $4\Delta x$ wave, we have

$$|\lambda| = \left(1 - C^2 + C^4\right)^{1/2}. \tag{12.3.38}$$

Thus, the amount of damping is quite large for shorter waves.

The phase error, $1 - \tilde{c}_p / c$, can also be calculated from

$$\frac{\tilde{c}_p}{c} = \frac{\tan^{-1}\left\{-C \sin k\Delta x / \left(1 - C^2\left(1 - \cos k\Delta x\right)\right)\right\}}{-C\ k\ \Delta x}. \tag{12.3.39}$$

Since $\tilde{c}_p$ is a function of wave number (*k*), *the Lax-Wendroff scheme* is *numerically dispersive*. The scheme has a predominantly lagging phase error except in cases of large wave numbers where $\sqrt{0.5} < C < 1$.

The Lax-Wendroff scheme has been modified by the following formula and is also known as the *Crowley scheme* (1968):

$$u_i^{\tau+1} = u_i^{\tau} - \frac{C}{2}\left(u_{i+1}^{\tau} - u_{i-1}^{\tau}\right) + \frac{C^2}{2}\left(u_{i+1}^{\tau} - 2u_i^{\tau} + u_{i-1}^{\tau}\right) + \frac{C}{12}\left(1 - C^2\right)\left(u_{i+2}^{\tau} - 2u_{i+1}^{\tau} + 2u_{i-1}^{\tau} - u_{i-2}^{\tau}\right) \tag{12.3.40}.$$

The last term in the above equation is the third-order space correction term.

(d) Multi-stage schemes

The advection equation (12.3.1) can be generalized through the following form:

$$\frac{\partial u}{\partial t} = F(u) \tag{12.3.41}$$

where $F(u)$ is the forcing term that includes the advection term of (12.3.1). To improve the accuracy of two-time-level schemes, the *multi-stage scheme* can be used to approximate (12.3.41),

$$\tilde{u}^{\tau+\alpha} = u^{\tau} + \alpha \Delta t F(u^{\tau}),$$

$$u^{\tau+1} = u^{\tau} + \Delta t \left[\beta F(\tilde{u}^{\tau+\alpha}) + (1-\beta) F(u^{\tau})\right]. \tag{12.3.42}$$

The above method reduces to the second-order *Runge-Kutta schemes* for any combinations of α and β leading to $\alpha\beta = 1/2$, which leads to a special scheme called the *Heun scheme* when $\alpha = 1,\ \beta = 1/2$. An example of a non-Runge-Kutta scheme is the *forward-backward* (*Matsuno*) *scheme* for which $\alpha = \beta = 1$ (Matsuno 1966).

12.3.2 Three-time-level schemes

(a) Adams-Bashforth scheme

Under the Adams-Bashforth scheme, (12.3.41) is approximated by

$$u^{\tau+1} = u^{\tau} + \Delta t \left(\frac{3}{2} F(u^{\tau}) - \frac{1}{2} F(u^{\tau-1}) \right). \tag{12.3.43}$$

The advantage of this scheme is that it generates neither the time splitting produced by the leapfrog scheme nor the numerical diffusion produced by the upstream difference (e.g., Lilly 1965; Durran 1998). The nonlinear advection terms and energy components may generate large errors with this scheme.

(b) Leapfrog-in-time and centered-in-space schemes

The advection equation can also be approximated by the leapfrog (second-order centered) in time and second-order centered difference in space scheme,

$$\frac{u_i^{\tau+1} - u_i^{\tau-1}}{2\Delta t} = -c \frac{u_{i+1}^{\tau} - u_{i-1}^{\tau}}{2\Delta x}. \tag{12.3.44}$$

Again, in order to examine the stability of this scheme, we substitute the wave solution (12.3.8) into (12.3.44). This yield

$$\left(\lambda e^{-i\omega_r \Delta t} - \frac{1}{\lambda e^{-i\omega_r \Delta t}}\right) = -C\, 2i \sin k\Delta x, \quad \lambda \equiv e^{\omega_i \Delta t}; \quad C \equiv \frac{c\Delta t}{\Delta x}. \tag{12.3.45}$$

The above equation can be rearranged to obtain

$$\lambda^2 e^{-2i\omega_r \Delta t} + 2i\alpha\lambda e^{-i\omega_r \Delta t} - 1 = 0, \tag{12.3.46}$$

where $\alpha = C \sin k\Delta x$ is a temporary parameter. Regarding $\lambda e^{-i\omega_r \Delta t}$ as the unknown in the above equation, we obtain

$$\lambda e^{-i\omega_r \Delta t} = -i\alpha \pm \sqrt{1-\alpha^2}\,. \tag{12.3.47}$$

Separating the real and imaginary parts of the above equation gives two possible cases, namely, (1) $\alpha^2 \le 1$ and (2) $\alpha^2 > 1$. In case 1, we have

$$\lambda \cos \omega_r \Delta t = \pm\sqrt{1-\alpha^2}\,, \tag{12.3.48a}$$

$$\lambda \sin \omega_r \Delta t = \alpha\,. \tag{12.3.48b}$$

Summing the squares of the above two equations yields

$$|\lambda| = 1\,.$$

Therefore, the *leapfrog- in-time and centered-in-space scheme is neutral when* $\alpha^2 \le 1$.

In case 2, we have

$$\lambda \cos\omega_r \Delta t = 0\,, \tag{12.3.49a}$$

$$\lambda \sin \omega_r \Delta t = \alpha \mp \sqrt{\alpha^2 - 1}\,. \tag{12.3.49b}$$

Summing the squares of the above two equations leads to

$$\lambda^2 = \left(\alpha \mp \sqrt{\alpha^2 - 1}\right)^2 \qquad \text{if } \alpha^2 > 1\,. \tag{12.3.50}$$

We may claim that this scheme is unstable when $\alpha^2 > 1$. To prove that this scheme is unstable when $\alpha^2 > 1$, we only need to find one counter example for which $|\lambda| > 1$. We can assume $\alpha = 1 + \varepsilon$, where ε is a small positive number. Substituting α into the positive root of (12.3.50) gives

$$|\lambda| = 1 + \varepsilon \mp \sqrt{2\varepsilon + \varepsilon^2}\ . \tag{12.3.51}$$

Since either root is possible, we look at the solution with the positive root,

$$|\lambda| = 1 + \varepsilon + \sqrt{2\varepsilon + \varepsilon^2}\ . \tag{12.3.52}$$

The above equation gives $|\lambda| > 1$. Therefore, *the leapfrog-in-time and centered-in-space scheme is unstable when* $\alpha^2 > 1$. The stability is thus retained only when $\alpha^2 \le 1$. Based on the definition of α for (12.3.46), it requires

$$C^2 \sin^2 k\Delta x \le 1. \tag{12.3.53}$$

Since the maximum value of the sine square function is 1, the above equation is satisfied when

$$|C| \le 1. \tag{12.3.54}$$

In fact, the CFL criterion is not only a necessary condition but also a sufficient condition for the numerical stability of the leapfrog-in-time and centered-in-space scheme.

To demonstrate the phase characteristics associated with the leapfrog in time and second-order centered in space scheme, we divide (12.3.48b) by (12.3.48a):

$$\omega_r \Delta t = \tan^{-1}\left(\frac{\pm\alpha}{\sqrt{1-\alpha^2}}\right), \tag{12.3.55}$$

which gives us the numerical phase speeds

$$\tilde{c}_p = \frac{\omega_r}{k} = \frac{\pm 1}{k\Delta t} \tan^{-1}\left(\frac{\alpha}{\sqrt{1-\alpha^2}} \right). \tag{12.3.56}$$

The *phase error* can be obtained by comparing the numerical phase speed and the physical phase speed,

$$\frac{\tilde{c}_p}{c} = \frac{\pm 1}{kc\Delta t} \tan^{-1}\left(\frac{\alpha}{\sqrt{1-\alpha^2}} \right). \tag{12.3.57}$$

For $c > 0$, (12.3.56) has two solutions, one propagating to the right ($\tilde{c}_p > 0$), and the other propagating to the left ($\tilde{c}_p < 0$). The first solution represents the *physical mode* because it approximates the solution to the original advection equation. The second solution represents the *computational mode*, which is purely generated by the numerical scheme. If the computational mode is not damped, it slowly amplifies and eventually becomes unstable while it propagates in an opposite direction (left) to the physical mode during the simulation of wave propagation. The behavior of the computational (numerical) mode is a phenomenon known as *time-splitting*, and this scheme also induces numerical dispersion since the numerical phase speed is a function of wave number.

In summary, the leapfrog-in-time and second-order centered-in-space scheme preserves the amplitude when $\alpha^2 \le 1$ (where $\alpha = C \sin k\Delta x$) but can generate phase errors. Figure 12.6 compares the numerical solution of the leapfrog-in-time and second-order centered-in-space scheme to that of the forward-in-time and upstream-in-space scheme and the analytical solution. The numerical solutions are initialized by a rectangular wave centered at $x = 0$. The figure indicates that this scheme preserves the amplitude of the initial rectangular wave much better than the forward-in-time and

upstream-in-space scheme. However, it produces more severe numerical dispersion than the forward-in-time and upstream-in-space scheme (e.g., Haltiner and Williams 1980).

In addition to the second-order centered schemes for spatial difference, a scheme with higher-order accuracy can be derived. For example, consider the following Taylor series expansions for $f(x+\Delta x)$ and $f(x-\Delta x)$,

$$f(x+\Delta x) = f(x) + f'(x)\Delta x + f''(x)\frac{\Delta x^2}{2!} + f'''(x)\frac{\Delta x^3}{3!} + ... \quad (12.3.58)$$

$$f(x-\Delta x) = f(x) - f'(x)\Delta x + f''(x)\frac{\Delta x^2}{2!} - f'''(x)\frac{\Delta x^3}{3!} + ... \quad (12.3.59)$$

Subtracting (12.3.59) from (12.3.58) leads to

$$f(x+\Delta x) - f(x-\Delta x) = 2f'(x)\Delta x + \frac{1}{3}f'''(x)\Delta x^3 + ... \quad (12.3.60)$$

Now, consider the following Taylor series expansions for $f(x+2\Delta x)$ and $f(x-2\Delta x)$,

$$f(x+2\Delta x) = f(x) + 2f'(x)\Delta x + f''(x)\frac{4\Delta x^2}{2!} + f'''(x)\frac{8\Delta x^3}{3!} + ... \quad (12.3.61)$$

$$f(x-2\Delta x) = f(x) - 2f'(x)\Delta x + f''(x)\frac{4\Delta x^2}{2!} - f'''(x)\frac{8\Delta x^3}{3!} + ... \quad (12.3.62)$$

Subtracting (12.3.62) from (12.3.61) leads to

$$f(x+2\Delta x) - f(x-2\Delta x) = 4f'(x)\Delta x + \frac{8}{3}f'''(x)\Delta x^3 + ... \quad (12.3.63)$$

Eliminating *f'''(x)* terms from (12.3.60) - (12.3.63) yields

$$f'(x) = \frac{8\left[f(x+\Delta x) - f(x-\Delta x)\right] - \left[f(x+2\Delta x) - f(x-2\Delta x)\right]}{12\Delta x} + O(\Delta x^4). \quad (12.3.64)$$

Equation (12.3.64) gives a *fourth-order centered difference* scheme for $f'(x)$. Note that the boundary points are approximated by adjacent interior points. It can be shown that

(12.3.64) can be obtained by extrapolation for the value $2\Delta x/3$ of the quotients of $f'(x)$ from (12.3.62) and (12.3.63).

Now apply the fourth-order centered difference scheme in space and the leapfrog scheme in time to the advection equation (12.3.1),

$$\frac{u_i^{\tau+1}-u_i^{\tau-1}}{2\Delta t}+c\left(\frac{8\,(u_{i+1}^{\tau}-u_{i-1}^{\tau})-(u_{i+2}^{\tau}-u_{i-2}^{\tau})}{12\Delta x}\right)=0\,. \tag{12.3.65}$$

Solve for $u_i^{\tau+1}$,

$$u_i^{\tau+1}=u_i^{\tau-1}-\frac{C}{6}\left(8(u_{i+1}^{\tau}-u_{i-1}^{\tau})-(u_{i+2}^{\tau}-u_{i-2}^{\tau})\right), \tag{12.3.66}$$

where *C* is the *Courant number*, as defined earlier. It can be shown that

$$\frac{\tilde{c}_p}{c}=\frac{4}{3}\frac{\sin k\Delta x}{k\Delta x}-\frac{1}{3}\frac{\sin 2k\Delta x}{2k\Delta x} \tag{12.3.67}$$

for the fourth-order centered difference scheme. Compared with that of the second-order centered in space scheme,

$$\frac{\tilde{c}_p}{c}=\frac{\sin k\Delta x}{k\Delta x}\,, \tag{12.3.68}$$

we have

$$\tilde{c}_p=c\left[1-\frac{4}{5!}(k\Delta x)^2+\ldots\right] \quad \text{for the fourth -order scheme, and} \tag{12.3.69}$$

$$\tilde{c}_p=c\left[1-\frac{1}{3!}(k\Delta x)^2+\ldots\right] \quad \text{for the second-order scheme.} \tag{12.3.70}$$

Both schemes are therefore numerically dispersive. However, using the 4th-order scheme greatly increases in accuracy of the phase speed for longer waves (smaller *k*) (Fig. 12.7).

In addition, for shorter waves there is more numerical dispersion associated with the fourth-order scheme since the slope of c_4 is larger than c_2.

12.4 Implicit schemes

With the above finite difference schemes, the advection term is evaluated at time step τ, thus the variables at time step $\tau+1$ can be predicted explicitly by those at time step τ and/or $\tau-1$. These schemes are referred to as *explicit schemes*. However, all explicit advection schemes are, at most, conditionally stable numerically, and the CFL condition type stability criterion imposes a severe restriction on the time interval with a resultant increase in computational time, and this restriction can be relaxed by evaluating the advection term at time step $\tau+1$. For example, the spatial differencing of (12.3.5), i.e. the forward-in-time and second-order centered in space scheme, can be applied at time step $\tau+1$,

$$\frac{u_i^{\tau+1}-u_i^{\tau}}{\Delta t}+c\left(\frac{u_{i+1}^{\tau+1}-u_{i-1}^{\tau+1}}{2\Delta x}\right)=0, \tag{12.4.1}$$

and this method is called the *Euler implicit method* (e.g., Tannehill et al. 1997). In order to solve for u at time step $\tau+1$, we move all of them to the left side

$$-\frac{C}{2}u_{i-1}^{\tau+1}+u_i^{\tau+1}+\frac{C}{2}u_{i+1}^{\tau+1}=u_i^{\tau}, \quad i=1,2,3,\ldots N-1, \tag{12.4.2}$$

where N is the total grid number. Thus, one cannot solve the equation for a general point, $u_i^{\tau+1}$, alone. Instead, we have to solve the system of algebraic equations, as shown in Fig. 12.8. In Fig. 12.8, we have assumed that boundary conditions are as follows: $u(0,t)=u_0^{\tau+1}=0$ and $u[N\Delta x,t]=u_N^{\tau+1}=0$. If a zero-gradient boundary condition

($\partial u/\partial x = 0$) is imposed, then the coefficients at the upper left and lower right corners of the matrix in Fig. 12.8 become $1-C/2$ and $1+C/2$, respectively. In general, one can introduce a weighting factor α and replace (12.4.2) by

$$-\frac{\alpha C}{2}u_{i-1}^{\tau+1}+u_i^{\tau+1}+\frac{\alpha C}{2}u_{i+1}^{\tau+1}=\frac{(1-\alpha)C}{2}u_{i-1}^{\tau}+u_i^{\tau}-\frac{(1-\alpha)C}{2}u_{i+1}^{\tau}, \qquad i=1,2,3,...N-1. \tag{12.4.3}$$

If $\alpha = 0$, the above formula reduces to the completely explicit scheme, i.e. the forward-in-time and centered-in-space scheme, (12.3.6). If $\alpha = 1$, the Euler implicit scheme, (12.4.2) is recovered. To find out the numerical stability, we substitute (12.3.8) into (12.4.2) to obtain

$$|\lambda| = \frac{1}{\sqrt{1+C^2\sin^2 k\Delta x}}. \tag{12.4.4}$$

The above equation implies that *the implicit Euler scheme is unconditionally stable* because the right hand side of (12.4.4) is always less than 1. In general, the use of an implicit scheme permits larger time steps than the explicit form without causing numerical instability. To invert the matrix, either direct (e.g., Gaussian elimination, LU decomposition) or iterative (e.g., Jacobi, Gauss-Seidel, relaxation) methods can be applied. Discussions of these methods can be found in numerical analysis textbooks.

To lessen the computational burden, *semi-implicit schemes* have been developed. In a semi-implicit scheme, terms which are primarily responsible for the propagation of faster waves (e.g., gravity waves) are treated implicitly, while other terms are treated explicitly. For example, the linear shallow water *x*-momentum equation, (3.4.7), can be shown by the *trapezoidal semi-implicit scheme* (with primes dropped) as

$$\left(\frac{u_i^{\tau+1}-u_i^{\tau}}{\Delta t}\right)+U\left(\frac{u_{i+1}^{\tau}-u_{i-1}^{\tau}}{2\Delta x}\right)+\frac{g}{2}\left[\left(\frac{h_{i+1}^{\tau}-h_{i-1}^{\tau}}{2\Delta x}\right)+\left(\frac{h_{i+1}^{\tau+1}-h_{i-1}^{\tau+1}}{2\Delta x}\right)\right]=0. \qquad (12.4.5)$$

Note that the advection term is treated in explicit manner and the spatial derivative is centered at $\tau+1/2$ time step by averaging values at time steps τ and τ+1, and it can be shown that the *trapezoidal semi-implicit scheme* is unconditionally stable (Mesinger and Arakawa, 1976). Although theoretically a very large Courant number (or say, time interval) can be used with implicit schemes, practically there is a limit in its use. For example, the trapezoidal semi-implicit scheme has a serious phase error when the Courant number is large (Haltiner and Williams, 1980).

12.5 Semi-Lagrangian methods

Ideally, one should be able to integrate the advection equation by following the fluid particles in a Lagrangian manner, so that the local rate of change and advection terms do not have to be considered separately. In fact, taking a *Lagrangian approach*, a graphical method has been developed to solve the barotropic vorticity equation using a single time step of 24 h by following a set of fluid particles (Fjortoft 1952). However, in general a set of fluid particles, which are initially distributed regularly, will soon become greatly deformed and are thus rendered unsuitable for numerical integration (Welander 1955). To avoid this difficulty, the *semi-Lagrangian method* (occasionally referred to as *quasi-Lagrangian method*) whereby a set of particles that arrive at a regular set of grid points are traced backward over a single time step to their departure points was proposed (Wiin-Nielsen 1959). The values of the dynamical quantities at the departure points are obtained by interpolating known values at neighboring grid points. Note that in a semi-

Lagrangian method, the set of fluid particles in question changes at each time step, which is different from the pure Lagrangian method. In addition, a combination of these schemes, i.e. *semi-Lagrangian semi-implicit scheme*, has been proposed (Robert 1982; Staniforth and Côté 1991).

To examine the stability property of the semi-Lagrangian method, we consider the one-dimensional nonlinear advection equation in the form of total derivative,

$$\frac{D\psi}{Dt} = 0, \tag{12.5.1}$$

where $D/Dt \equiv \partial/\partial t + u\partial/\partial x$ and ψ is any variable under consideration. By integrating over the trajectory of a fluid particle that arrives at a grid point $i\Delta x$, denoted as P in Fig. 12.9, and at time $(\tau+1)\Delta t$, we have

$$\psi_i^{\tau+1} = \psi_*^{\tau}, \tag{12.5.2}$$

where ψ_*^{τ} is the value of ψ at the departure point of the particle at time $\tau\Delta t$. The value ψ_*^{τ} is obtained by polynomial interpolation from the neighboring grid points. The stability and accuracy of the scheme depends on the interpolation method used. For example, we may consider the linear interpolation from the surrounding grid points $(i-p)$ and $(i-p-1)$ for ψ_*^{τ},

$$\frac{\psi_{i-p}^{\tau} - \psi_*^{\tau}}{u\Delta t - p\Delta x} = \frac{\psi_{i-p}^{\tau} - \psi_{i-p-1}^{\tau}}{\Delta x}, \tag{12.5.3}$$

where *u* is the advection velocity as represented in (12.5.1). The above equation may be rearranged as

$$\psi_*^{\tau} = \psi_{i-p}^{\tau} - \left(\frac{u\Delta t}{\Delta x} - p\right)\left(\psi_{i-p}^{\tau} - \psi_{i-p-1}^{\tau}\right). \tag{12.5.4}$$

or

$$\psi_*^\tau = \psi_{i-p}^\tau - \hat{\alpha}\left(\psi_{i-p}^\tau - \psi_{i-p-1}^\tau\right), \tag{12.5.5}$$

where

$$\hat{\alpha} = \alpha - p, \qquad \alpha = u\Delta t / \Delta x . \tag{12.5.6}$$

Therefore, from (12.5.2) we have

$$\psi_i^{\tau+1} = \psi_{i-p}^\tau - \hat{\alpha}\left(\psi_{i-p}^\tau - \psi_{i-p-1}^\tau\right). \tag{12.5.7}$$

According to (12.5.6) and Fig. 12.9, $\hat{\alpha}$ is the fractional part, and p is the integral part after advection of a non-dimensional distance $u\Delta t / \Delta x$.

To examine whether the semi-Lagrangian method is numerically stable or not, we may again assume a wave-like solution,

$$\psi_i^\tau = \hat{\psi}\, e^{-i\omega_r \tau \Delta t} e^{ikn\Delta x} \lambda^\tau . \tag{12.5.8}$$

Substituting (12.5.8) into (12.5.7) yields

$$\lambda^2 = 1 - 2\hat{\alpha}(1-\hat{\alpha})(1-\cos k\Delta x), \quad \lambda \equiv e^{\omega_i \Delta t} . \tag{12.5.9}$$

Thus, in order to have a numerically stable solution ($|\lambda| \le 1$), we require

$$0 \le \hat{\alpha} \le 1 . \tag{12.5.10}$$

That is, the departure points must lie within the interpolation interval (*i-p*-1, *i-p*): however, this is just the choice of points used for interpolation. Therefore, *the semi-Lagrangian scheme is unconditionally stable*.

The *semi-implicit method* may be incorporated into the integration by treating the other terms, such as the pressure gradient force term in the momentum equation, as time averages along the trajectory, while the total time derivative is evaluated by either

leapfrog, forward or other time difference schemes. To elucidate this, let us consider the following Boussinesq, horizontal momentum equation:

$$\frac{Du}{Dt} - fv + \frac{1}{\rho_o}\frac{\partial p}{\partial x} = 0. \tag{12.5.11}$$

The total derivative of the above equation may be approximated by the forward-in-time scheme,

$$\frac{Du}{Dt} = \frac{u(x,t+\Delta t) - u(x-a,t)}{\Delta t}, \tag{12.5.12}$$

where

$$a = u(x-a,t)\Delta t. \tag{12.5.13}$$

Equation (12.5.12) can be solved by using an iterative method to obtain the upstream displacement or the departure point, *a*. We can apply the semi-implicit approximation to the other terms on the left hand side of (12.5.11)

$$\psi_{av}^{t} = \frac{\psi(x,t+\Delta t) + \psi(x-a,t)}{2}, \tag{12.5.14}$$

where the subscript *av* denotes the time average. Then the horizontal momentum equation can be approximated by the *semi-implicit semi-Lagrangian scheme*,

$$\frac{u(x,t+\Delta t) - u(x-a,t)}{\Delta t} - fv_{av}^{t} + \left(\frac{1}{\rho_o}\frac{\partial p}{\partial x}\right)_{av}^{t} = 0. \tag{12.5.15}$$

The above equation can also be rewritten as

$$\left([u]_{x}^{t+\Delta t} - [u]_{x-a}^{t}\right) - \frac{f\Delta t}{2}\left([v]_{x}^{t+\Delta t} + [v]_{x-a}^{t}\right) + \frac{\Delta t}{2\rho_o}\left(\left[\frac{\partial p}{\partial x}\right]_{x}^{t+\Delta t} + \left[\frac{\partial p}{\partial x}\right]_{x-a}^{t}\right) = 0. \tag{12.5.16}$$

Moving all terms at time $t+\Delta t$ to the left hand side gives the followingsemi-implicit semi-Lagrangian equation:

$$[u]_x^{t+\Delta t} - \frac{f\Delta t}{2}[v]_x^{t+\Delta t} + \frac{\Delta t}{2\rho_o}\left[\frac{\partial p}{\partial x}\right]_x^{t+\Delta t} = [u]_{x-a}^{t} + \frac{f\Delta t}{2}[v]_{x-a}^{t} - \frac{\Delta t}{2\rho_o}\left[\frac{\partial p}{\partial x}\right]_{x-a}^{t}. \quad (12.5.17)$$

This forms a set of linear algebraic equations, which can be written in a matrix form. Thus, a method for inverting the matrix can be applied to obtain the solution for time step $t + \Delta t$. Note that the advantage of the semi-implicit semi-Lagrangian scheme is that it is unconditionally stable so that a relatively large time step can be used. The disadvantage of this scheme is that the iterative method for finding the departure points and the method for inverting the matrix is computationally expensive.

Appendix 12.1: Formulas for Finite Difference Approximations of Derivatives (Adapted after Gerald and Wheatley 2003)

Formulas for the first derivative:

$$f'(x) = \frac{f(x+\Delta x) - f(x)}{\Delta x} + O(\Delta x), \qquad \text{(forward difference)}$$

$$f'(x) = \frac{f(x+\Delta x) - f(x-\Delta x)}{2\Delta x} + O(\Delta x^2), \qquad \text{(second-order centered difference)}$$

$$f'(x) = \frac{-f(x+2\Delta x) + 4f(x+\Delta x) - 3f(x)}{2\Delta x} + O(\Delta x^2), \qquad \text{(as above except for a left boundary point)}$$

$$f'(x) = \frac{-f(x+2\Delta x) + 8f(x+\Delta x) - 8f(x-\Delta x) + f(x-2\Delta x)}{12\Delta x} + O(\Delta x^4).$$

(fourth-order centered difference)

Formulas for the second derivative:

$$f''(x) = \frac{f(x+2\Delta x) - 2f(x+\Delta x) + f(x)}{\Delta x^2} + O(\Delta x), \qquad \text{(forward difference)}$$

$$f''(x) = \frac{f(x+\Delta x) - 2f(x) + f(x-\Delta x)}{\Delta x^2} + O(\Delta x^2), \quad \text{(second-order centered difference)}$$

$$f''(x) = \frac{-f(x+2\Delta x) + 16f(x+\Delta x) - 30f(x) + 16f(x-\Delta x) - f(x-2\Delta x)}{12\Delta x^2} + O(\Delta x^4).$$

(fourth-order centered difference)

$$f''(x) = \frac{-f(x+3\Delta x) + 4f(x+2\Delta x) - 5f(x+\Delta x) + 2f(x)}{\Delta x^2} + O(\Delta x^2), \text{ (as above except}$$

for a left boundary point)

Formulas for the third derivative:

$$f'''(x) = \frac{f(x+2\Delta x) - 2f(x+\Delta x) + 2f(x-\Delta x) - f(x-2\Delta x)}{2\Delta x^3} + O(\Delta x^2).$$

(centered difference)

$$f'''(x) = \frac{f(x+3\Delta x) - 3f(x+2\Delta x) + 3f(x+\Delta x) - f(x)}{\Delta x^3} + O(\Delta x), \text{ (for a left boundary}$$

point)

Formulas for the fourth derivative:

$$f^{iv}(x) = \frac{f(x+2\Delta x) - 4f(x+\Delta x) + 6f(x) - 4f(x-\Delta x) + f(x-2\Delta x)}{\Delta x^4} + O(\Delta x^2).$$

(centered difference)

$$f^{iv}(x) = \frac{f(x+4\Delta x) - 4f(x+3\Delta x) + 6f(x+2\Delta x) - 4f(x+\Delta x) + f(x)}{\Delta x^4} + O(\Delta x), \text{ (for a}$$

left boundary point)

References

Colella, P. and P. R. Woodward, 1984: The piecewise parabolic method (PPM) for gas dynamical simulations. *J. Comput. Phys.*, **54,** 174-201.

Crowley, W. P., 1968: Numerical advection experiments. *Mon. Wea. Rev.*, **96**, 1-11.

Durran, D. R., 1998: *Numerical Methods for Wave Equations in Geophysical Fluid Dynamics*. Springer-Verlag Inc., 465pp.

Fjortoft, 1952: On a numerical method of integrating the barotropic vorticity equation. *Tellus*, **4**, 179-194.

Gerald, C. F., and P. O. Wheatley, 2003: *Applied Numerical Analysis*. 7th Edn., Addison-Wesley Publ. Co., 624pp.

Haltiner, G. J., and R. T. Williams, 1980: *Numerical Prediction and Dynamic Meteorology*. 2nd Edn., John Wiley & Sons, Inc., 477pp.

Lax, P. and B. Wendroff, 1960: Systems of conservation laws. *Comm. Pure Appl. Math.*, **13**, 217-237.

Lilly, D. K., 1965: On the computational stability of numerical solutions of time-dependent non-linear geophysical fluid dynamics problems. *Mon. Wea. Rev.*, **93**, 11-26.

Matsuno, T., 1966: Numerical integrations of the primitive equations by a simulated backward difference method. *J. Meteor. Soc. Japan*, Ser. 2, **44**, 76-84.

Mesinger, F. and A. Arakawa, 1976: Numerical methods used in atmospheric models. Vol. 1, GARP Publ. Ser., **17**, WMO-ICSU, 64 pp.

Pielke, R. A., 2002: *Mesoscale Meteorological Modeling*. 2nd Edn., Academic Press, Inc., 676pp.

Robert, A., 1982: A semi-Lagrangian and semi-implicit numerical integration for the primitive meteorological equations. *J. Meteor. Soc. Japan*, **60**, 319-324.

Staniforth, A. and J. Côté, 1991: Semi-Lagrangian integration schemes for atmospheric models - A review, *Mon. Wea. Rev.*, **119**, 2206-2223.

Tannehill, J. C., D. A. Anderson, and R. H. Pletcher, 1997: *Computational Fluid Mechanics and Heat Transfer*. 2nd Edn., Taylor & Francis, 792pp.

Welander, P., 1955: Studies on the general development of motion in a two-dimensional, ideal fluid. *Tellus*, **7**, 141-156.

Wiin-Nielsen, A., 1959: On the application of trajectory methods in numerical forecasting. *Tellus*, **11**, 180-196.

Wurtele, M. G., 1961: On the problem of truncation error. *Tellus*, **13**, 379-391.

Problems

12.1 Make a sketch similar to Fig. 12.1 except for the backward difference scheme of (12.2.6).

12.2 Replace the sine wave by a cosine wave in (12.2.12), and show that you can obtain the same conclusion as (12.2.17).

12.3 Show that if $C = 0.5$, then $2\Delta x$ waves are completely eliminated by the forward in time and upstream in space scheme. Note that the $2\Delta x$ wave can be represented by a function with two constants, *a* and *b*: $u_i^\tau = a + (-1)^i b$, where i is an integer.

12.4 Prove that (12.3.26) by plotting $\tilde{c}_p / c_p$ as a function of the Courant number (C), according to (12.3.25).

12.5 Prove (12.3.34).

12.6 Derive (12.3.35).

12.7 Prove (12.3.67): $\frac{c_p}{c} = \frac{4}{3}\frac{\sin k\Delta x}{k\Delta x} - \frac{1}{3}\frac{\sin 2k\Delta x}{2k\Delta x}$.

12.8 Prove that the coefficients at the upper left and lower right corners of the matrix in Fig. 12.8 for the Euler implicit scheme become $1-C/2$ and $1+C/2$, respectively.

Modeling Projects

Project A

12.A1 An advection model (**advec1.f**), which is written in FORTRAN, solves a one-dimensional advection equation numerically. This model may be obtained from the website: http://www.cambridge.org/9780521808750. There is no need to be concerned with the numerical details of the model at this moment. Run the program to generate a data set and modify the plotting program (**sample_plot.f**) to plot the curves (not contours). You can adjust the flag "NPR" to write out the data for one or more time steps, and then plot these curves. Explain your results.

12.A2 Modify the program to have two different initial fields (e.g., the shape and amplitude of the initial field U1), plot and then explain the results.

12.A3 Repeat project 12.A2 for a nonlinear case by setting NL=1. Describe your results.

Project B

12.B1 Make some sensitivity tests on the Advection Model (Project A) to find out the maximum time interval (Δt) that gives a well-behaved solution. Construct a table

for the cases you have performed that shows the maximum amplitude of u' versus the time interval.

12.B2 Based on the Advection Model, develop it into a Tank Model with the one-layer shallow water equations

$$\frac{\partial u'}{\partial t} + (U + u')\frac{\partial u'}{\partial x} + g\frac{\partial h'}{\partial x} = 0,$$

$$\frac{\partial h'}{\partial t} + (U + u')\frac{\partial h'}{\partial x} + (H + h')\frac{\partial u'}{\partial x} = 0,$$

The above equation set has been discussed in Section 3.4. You may follow the following procedure for the model development:

(i) Define a new variable for h' with the same dimension as u' adding finite difference approximation of $(g\,\partial h'/\partial x)$ to the advection model by mimicking the approximation of $\partial u'/\partial x$.

(ii) Once the horizontal momentum equation, i.e. the first equation, is finished, then simply formulate the continuity equation, i.e. the second equation, by replacing u' by h' and g by $(H + h')$.

(iii) When you implement the second equation of h' into your model, you need to adopt a lateral boundary condition analogous to that used in u' by simply replacing *u'* by *h'*.

12.B3

(a) Run the Tank Model with $U = 0$ by giving an initial field, at your choice, in either u' or h',

(b) Repeat (a), but use a basic wind, i.e. $U \neq 0$.

Plot and explain your results. Perform both linear and nonlinear simulations by resetting the flag *NL*.

12.B4 Extend your tank model to a two-layer tank model with a bottom topography, i.e.

$$\frac{\partial u'}{\partial t} + (U + u')\frac{\partial u'}{\partial x} + g'\frac{\partial h'}{\partial x} = 0,$$

$$\frac{\partial h'}{\partial t} + (U + u')\frac{\partial h'}{\partial x} + (H + h' - h_s)\frac{\partial u'}{\partial x} = (U + u')\frac{\partial h_s}{\partial x}.$$

Then simulate a flow with $U = 10\,ms^{-1}$ over a bell-shaped bump, i.e. $h_s = h_m a^2/(x^2 + a^2)$. Describe and explain your results. Again, run for both linear and nonlinear cases. Perform several experiments to obtain both solutions with supercritical and subcritical flow regimes (see Sec. 3.4).

12. Basic numerical methods

Figure Captions

Fig. 12.1: A sketch of the forward finite difference scheme, based on (12.2.3).

Fig. 12.2: A sketch of the relationship of $f'(x)$ and its centered difference approximation, $f_D'(x)$.

Fig. 12.3: A sketch of the propagation of $u(x,t)$ along a constant phase line, x-$ct = \text{constant} = 0$.

Fig. 12.4: The grid system and algorithm for the forward-in-time and centered-in-space finite difference scheme of the advection equation. The values of u at $t = 0$ are provided by the initial condition (i.c.), and the values at the left and right boundaries are determined by the boundary conditions (b.c.).

Fig. 12.5: Grid stencil for the Lax-Wendroff scheme.

Fig. 12.6: An example of numerical damping and dispersion. Comparisons between the analytical solution (thin sold curve) and numerical solutions of applying the leapfrog in time and second-order centered in space scheme (dashed curve) and the forward in time and upstream in space scheme (heavy solid curve) to the advection equation with an initial rectangular wave centered at $x = 0$. Three nondimensional times are shown. (Adapted after Wurtele 1961)

Fig. 12.7: The numerical phase speeds of the second-order (c_2) and fourth-order (c_4) centered difference schemes. The phase speed of the linear advection equation (12.3.1) is c. (Adapted after Mesinger and Arakawa 1976)

Fig. 12.8: The system of algebraic equations of (12.4.2) for the Euler implicit method with boundary conditions, $u(0,t) = u_0^{\tau+1} = 0$ and $u(N\Delta x, t) = u_N^{\tau+1} = 0$, where N is the total number of grid intervals.

Fig. 12.9: A schematic of the semi-Lagrangian method. A fluid particle that arrives at a grid point $i\Delta x$ and at time $(\tau+1)\Delta t$ is denoted as P, which is located at x_* and at time $\tau\Delta t$. The value of the variable at this time and location (ψ_*^τ) is obtained by polynomial interpolation from the neighboring grid points, ψ_{i-p-1}^τ and ψ_{i-p}^τ, as expressed in (12.5.4) or (12.5.5).

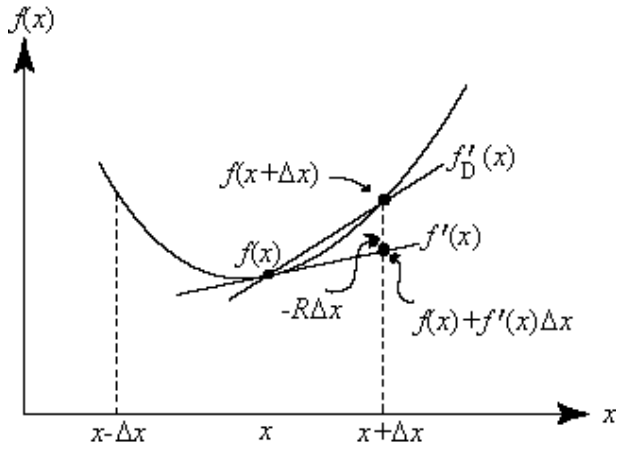

Fig. 12.1: A sketch of the forward finite difference scheme, based on (12.2.3).

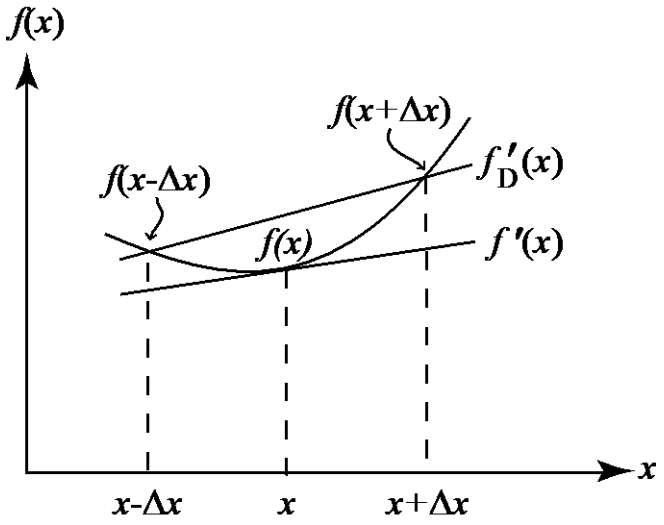

Fig. 12.2: A sketch of the relationship of $f'(x)$ and its centered difference approximation, $f_D'(x)$.

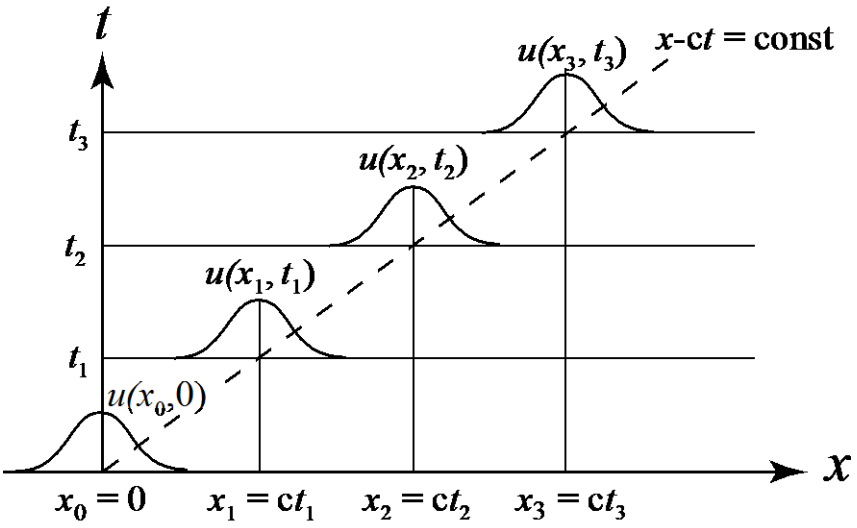

Fig. 12.3: A sketch of the propagation of $u(x,t)$ along a constant phase line, $x-ct = \text{constant} = 0$.

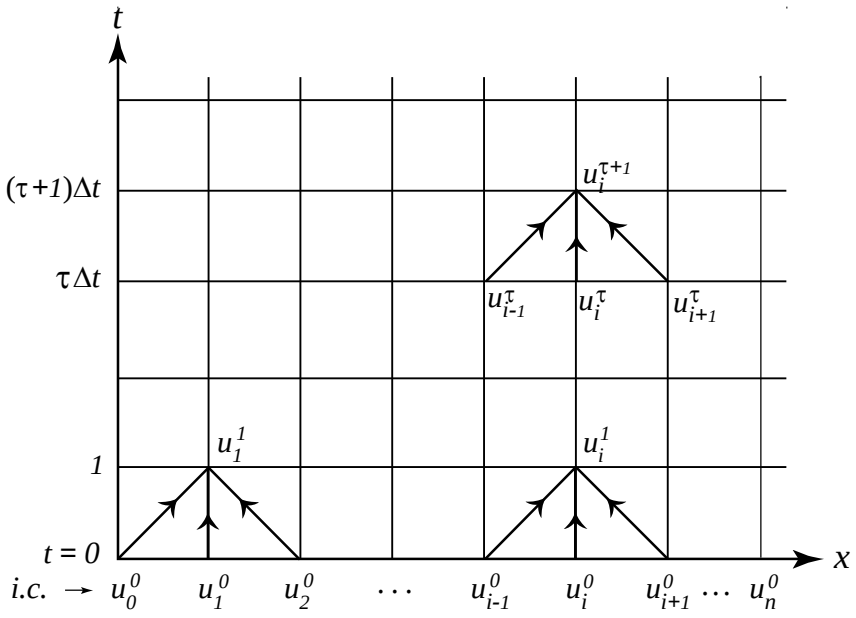

Fig. 12.4: The grid system and algorithm for the forward-in-time and centered-in-space finite difference scheme of the advection equation. The values of u at $t = 0$ are provided by the initial condition (i.c.), and the values at the left and right boundaries are determined by the boundary conditions (b.c.).

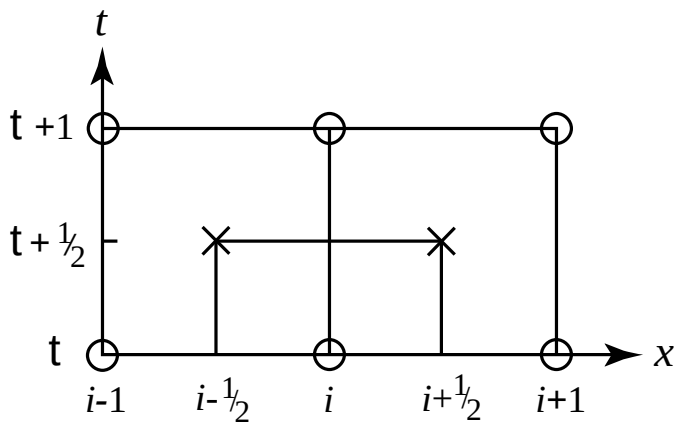

Fig. 12.5: Grid stencil for the Lax-Wendroff scheme.

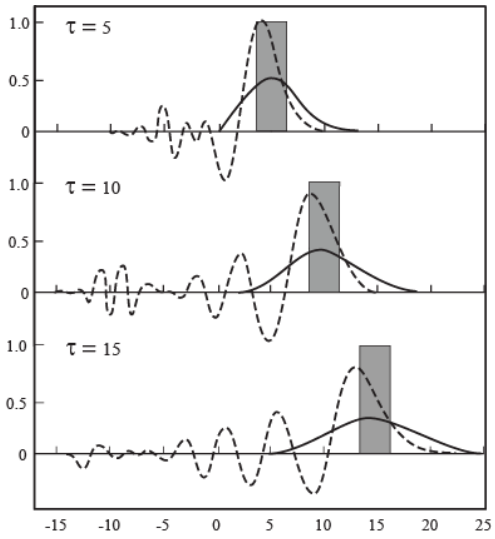

Fig. 12.6: An example of numerical damping and dispersion. Comparisons between the analytical solution (thin sold curve) and numerical solutions of applying the leapfrog in time and second-order centered in space scheme (dashed curve) and the forward in time and upstream in space scheme (heavy solid curve) to the advection equation with an initial rectangular wave centered at $x = 0$. Three nondimensional times are shown. (Adapted after Wurtele 1961)

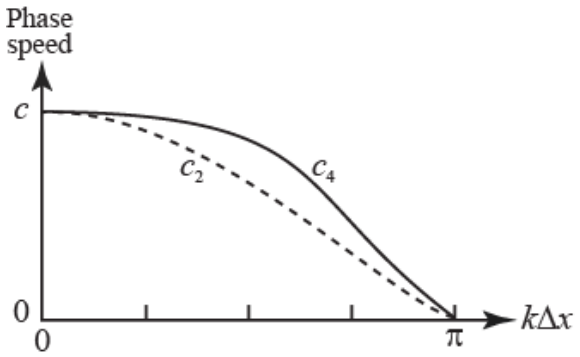

Fig. 12.7: The numerical phase speeds of the second-order (c_2) and fourth-order (c_4) centered difference schemes. The phase speed of the linear advection equation (12.3.1) is c. (Adapted after Mesinger and Arakawa 1976)

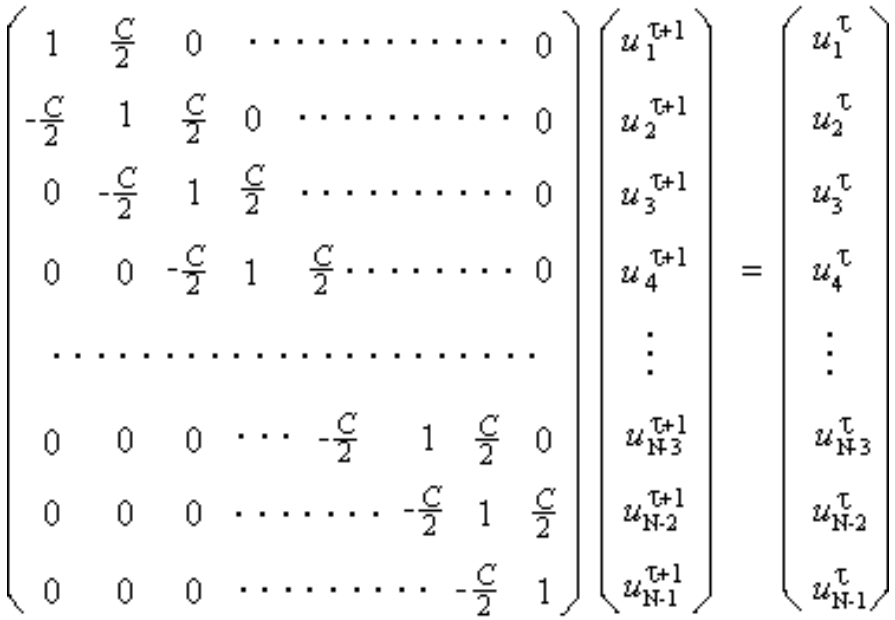

Fig. 12.8: The system of algebraic equations of (12.4.2) for the Euler implicit method with boundary conditions, $u(0,t) = u_0^{\tau+1} = 0$ and $u(N\Delta x, t) = u_N^{\tau+1} = 0$, where N is the total number of grid intervals.

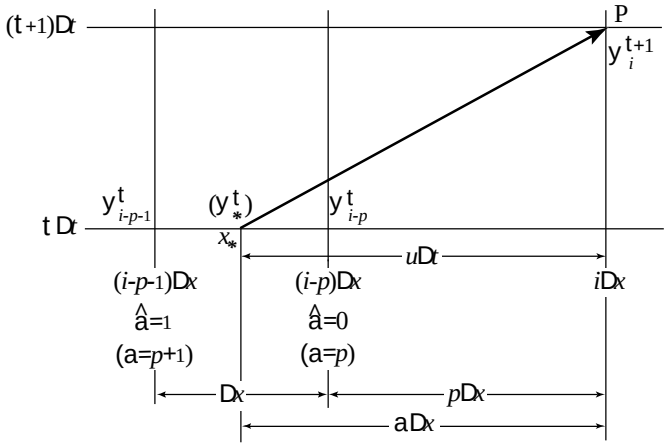

Fig. 12.9: A schematic of the semi-Lagrangian method. A fluid particle that arrives at a grid point $i\Delta x$ and at time $(\tau+1)\Delta t$ is denoted as P, which is located at x_* and at time $\tau\Delta t$. The value of the variable at this time and location (ψ_*^τ) is obtained by polynomial interpolation from the neighboring grid points, ψ_{i-p-1}^τ and ψ_{i-p}^τ, as expressed in (12.5.4) or (12.5.5).