

## 7.1 Introduction

The distribution of chemical species in the atmosphere is affected by air motions ranging from the global circulation down to the millimeter scale, at which point molecular diffusion takes over to dissipate kinetic energy. Air motions conserve the mixing ratios of the transported species since the air molecules are transported in the same way as the species. A plume of an inert chemical species transported in the atmosphere may stretch and filament, but it retains its initial mixing ratio until the filaments have become thin enough for molecular diffusion to dissipate gradients.

Representing this conservative transport in an atmospheric model is a major challenge because models cannot resolve the full range of spatial and temporal scales involved. Even if they could, chaotic behavior in solving the equation of motion would prevent a deterministic representation of the flow. Assimilation of meteorological observations can force model winds to approximate the real atmosphere, but only on the large scales of the observational network and at the cost of small-scale numerical noise introduced by the assimilation process.

From a model perspective, it is useful to distinguish between transport by the resolved large-scale winds, which can be simulated deterministically; and transport by the unresolved small-scale winds, which must be represented stochastically. The distinction between large-scale and small-scale is defined by the resolution of the model. Transport by resolved winds is commonly called *advection*, while transport by unresolved winds is called *eddy flow*, *turbulence*, or (in the vertical) *convection*. This chapter focuses on the numerical schemes used to solve the advection problem. Schemes for smaller-scale unresolved transport are presented in Chapter 8.

Numerical methods should preserve the properties of the continuous partial differential equations (PDEs) that they attempt to approximate. Desirable properties of numerical methods for advection are listed by Rasch and Williamson (1990a), Williamson (1992), and Lauritzen *et al.* (2011). They include:

1. *Accuracy*. The solution must be close to the true state.
2. *Stability*. The solution must not diverge away from the true state.
3. *Monotonicity*. The solution should not generate spurious maxima or minima in mixing ratios. Since initial conditions for mixing ratios are positive, monotonicity implies positivity of the solution.
4. *Conservation*. In the absence of sources and sinks, total mass must be conserved during advection. The algorithm should also conserve the second moment (variance) of the advected quantity.

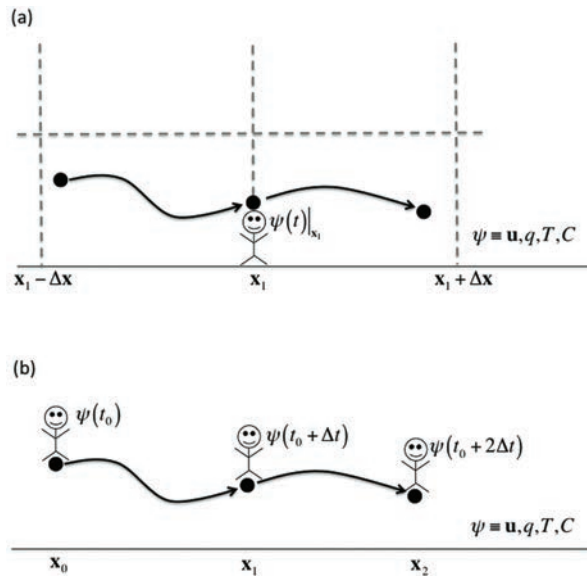


Figure 7.1

Eulerian and Lagrangian perspectives. In the Eulerian representation (a), the observer is located at fixed points (model grid points) and tracks the change in the calculated state variable  $\psi$  (e.g., mixing ratio  $C$ ) as air parcels move by. In the Lagrangian representation (b), the observer tracks the change in the variable  $\psi$  in individual air parcels as they move with the flow. Reproduced from Lin (2012).

5. *Transportivity*. Transport should be downwind only.
6. *Locality*. The solution at a given point must not be controlled by the concentrations far away from that point.
7. *Correlativity*. Relationships between species in the flow must be preserved.
8. *Flexibility*. An advection scheme is most useful if it can be implemented on different grids and at different resolutions; this makes it in particular applicable for adaptive grids.
9. *Efficiency*. A more computationally efficient algorithm facilitates simulations with higher resolution, over longer periods, and/or involving a larger number of transported species.

In Chapters 1 and 4 we drew a distinction between *Eulerian* and *Lagrangian* models for atmospheric transport (Figure 7.1). A Eulerian model solves the advection equation on a fixed reference grid, while a Lagrangian model tracks particles as they are transported in the atmosphere. Both have advantages and disadvantages. A Eulerian model provides a complete solution over the atmospheric domain with regular spatial resolution, but is subject to numerical noise and to stability constraints. A Lagrangian model has no limitations from numerical diffusion or stability, but it has uneven spatial resolution and cannot easily handle nonlinear chemistry. Eulerian and Lagrangian approaches are sometimes combined to benefit from the advantages of each, as in semi-Lagrangian advection schemes.

We focus this chapter on the basic approaches to solve the advection equation, including description of some classic schemes. The schemes used in current models

of atmospheric composition are based on these classic schemes, but often include refinements that we cannot present in detail.

Section 7.2 presents different forms of the advection equation. Section 7.3 reviews *finite difference* methods used to solve the advection equation in a Eulerian framework, while Section 7.4 focuses on *finite volume* methods. Flux-corrected methods introduced to preserve the monotonicity of the solution are discussed in Section 7.5. Selected advanced *Eulerian* numerical methods are presented in Section 7.6. Sections 7.7 and 7.8 describe the methods used in *Lagrangian* and *semi-Lagrangian* models.

## 7.2 The Advection Equation

An atmospheric species transported in a model is commonly called a *tracer*. The atmospheric advection of a tracer  $i$  is determined by its local mass density  $\rho_i$  and by the wind field  $\mathbf{v}$ . The corresponding mass flux  $\mathbf{F}_i$  is the product

$$\mathbf{F}_i = \rho_i \mathbf{v} \quad (7.1)$$

As shown in Chapter 4, the local rate of change in the density due to advective transport is the divergence  $\nabla \cdot \mathbf{F}_i$  of this flux. This defines the continuity equation for an inert tracer (no local sources or sinks):

$$\frac{\partial \rho_i}{\partial t} + \nabla \cdot (\rho_i \mathbf{v}) = 0 \quad (7.2)$$

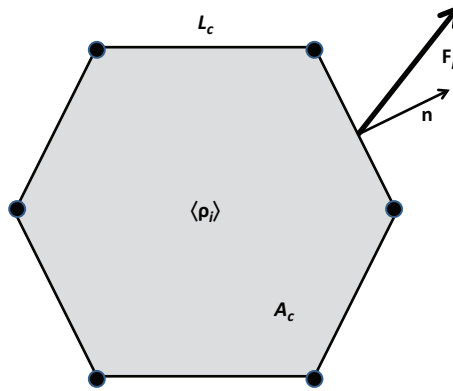
Equation (7.2) is the Eulerian flux form of the advection equation, previously discussed in Chapter 4. Transport in a Eulerian model is often expressed in terms of mass fluxes across grid cell interfaces. This can be expressed by integration over a finite volume  $V_c$  (usually a model grid cell) of the advection equation (7.2):

$$\frac{1}{V_c} \frac{\partial}{\partial t} \int_{V_c} \rho_i dV + \frac{1}{V_c} \int_{V_c} \nabla \cdot (\rho_i \mathbf{v}) dV = 0 \quad (7.3)$$

The first term in this integral equation represents the time evolution of the mean density  $\langle \rho_i \rangle$  of tracer  $i$  in the finite volume  $V_c$ . The second term can be transformed by applying the divergence (or Gauss–Ostrogradsky) theorem (see Appendix E), which states the equivalence between the volume integral over  $V_c$ , and the surface integral over the closed boundary of volume  $V_c$ . Equation (7.3) then becomes:

$$\frac{\partial \langle \rho_i \rangle}{\partial t} + \frac{1}{V_c} \int_{S_c} \mathbf{F}_i \cdot \mathbf{n} dS = 0 \quad (7.4)$$

where  $\mathbf{F}_i$  represents the flux vector of tracer  $i$  across surface  $S_c$  of the boundary of the cell and  $\mathbf{n}$  is a unit outward vector normal to the cell boundary. This expression states that the change of the mean density inside the finite cell is determined by the net flux of material in and out of the cell. Equations (7.2) and (7.4) are often called *conservative forms* of the advection equation.



**Figure 7.2** Formulation of tracer advection using (7.5) in the case of a 2-D hexagonal grid cell with surface  $A_c$  and boundaries  $L_c$ . The flux  $\mathbf{F}_i$  across a particular side of the hexagon is schematically represented. Unit vector  $\mathbf{n}$  is normal (outward) to the cell boundary and  $\langle \rho_i \rangle$  is the average density of tracer  $i$  in the cell.

When applied to a 2-D (e.g., horizontal) model, (7.4) becomes

$$\frac{\partial \langle \rho_i \rangle}{\partial t} + \frac{1}{A_c} \int_{L_c} \mathbf{F}_i \cdot \mathbf{n} \, dl = 0 \quad (7.5)$$

where the integral is now calculated along the boundary line  $L_c$  of the 2-D cell with area  $A_c$ . Figure 7.2 illustrates the different elements needed to solve (7.5) in the case of a 2-D hexagonal grid cell. Finally, in the case of a 1-D problem (along direction  $x$ ), the expression becomes

$$\frac{\partial \langle \rho_i \rangle}{\partial t} + \frac{1}{\Delta x} (F_i^r - F_i^l) = 0 \quad (7.6)$$

where  $\Delta x$  represents the size of the 1-D grid cell.  $F_i^r$  and  $F_i^l$  are the tracer fluxes at the right and left edges of the cell (positive rightward). As stated in Chapter 4, these expressions constitute the basis for finite volume methods; they are particularly suitable when solving the equations on complex or irregular grids.

When expressed in terms of mass mixing ratio  $\mu_i = \rho_i / \rho_a$  where  $\rho_a$  is the density of air, the continuity equation becomes (advective form):

$$\frac{\partial \mu_i}{\partial t} + \mathbf{v} \cdot \nabla \mu_i = 0 \quad (7.7)$$

or

$$\frac{d\mu_i}{dt} = 0 \quad (7.8)$$

where the total derivative operator (derivative along the flow) is expressed by:

$$\frac{d}{dt} = \frac{\partial}{\partial t} + \mathbf{v} \cdot \nabla \quad (7.9)$$

These forms were previously derived in Chapter 4. Equation (7.8) specifies the invariance of the mixing ratio along flow trajectories. By contrast, tracer densities  $\rho_i$  along flow trajectories may change because air is a compressible fluid (see

example below). Equation (7.7) is the Eulerian advective form of the advection equation, while (7.8) is the Lagrangian form.

Solution of the advection equation in 3-D models involves operator splitting along individual dimensions. Thus, the 1-D advection equation is solved successively in the three dimensions to obtain the 3-D solution. We will focus our discussion in the following sections on the 1-D problem as it is most relevant for model applications. In this case, the flux-form advection equation (7.2) becomes:

$$\frac{\partial \rho_i(x, t)}{\partial t} + \frac{\partial}{\partial x} [u(x, t) \rho_i(x, t)] = 0 \quad (7.10)$$

and the advective-form equation (7.7) becomes:

$$\frac{\partial \mu_i(x, t)}{\partial t} + u(x, t) \frac{\partial \mu_i(x, t)}{\partial x} = 0 \quad (7.11)$$

where  $u(x, t)$  denotes the 1-D wind velocity. Rewriting the flux-form equation (7.10) as

$$\frac{\partial \rho_i(x, t)}{\partial t} + u(x, t) \frac{\partial \rho_i(x, t)}{\partial x} = -\rho_i(x, t) \frac{\partial u(x, t)}{\partial x} \quad (7.12)$$

shows that it is identical to the advective form (7.11), but with an additional term  $\rho_i \partial u / \partial x$  that describes the compressibility of the flow. This term acts as a source of  $\rho_i$  when  $u(x, t)$  decreases with  $x$  (compression) and as a sink when  $u(x, t)$  increases with  $x$  (expansion). If the wind velocity is constant in space and time ( $u(x, t) = c$ ), the flux form of the advection equation is simply

$$\frac{\partial \rho_i(x, t)}{\partial t} + c \frac{\partial \rho_i(x, t)}{\partial x} = 0 \quad (7.13)$$

Thus the advection equation applies identically to density and mixing ratio in an incompressible flow.

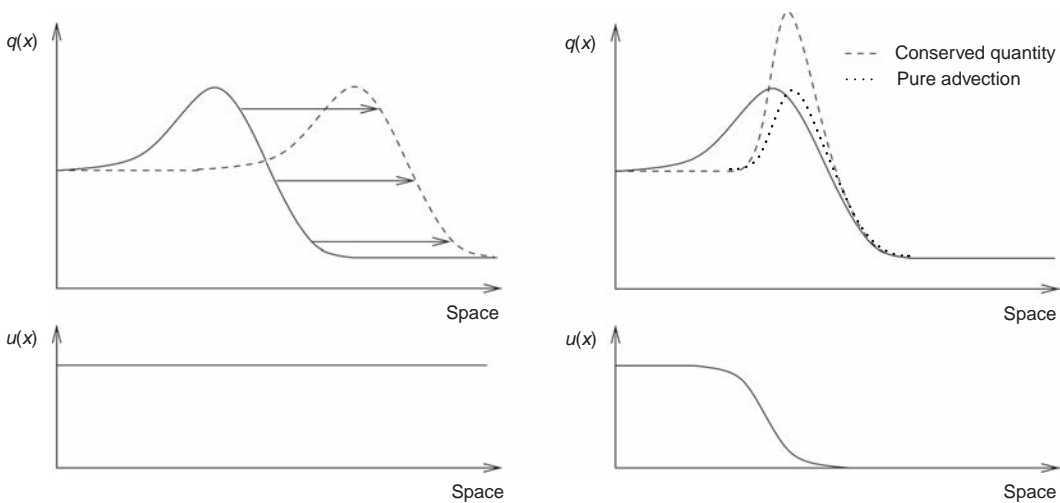
Figure 7.3 shows the solution of the 1-D advection equation for the two cases of a constant and a spatially variable wind velocity. For a constant velocity, the advection of both the density and the mixing ratio is represented by a simple translation (without deformation) of the initial function in the direction of the velocity. If the velocity decreases with space, the initial distribution of both quantities is distorted as the material is advected. The value of the maximum mixing ratio is unchanged, but the maximum value of the density is enhanced. Advection can thus modify extrema of tracer densities in a diverging flow.

In many applications, tracers are not only advected but are also diluted by turbulent diffusion. The 1-D advection–diffusion equation is given by:

$$\frac{\partial \rho_i(x, t)}{\partial t} + c \frac{\partial \rho_i(x, t)}{\partial x} - K \frac{\partial^2 \rho_i(x, t)}{\partial x^2} = 0 \quad (7.14)$$

where  $K \text{ [m}^2 \text{ s}^{-1}\text{]}$  is a diffusion coefficient. The relative importance of advection versus diffusion is measured by the dimensionless *Péclet number*  $Pe$ :

$$Pe = \frac{cL}{K} \quad (7.15)$$

**Figure 7.3**

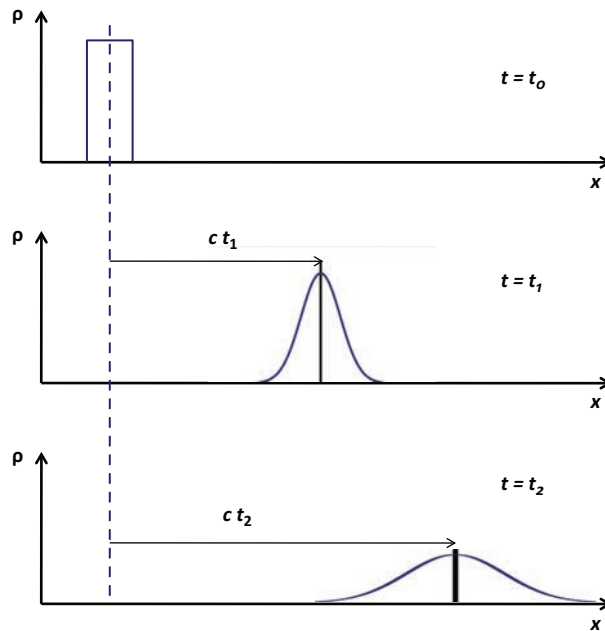
One-dimensional advection in the  $x$ -direction of a scalar function, noted here  $q(x, t)$ , for a constant velocity  $u$  (left) and a spatially varying velocity  $u(x)$  (right). The initial distribution of the function is represented by the solid line. On the left panel, this initial function (density or mixing ratio) is translated in the direction of the constant velocity  $u$ . On the right panel, the dotted curve (labeled pure advection) corresponds to a case where the term  $-q \partial u / \partial x$  is omitted and is therefore representative of the advection of a tracer mixing ratio. In this case, the spatial distribution of the function is modified, but the maximum value of the function is unchanged. The dashed curve is obtained by including the term  $-q \partial u / \partial x$  and is therefore representative of the advection of a tracer density. In this case, the area under the curve is maintained during the advection process, but the maximum value of the function is not preserved. From C. P. Dullemond with permission.

where  $L$  is a characteristic length for the advection. The Péclet number can be regarded as a measure of the ratio of the diffusive timescale to the advective timescale. For conditions typical of horizontal flow with a wind velocity  $c \sim 10 \text{ m s}^{-1}$ , a characteristic length  $L$  for long-range transport  $\sim 1000 \text{ km}$  and an eddy diffusion coefficient  $K$  of  $10^5 \text{ m}^2 \text{ s}^{-1}$ , the Péclet number is  $\sim 100$  and the transport is thus dominated by advection. For vertical transport in the boundary layer with a typical vertical velocity  $c \sim 0.01 \text{ m s}^{-1}$ , a characteristic length  $L$  of  $100 \text{ m}$ , and an eddy diffusion coefficient of  $100 \text{ m}^2 \text{ s}^{-1}$ , the Péclet number is equal to  $10^{-2}$  and diffusion becomes dominant. The solution of the transport equation for a boxcar function (shock front) subject to simultaneous advection and diffusion under a Péclet number of approximately 1 is shown schematically in Figure 7.4. Note the gradual deformation of the shape of the function under the influence of diffusion. The area under the curve, however, is conserved.

When considering a discretized form of the advection–diffusion equation, one often introduces the *numerical Péclet number* as:

$$Pe = \frac{c \Delta x}{K} \quad (7.16)$$

where  $\Delta x$  represents the grid size of the model. The numerical Péclet number measures the relative importance of advection and turbulent diffusion at the smallest spatial scale

**Figure 7.4**

Solution of the 1-D advection–diffusion equation for a Péclet number of the order of 1. The initial function at time  $t = t_0$  is represented by a boxcar function. The gradual displacement of the function is due to the advection term and deformation (spread) of the shape of the function is caused by the diffusion term. Adapted from Slingerland and Kump (2011).

resolved by the model. The advection–diffusion equation is either parabolic (diffusion-dominated) or hyperbolic (advection-dominated), depending on the Péclet number. Numerical algorithms treating atmospheric diffusion are discussed in Chapter 8.

## 7.3 Elementary Finite Difference Methods

We examine here the fundamental properties of several advection algorithms applied to the simple 1-D advection equation (along direction  $x$ ), with a constant velocity  $c$  (taken to be positive) and fixed grid spacing  $\Delta x$ . Generalization to variable wind velocity and grid spacing is presented in Section 7.3.5. As in the previous chapters dealing with numerical algorithms, we represent the field of the transported quantity by the generic mathematical symbol  $\Psi$ . The advection of a non-negative scalar function  $\Psi$  is described by the first-order hyperbolic PDE:

$$\frac{\partial \Psi}{\partial t} + c \frac{\partial \Psi}{\partial x} = 0 \quad (7.17)$$

To solve this equation, initial and boundary conditions must be specified. The initial condition can be expressed as

$$\Psi(x, 0) = G(x) \quad (7.18)$$

where  $G(x)$  represents the spatial distribution of the tracer distribution at  $t = 0$ . The resulting analytic solution is simply the translation without any change in shape of function  $G(x)$  in the  $x$ -direction at a velocity  $c$ . Thus,

$$\Psi(x, t) = G(x - ct) \quad (7.19)$$

When the spatial domain, rather than being infinite, extends from  $x = a$  to  $x = b$ , a condition must be specified at one boundary of the domain. An advection problem (hyperbolic equation) is well-posed if a boundary condition on the value of  $\Psi$  (Dirichlet condition) is imposed at the inflow boundary. If  $c > 0$ , the condition must be expressed at  $x = a$ :

$$\Psi(a, t) = H_a(t) \quad (7.20)$$

while if  $c < 0$  the condition must be applied at  $x = b$ :

$$\Psi(b, t) = H_b(t) \quad (7.21)$$

A periodic boundary condition, such as on a sphere, can be imposed as

$$\Psi(b, t) = \Psi(a, t) \quad (7.22)$$

In this case, the mass that flows out of the domain at boundary  $x = b$  flows back into the domain at boundary  $x = a$ .

Function  $\Psi(x, t)$  can be represented in an unbounded or periodic domain as a Fourier series with components (harmonics) characterized by their wavenumbers  $k$  [ $\text{m}^{-1}$ ] corresponding to wavelengths  $L = 2\pi/k$ :

$$\Psi(x, t) = \sum_{k=-\infty}^{\infty} A_k(t) e^{ikx} \quad (7.23)$$

In order to analyze the fundamental properties of different elementary algorithms, we assume that the problem is periodic in space and consider that a single harmonic ( $k$ ) of function  $\Psi(x, t)$  at grid point  $x_j = j \Delta x$  and time  $t$  takes the value

$$\Psi_k(x_j, t) = A_k(t) e^{ikj\Delta x} \quad (7.24)$$

Here,  $\Delta x$  represents the grid spacing, assumed to be uniform, and  $j$  is the grid index. The advection equation becomes:

$$\frac{\partial \Psi_k}{\partial t} + c i k \Psi_k = 0 \quad (7.25)$$

In this idealized case, the advection velocity  $c$  can be interpreted as the phase speed of the wave defined by  $\Psi_k$ . Because  $c$  is constant, the phase speed is independent of wavenumber  $k$ . Therefore all waves propagate at the same speed. Accurate numerical algorithms have to preserve this property as much as possible. In this particular case, the group speed  $c_g$ , which is indicative of the rate at which the energy propagates, is equal to  $c$  and is independent of the wavenumber  $k$ . This analytic form of the advection equation will later be compared to the forms provided by numerical analogs.



### 7.3.1 Methods Using Centered Space Differences

When adopting a uniform grid spacing  $\Delta x$ , the space derivative  $\partial\Psi/\partial x$  can be approximated by a *centered space difference* to yield the second-order accurate expression:

$$\frac{\partial\Psi}{\partial x} = \frac{\Psi_{j+1} - \Psi_{j-1}}{2\Delta x} + O(\Delta x^2) \quad (7.26)$$

where  $O(\Delta x^2)$  is the truncation error. For a wave with wavenumber  $k$ , the finite difference expression can be written as

$$\frac{\Psi_{j+1} - \Psi_{j-1}}{2\Delta x} = \frac{ik \sin(k \Delta x)}{k \Delta x} A e^{ikj\Delta x} \quad (7.27)$$

Under these conditions, the approximate form of the advection equation

$$\frac{\partial\Psi}{\partial t} + \left[ \frac{c ik \sin(k \Delta x)}{k \Delta x} \right] \Psi = 0 \quad (7.28)$$

can be compared with the exact analytic equation (7.25). When centered finite differences are used to approximate space derivatives, the phase velocity  $c^*$  associated with the numerical solution

$$c^* = c \left[ \frac{\sin(k\Delta x)}{k\Delta x} \right] \quad (7.29)$$

varies with wavenumber  $k$ , while the true phase velocity  $c$  is independent of  $k$  (see Figure 7.5). Thus, even though all wavenumber components that characterize function  $\Psi$  should move at exactly the same speed, the shorter wavelengths are trailing the longer waves. As a result, the different Fourier components of the advected function are displaced along axis  $x$  at different velocities and the numerical solution is distorted. This property that arises from the space differencing is named *numerical dispersion* and leads to phase errors. In space-centered approximations, the shortest wavelength that can be resolved by the model ( $L = 2\Delta x$  or  $k\Delta x = \pi$ ) does not move at all since its phase speed is zero (see Figure 7.5). For long waves (small values of  $k\Delta x$ ), the phase speed  $c^*$  provided by the numerical scheme approaches the true value of  $c$ . Figure 7.5 also shows the variation of the group velocity  $c_g^*$ ,

$$c_g^* = \frac{d(kc)}{dk} = c \cos(k\Delta x) \quad (7.30)$$

for the centered difference scheme. For wavelengths between  $4\Delta x$  and  $2\Delta x$ , the group velocity  $c_g^*$  is negative. This means that energy can propagate upstream, which is an undesirable property of the scheme.

When the space derivative is approximated by a fourth-order scheme over a grid with constant spacing,

$$\frac{\partial\Psi}{\partial x} = \frac{\Psi_{j-2} - 8\Psi_{j-1} + 8\Psi_{j+1} - \Psi_{j+2}}{12 \Delta x} + O(\Delta x^4) \quad (7.31)$$

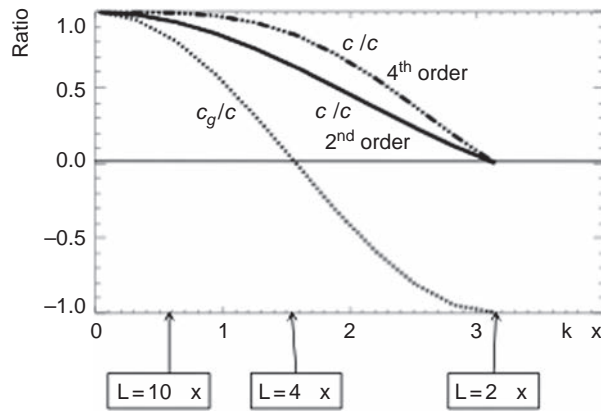


Figure 7.5

Ratios between the *phase velocity*  $c^*$  associated with the numerical solution and the true phase velocity  $c$  as a function of  $k\Delta x$  for the second-order and fourth-order space derivatives. Here  $k$  refers to the wavenumber of the different Fourier components of the signal and  $\Delta x$  to the grid spacing. The corresponding wavelengths  $L$  for three particular waves are indicated. The graph highlights the lag in the advection of the waves relative to the advective motion. This effect is more pronounced for the shortest wavelengths. The ratio between the group velocity  $c_g$  resulting from the algorithm and the true group velocity  $c$  is also shown in the case of the second-order derivative.

the phase velocity  $c^*$  becomes:

$$c^* = c \left[ \frac{4}{3} \frac{\sin(k\Delta x)}{k\Delta x} - \frac{1}{3} \frac{\sin(2k\Delta x)}{(2k\Delta x)} \right] \quad (7.32)$$

This improves over the second-order scheme, as shown in Figure 7.5, but still fails for wavelengths close to  $L = 2\Delta x$ .

### Euler Forward Scheme (FTCS)

A simple numerical method to solve (7.17) is the *Euler forward scheme*, which approximates the time derivative by a forward difference and the space derivative by a centered difference:

$$\frac{\Psi_j^{n+1} - \Psi_j^n}{\Delta t} = -c \frac{\Psi_{j+1}^n - \Psi_{j-1}^n}{2\Delta x} \quad (7.33)$$

where  $\Delta t$  is the time step and  $\Delta x$  the grid spacing (both assumed to be constant), and  $n$  and  $j$  are the indices referring to time and space, respectively, with

$$t_n = n \Delta t \quad \text{and} \quad x_j = j \Delta x$$

This algorithm, which is also referred to as the FTCS method (**f**orward-in-time, **c**entered-in-space), is first-order accurate in time and second-order accurate in space. Its solution,

$$\Psi_j^{n+1} = \Psi_j^n - \frac{\alpha}{2} (\Psi_{j+1}^n - \Psi_{j-1}^n) \quad (7.34)$$

where

$$\alpha = c \frac{\Delta t}{\Delta x} \quad (7.35)$$

is the **Courant number**, can be easily computed because the algorithm is *explicit* (the solution  $\Psi_j^{n+1}$  at time  $t_{n+1}$  for each point  $x_j$  is derived directly from quantities that are already known at time  $t_n$ ). The method is also *one-step* because only one calculation is required to advance the integration from time level  $t_n$  to the new time level  $t_{n+1}$ . It is a *two-level* scheme since only two time levels ( $t_n$  and  $t_{n+1}$ ) are involved in the calculation.

An important consideration when assessing an algorithm is its *stability*. To address this, we apply the *von Neumann's stability analysis* presented in Box 7.1. For the FTCS algorithm, the **amplification coefficient**  $g(k)$  as a function of wavenumber  $k$  is:

$$g(k) = 1 - i\alpha \sin(k\Delta x) \quad (7.36)$$

The **amplification factor**,

$$|g(k)| = [1 + \alpha^2 \sin^2(k\Delta x)]^{\frac{1}{2}} \quad (7.37)$$

is greater than one for all values of the wavenumber  $k$ . As a result, any numerical error produced by the algorithm grows exponentially with time. As highlighted in Box 7.1, the Euler FTCS algorithm is thus *unconditionally unstable* and must be rejected.

However, the FTCS method can become conditionally stable if a numerical diffusion term is added to the right-hand side of the advective equation, with diffusion coefficient  $K$ :

$$K \frac{\partial^2 \Psi}{\partial x^2} \approx \frac{K}{\Delta x^2} (\Psi_{j+1}^n - 2\Psi_j^n + \Psi_{j-1}^n) \quad (7.38)$$

In this case, the discretized equation becomes:

$$\Psi_{j+1}^n = \Psi_j^n - \frac{\alpha}{2} (\Psi_{j+1}^n - \Psi_{j-1}^n) + \beta (\Psi_{j+1}^n - 2\Psi_j^n + \Psi_{j-1}^n) \quad (7.39)$$

Here  $\alpha$  is again the Courant number, and

$$\beta = K \frac{\Delta t}{\Delta x^2} \quad (7.40)$$

is the so-called Fourier number. The amplification coefficient for wavenumber  $k$  derived from the von Neumann's analysis becomes

$$g(k) = 1 + 2\beta[\cos(k\Delta x) - 1] - i\alpha \sin(k\Delta x) \quad (7.41)$$

with the corresponding amplitude (Figure 7.6)

$$|g(k)| = [(1 - 2\beta[1 - \cos(k\Delta x)])^2 + \alpha^2 \sin^2(k\Delta x)]^{\frac{1}{2}} \quad (7.42)$$

and phase

$$\tan \Phi(k) = \frac{-\alpha \sin(k\Delta x)}{1 - 2\beta[1 - \cos(k\Delta x)]} \quad (7.43)$$

## Box 7.1

## The von Neumann Stability Analysis

The von Neumann analysis provides a methodology for assessing the stability of numerical algorithms. It applies to linear PDEs with periodic boundary conditions. We first consider the analytic solution of the advection equation by noting that any function  $\Psi$  can be expressed as the superposition of an infinite number of waves. We perform therefore a discrete Fourier transform of function  $\Psi$ , which is advected in the  $x$ -direction with a positive and constant velocity  $c$ . Consider the advection of a single wave harmonic with wavenumber  $k$ . At time  $t$ , this harmonic is expressed by:

$$\Psi_k(x, t) = a e^{ik(x-ct)}$$

After a time interval  $\Delta t$  corresponding to a displacement of the wave over a distance  $c\Delta t$ , function  $\Psi$  takes the form:

$$\Psi_k(x, t + \Delta t) = a e^{ik[x-c(t+\Delta t)]} = \Psi_k(x, t) e^{-ikc\Delta t}$$

The *amplification coefficient* or *gain*  $g(k)$  for harmonic  $k$  is the complex function defined as the ratio between function  $\Psi_k$  after and before the advection step. Thus,

$$g(k) = \frac{\Psi_k(t + \Delta t)}{\Psi_k(t)} = e^{-ikc\Delta t}$$

with a modulus (also called amplitude and here *amplification factor*)

$$|g(k)| = [g(k) \cdot g^*(k)]^{1/2} = 1$$

and a *phase*

$$\varphi(k) = -kc\Delta t$$

Here  $g^*$  is the complex conjugate of  $g$ . The amplification factor  $|g(k)|$  represents the relative change in the amplitude of the harmonic of wavenumber  $k$  after one computational time step.

These relations highlight two properties that should be reproduced as closely as possible when numerical approximations to the exact solution are sought: (1) the amplitude of all wave harmonics is unchanged during an advection process, and (2) the phase of harmonics  $k$  varies according to  $\varphi = -\alpha k \Delta x$ , where  $\alpha$  is the Courant number and  $\Delta x$  the grid spacing of the model. If, for example, the value of  $g(k)$  resulting from the use of a numerical approximation is less than 1, the amplitude of the wave is reduced by advection. Conversely, if it is larger than 1, the amplitude of the wave grows and the method becomes rapidly *unstable*. Thus, the numerical stability of an algorithm for advection requires that  $|g(k)| \leq 1$  for all waves resolved by the model. Similarly, errors on the phase cause waves of a spectrum to lag the displacement of other waves, leading to numerical dispersion.

To apply the von Neumann analysis to a numerical algorithm, let us assume that the advection equation (7.17) is approximated by the FTCS algorithm (centered difference scheme for the space derivative):

$$\Psi_j^{n+1} = \Psi_j^n - \frac{\alpha}{2} (\Psi_{j+1}^n - \Psi_{j-1}^n)$$

where  $\alpha$  is the Courant number. We apply a discrete Fourier transform and consider a single harmonic with wavenumber  $k$ :

$$\Psi_j^n(k) = e^{ikx_j}$$

It results from the above FTCS formulation that:

$$\begin{aligned}\Psi_j^{n+1} &= e^{ikx_j} - \frac{\alpha}{2} \left[ e^{ik(x_j+\Delta x)} - e^{ik(x_j-\Delta x)} \right] = e^{ikx_j} \left[ 1 - \frac{\alpha}{2} (e^{ik\Delta x} - e^{-ik\Delta x}) \right] \\ &= \Psi_j^n [1 - i\alpha \sin(k\Delta x)]\end{aligned}$$

The transfer function based on the FTCS algorithm is therefore:

$$g(k) = \frac{\Psi_j^{n+1}}{\Psi_j^n} = 1 - i\alpha \sin(k\Delta x)$$

Its amplitude is given by

$$|g(k)| = \left[ (\text{Re}[g(k)])^2 + (\text{Im}[g(k)])^2 \right]^{\frac{1}{2}} = [1 + \alpha^2 \sin^2(k\Delta x)]^{\frac{1}{2}}$$

and the phase  $\Phi(k)$  associated with this particular algorithm is derived from:

$$\tan \Phi(k) = \frac{\text{Im}[g(k)]}{\text{Re}[g(k)]} = -\alpha \sin(k\Delta x)$$

A comparison of these expressions with the values derived from the analytic solution shows that the FTCS scheme is *unconditionally unstable* since the modulus of the transfer function is larger than one for all values of wavenumber  $k$ , even for very small time steps. The phase error  $E(k)$  is given by:

$$E(k) = \Phi(k) - \phi(k) = \Phi(k) + \alpha k \Delta x$$

The von Neumann analysis is adopted in this chapter to assess the stability conditions of different numerical schemes. It is a necessary and sufficient condition for stability in the case of linear finite difference equations with constant coefficients. It is a necessary but not sufficient stability condition for nonlinear equations.

Necessary and sufficient conditions for the stability of this scheme are

$$0 \leq \beta \leq \frac{1}{2} \quad \text{and} \quad \alpha^2 \leq 2\beta \quad (7.44)$$

Thus, the FTCS method can be used if a small diffusion term is added to the advection equation and the time step is sufficiently short.

### Lax Scheme

In the **Lax method** (Lax, 1954), the term  $\Psi_j^n$  used in the FTCS scheme is replaced by  $\frac{1}{2} (\Psi_{j+1}^n + \Psi_{j-1}^n)$ , and the approximate form of the advection equation becomes:

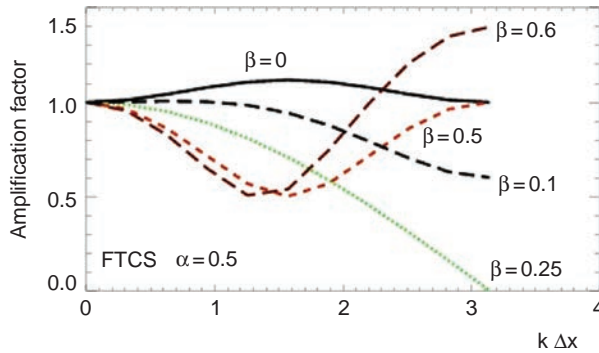


Figure 7.6

Amplification factor  $|g(k)|$  as a function of  $k\Delta x$  when the advection equation is approximated by the FTCS algorithm for a constant velocity  $c$  and the Courant number equal to 0.5. Parameter  $\beta = k\Delta t/\Delta x^2$  represents the effect of added diffusion to the advection equation. The case with  $\beta = 0$  corresponds to pure advection and is unconditionally unstable (amplification factor  $> 1$ ). When diffusion is added, the solution is stable ( $|g(k)| < 1$ ) for  $\beta < 0.5$ , but becomes unstable for larger values of  $\beta$ .

$$\Psi_j^{n+1} = \frac{1}{2} \left( \Psi_{j+1}^n + \Psi_{j-1}^n \right) - \alpha \frac{\Psi_{j+1}^n - \Psi_{j-1}^n}{2} \quad (7.45)$$

When applying the von Neumann stability analysis, we obtain the amplification coefficient:

$$g(k) = \cos(k\Delta x) - i\alpha \sin(k\Delta x) \quad (7.46)$$

whose amplitude

$$|g(k)| = [\cos^2(k\Delta x) + \alpha^2 \sin^2(k\Delta x)]^{\frac{1}{2}} \quad (7.47)$$

remains less than or equal to unity for Courant numbers  $\alpha \leq 1$ .

This stability condition  $\alpha \leq 1$ , which applies to many Eulerian schemes, is called the **Courant–Friedrichs–Lewy (CFL) criterion**. Over a time step  $\Delta t$ , the displacement of tracer should never exceed a distance larger than the grid spacing  $\Delta x$ . When a longitude–latitude grid is adopted in the model, this condition imposes severe limitations in the vicinity of the pole where the grid spacing in the longitudinal direction becomes very small. This is often circumvented by the application of numerical filters (see Section 7.10) or by the use of a reduced grid (see Section 4.7.3). The CFL condition also requires that increases in the spatial resolution of a model be accompanied by a proportional decrease in the value of the time step.

The stabilization of the solution in the Lax scheme can be understood by rearranging (7.45) as

$$\frac{\Psi_j^{n+1} - \Psi_j^n}{\Delta t} = -c \frac{\Psi_{j+1}^n - \Psi_{j-1}^n}{2\Delta x} + \frac{\Delta x^2}{2\Delta t} \left[ \frac{\Psi_{j+1}^n - 2\Psi_j^n + \Psi_{j-1}^n}{\Delta x^2} \right] \quad (7.48)$$

which is the FTCS form of equation

$$\frac{\partial \Psi}{\partial t} = -c \frac{\partial \Psi}{\partial x} + \frac{\Delta x^2}{2\Delta t} \frac{\partial^2 \Psi}{\partial x^2} \quad (7.49)$$

Thus, the Lax scheme stabilizes the FTCS solution by adding a numerical diffusion term to the advection equation with diffusion coefficient  $\Delta x^2/2\Delta t$ . Because of this numerical diffusion, a disturbance at grid point  $j$  propagates not only to downwind grid point  $(j+1)$ , but also to upwind point  $(j-1)$ . There results a transportivity error.

### Lax–Wendroff Scheme

The *Lax–Wendroff scheme* (Lax and Wendroff, 1960, 1964), also called the Leith (1965) or Crowley (1968) scheme, is designed to provide the minimum amount of added numerical diffusion required to provide stability to the FTCS solution. The value of advected function  $\Psi$  at time level  $n+1$  and at grid point  $j$  is derived from:

$$\Psi_j^{n+1} = \Psi_j^n - \alpha \left( \Psi_{j+1/2}^{n+1/2} - \Psi_{j-1/2}^{n+1/2} \right) \quad (7.50)$$

where the value of  $\Psi$  at half time level  $(n+1/2)$  and at half grid point  $(j+1/2)$  is estimated using the Lax scheme:

$$\Psi_{j+1/2}^{n+1/2} = \frac{1}{2} \left( \Psi_{j+1}^n + \Psi_j^n \right) - \frac{\alpha}{2} \left( \Psi_{j+1}^n - \Psi_j^n \right) \quad (7.51)$$

The resulting approximation to the advection equation:

$$\Psi_j^{n+1} = \Psi_j^n - \frac{\alpha}{2} \left( \Psi_{j+1}^n - \Psi_{j-1}^n \right) + \frac{\alpha^2}{2} \left( \Psi_{j+1}^n - 2\Psi_j^n + \Psi_{j-1}^n \right) \quad (7.52)$$

is second-order accurate in time even though it involves only two time levels. Here again, the third term in the right-hand side of the equation can be viewed as a diffusion term added to the FTCS scheme. In contrast to the Lax method, the effective diffusion coefficient  $c^2\Delta t/2$  is proportional to the time step  $\Delta t$ , so that numerical diffusion can be reduced by adopting smaller time steps.

The amplification coefficient is:

$$g(k) = 1 - i\alpha \sin(k\Delta x) - \alpha^2[1 - \cos(k\Delta x)] \quad (7.53)$$

with amplification factor

$$|g(k)| = \{1 - 4\alpha^2[1 - \alpha^2]\sin^4(k\Delta x/2)\}^{1/2} \quad (7.54)$$

The stability criterion is again satisfied for  $\alpha \leq 1$ . Some amplitude dampening occurs in the solution for all  $\alpha < 1$ , but it is weak for wavelengths that are large compared to the grid spacing  $\Delta x$ . For the smallest wave that can be resolved by the grid ( $L = 2\Delta x$  or equivalently  $k\Delta x = \pi$ ), the amplification coefficient is as low as zero for  $\alpha^2 = 0.5$  so the wave disappears. However, for a wave with  $L = 4\Delta x$ , the dissipation is already considerably smaller; in this case the minimum amplification coefficient is 0.8 for  $\alpha^2 = 0.5$ .

The phase  $\Phi(k)$  associated with wavenumber  $k$  (see Box 7.2) is deduced from

$$\tan \Phi(k) = \frac{-\alpha \sin(k\Delta x)}{1 - \alpha^2(1 - \cos(k\Delta x))} \quad (7.55)$$

An important consideration is the nature of the phase errors produced by second-order algorithms such as the Lax–Wendroff scheme. If one applies a Taylor

expansion to the trigonometric functions appearing in (7.55), one can show that the phase error for wavenumber  $k$  is given by

$$E(k) = -\arctan \left[ \frac{\alpha \sin(k\Delta x)}{1 - \alpha^2(1 - \cos(k\Delta x))} \right] + \alpha(k\Delta x) \approx \alpha(k\Delta x)^3 \frac{\alpha}{6} (1 - \alpha) \quad (7.56)$$

where the trigonometric functions have been approximated using Taylor expansions. The resulting lag per unit time in the displacement  $\delta x(k) = E(k)/k$  of the waves increases with the square of their wavenumber  $k$  since

$$\frac{d}{dt} \delta x(k) \approx \frac{E(k)}{k\Delta t} = c \frac{E(k)}{\alpha(k\Delta x)} \propto (k\Delta x)^2 \quad (7.57)$$

The propagation of the waves is therefore fastest for the shortest wavelengths (and thus for wavelengths that approach the grid size  $\Delta x$ ), which generates ripples in the advected signal. This type of behavior, shown here in the case of the Lax–Wendroff algorithm, is common to all second-order schemes and, as stated by *Godunov's theorem*, the monotone behavior of a numerical solution cannot be assured for linear finite difference methods with more than first-order accuracy. This theorem introduces a major limitation in the development of numerical schemes that treat advection.

### Implicit Schemes

The implicit or Euler backward-in-time, centered-in-space (BTCS)

$$\frac{\Psi_j^{n+1} - \Psi_j^n}{\Delta t} = -c \frac{\Psi_{j+1}^{n+1} - \Psi_{j-1}^{n+1}}{2\Delta x} \quad (7.58)$$

with the recursive expression

$$\Psi_j^{n+1} = \Psi_j^n - \frac{\alpha}{2} (\Psi_{j+1}^{n+1} - \Psi_{j-1}^{n+1}) \quad (7.59)$$

is first-order accurate in time and second-order in space. The amplification coefficient derived from the von Neumann analysis is:

$$g(k) = \frac{1}{1 + i\alpha \sin(k\Delta x)} \quad (7.60)$$

The resulting amplification factor

$$|g(k)| = \left( \frac{1}{1 + \alpha^2 \sin^2(k\Delta x)} \right)^{\frac{1}{2}} \quad (7.61)$$

is smaller than unity for any value of the Courant number. The method is therefore *unconditionally stable*, allowing for the adoption of any arbitrary time step  $\Delta t$ , which is a great advantage. The solution, however, cannot be retrieved as easily as in the case of explicit schemes. In the implicit case, a system of  $J$  algebraic equations (if  $J$  is the number of grid points that are not associated with boundary conditions) must be solved, which is computationally expensive. In the 1-D case, the system of equations is tridiagonal and can be solved efficiently with the Thomas algorithm (see Box 4.4). Another limitation of the method is that it has limited accuracy, with the shortest wavelengths being more rapidly attenuated than the longer wavelengths.



Accuracy of the solution can be improved by combining the FTCS and BTCS approaches. The Crank–Nicholson algorithm, written as

$$\frac{\Psi_j^{n+1} - \Psi_j^n}{\Delta t} = -c \frac{\Psi_{j+1}^{n+1} - \Psi_{j-1}^{n+1} + \Psi_{j+1}^n - \Psi_{j-1}^n}{4\Delta x} \quad (7.62)$$

is second-order accurate in time and space. It is implicit because it includes terms evaluated at time  $t_{n+1}$  on the right-hand side. The amplification coefficient is given by

$$g(k) = \frac{1 + i\alpha \sin(k\Delta x/2)}{1 - i\alpha \sin(k\Delta x/2)} \quad (7.63)$$

and the amplification factor is equal to 1 for all wavenumbers and all Courant numbers:

$$|g(k)| = 1 \quad (7.64)$$

The algorithm is thus unconditionally stable.

### Matsuno Scheme

The Matsuno scheme is a two-step explicit–implicit algorithm that is first-order accurate in time and second-order accurate in space. The first step is to predict an intermediate value  $\Psi_j^*$  of the transported quantity at time level  $n + 1$  by using a simple FTCS (Euler forward) approach:

$$\frac{\Psi_j^* - \Psi_j^n}{\Delta t} = -c \frac{\Psi_{j+1}^n - \Psi_{j-1}^n}{2\Delta x} \quad (7.65)$$

The predicted values are then substituted into the space derivative term, and a correction step is applied:

$$\frac{\Psi_j^{n+1} - \Psi_j^n}{\Delta t} = -c \frac{\Psi_{j+1}^* - \Psi_{j-1}^*}{2\Delta x} \quad (7.66)$$

By eliminating the intermediate terms  $\Psi^*$ , one derives after some manipulations:

$$\Psi_j^{n+1} = \Psi_j^n - \frac{\alpha}{2} (\Psi_{j+1}^n - \Psi_{j-1}^n) + \left(\frac{\alpha}{2}\right)^2 (\Psi_{j+2}^n - 2\Psi_j^n + \Psi_{j-2}^n) \quad (7.67)$$

This explicit expression approximates an advection equation with an additional diffusion term that approaches zero for very small  $\Delta t$ :

$$\frac{\partial \Psi}{\partial t} + c \frac{\partial \Psi}{\partial x} - \frac{c^2 \Delta t}{4} \frac{\partial^2 \Psi}{\partial x^2} = 0 \quad (7.68)$$

The amplification coefficient is:

$$g(k) = 1 - i\alpha \sin(k\Delta x) - \alpha^2 \sin^2(k\Delta x) \quad (7.69)$$

with corresponding amplification factor:

$$|g(k)| = [1 - \alpha^2 \sin^2(k\Delta x) + \alpha^4 \sin^4(k\Delta x)]^{\frac{1}{2}} \quad (7.70)$$

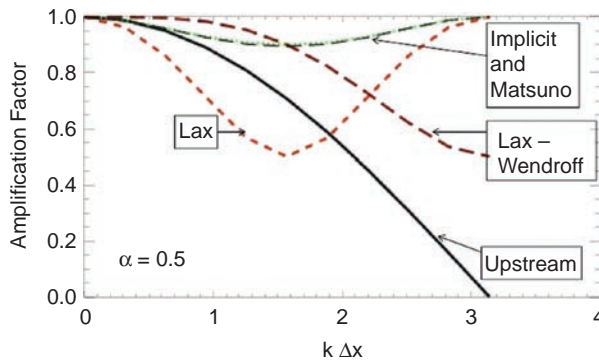
Even though this scheme bears some resemblance to implicit schemes, it is stable for the usual Courant condition ( $\alpha \leq 1$ ) of the explicit method rather than the condition that applies to implicit methods (unconditional stability).

If, in the Matsuno scheme, the correction step is replaced by

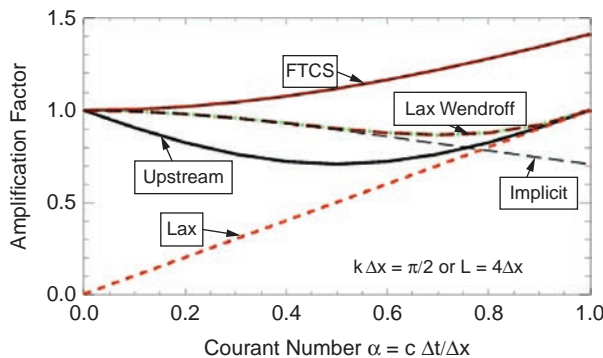
$$\frac{\Psi_j^{n+1} - \Psi_j^n}{\Delta t} = -c \frac{\Psi_{j+1}^* - \Psi_{j-1}^* + \Psi_{j+1}^n - \Psi_{j-1}^n}{4\Delta x} \quad (7.71)$$

in which the space derivative term is calculated as the average between the intermediate estimates and the estimates at time  $t_n$ , we obtain the *Heun* scheme (see Table 7.1), which is second-order accurate in space and time like the Crank–Nicholson algorithm described earlier. The method is unconditionally unstable unless a small diffusion term is artificially added to the advection equation. In this case, the scheme becomes conditionally stable. If, in the Heun scheme, the predictor step is a leapfrog algorithm (see Section 7.3.3), we obtain the method proposed by Kurihara (1965), which is second-order accurate in space and time, stable for the Courant condition, and free of numerical diffusion. Unlike the leapfrog method, it is not subject to drift, but it does not provide the exact solution for  $\alpha = 1$ .

Figures 7.7 and 7.8 show the amplification factors  $|g|$  as a function of parameters  $k \Delta x$  and  $c \Delta t / \Delta x$  for several of the algorithms described above.



**Figure 7.7** Amplification factor for different numerical methods as a function of parameter  $k \Delta x$  for a Courant number  $\alpha = 0.5$ .



**Figure 7.8** Amplification factor for different numerical methods as a function of the Courant number  $\alpha$  for parameter  $k \Delta x = \pi/2$  (corresponding to wavelength  $L = 4 \Delta x$ ).

### Fourth-Order in Space Method

The algorithms discussed so far use low-order explicit or implicit forms of the finite difference equations. These algorithms can be extended to higher-order formulations. For example, the forward-in-time implicit form of the fourth-order approximation

$$\frac{\Psi_j^{n+1} - \Psi_j^n}{\Delta t} = -c \frac{\Psi_{j-2}^{n+1} - 8\Psi_{j-1}^{n+1} + 8\Psi_{j+1}^{n+1} - \Psi_{j+2}^{n+1}}{12\Delta x} \quad (7.72)$$

is unconditionally stable. The matrix corresponding to this system is a banded matrix with five terms that can be inverted with a fast method (Press *et al.*, 2007).

### 7.3.2 Methods Using Space-Uncentered Differences

In the algorithms discussed previously, the space derivative  $\partial\Psi/\partial x$  is approximated by a second-order accurate centered difference. An alternative approach is to adopt a first-order accurate *backward-in-space finite difference*,

$$\frac{\partial\Psi}{\partial x} = \frac{\Psi_j - \Psi_{j-1}}{\Delta x} + O(\Delta x) \quad (7.73)$$

When introduced in the 1-D advection equation (7.17) together with a *forward in time derivative*, one obtains the *upstream* (or *upwind differencing*) *method* (Courant *et al.*, 1952; Godunov, 1959). Consistent with physical considerations, this algorithm provides a solution that depends on the behavior of  $\Psi$  in the direction from which the flow emanates, and not from the function downstream. Thus, for  $c > 0$ , we write a forward-in-time, backward-in-space (FTBS) expression:

$$\frac{\Psi_j^{n+1} - \Psi_j^n}{\Delta t} = -c \frac{\Psi_j^n - \Psi_{j-1}^n}{\Delta x} \quad \text{for } c > 0 \quad (7.74)$$

or equivalently:

$$\Psi_j^{n+1} = (1 - \alpha)\Psi_j^n + \alpha\Psi_{j-1}^n \quad \text{for } \alpha > 0 \quad (7.75)$$

For  $c < 0$ , the advection equation is approximated by a forward-in-time, forward-in-space (FTFS) expression:

$$\frac{\Psi_j^{n+1} - \Psi_j^n}{\Delta t} = -c \frac{\Psi_{j+1}^n - \Psi_j^n}{\Delta x} \quad \text{for } c < 0 \quad (7.76)$$

or

$$\Psi_j^{n+1} = (1 + \alpha)\Psi_j^n - \alpha\Psi_{j+1}^n \quad \text{for } \alpha < 0 \quad (7.77)$$

The amplification coefficient (for  $c > 0$ ) is:

$$g(k) = 1 - \alpha[1 - \cos(k\Delta x)] - i\alpha \sin(k\Delta x) \quad (7.78)$$

with amplitude

$$|g(k)| = (1 + 2\alpha(\alpha - 1)[1 - \cos(k\Delta x)])^{\frac{1}{2}} \quad (7.79)$$

The amplitude remains below unity as long as the Courant condition ( $\alpha \leq 1$ ) is verified. The phase  $\Phi(k)$  is given by:

$$\tan \Phi = \frac{-\alpha \sin(k\Delta x)}{1 - \alpha(1 - \cos(k\Delta x))} \quad (7.80)$$

At the stability limit, when  $\alpha = 1$ , the amplitude  $|g(k)| = 1$  and the phase  $\Phi = -k\Delta x$ . In this case, the solution provided by the upstream scheme is exact. In the general case with  $\alpha < 1$ , the solution is dampened (numerical diffusion) with the highest wave-numbers (or smallest wavelengths) more rapidly attenuated than the lower wave-numbers. This explains why the sharp corners of the square waves in Figures 7.10 and 7.12 are rounded by the upstream method.

The upstream method is monotonic and sign-preserving, but it is only first-order accurate in space and time and suffers therefore from numerical diffusion. This point can be intuitively understood by noting that the algorithm expressions (7.74) and (7.76) approximate to second-order in  $\Delta x$  and  $\Delta t$  the advection–diffusion equation:

$$\frac{\partial \Psi}{\partial t} + c \frac{\partial \Psi}{\partial x} = \frac{\partial}{\partial x} \left[ K \frac{\partial \Psi}{\partial x} \right] \quad (7.81)$$

with diffusion coefficient  $K = 0.5(c\Delta x - c^2\Delta t) = 0.5c\Delta x(1 - \alpha)$ .

Uncentered methods other than the upstream scheme have been proposed to reduce excessive numerical diffusion. For example, the approximation proposed by Warming and Beam (1976)

$$\Psi_j^{n+1} = \Psi_j^n - \alpha(\Psi_j^n - \Psi_{j-1}^n) - \frac{\alpha}{2}(1 - \alpha)(\Psi_j^n - 2\Psi_{j-1}^n + \Psi_{j-2}^n) \quad (7.82)$$

is second-order accurate in time and space and is stable for  $0 \leq \alpha \leq 2$ . It is equivalent to a Lax–Wendroff scheme in which the centered space differences are replaced by backward differences.

The Quadratic Upstream Interpolation for Convective Kinematics (QUICK) scheme of Leonard (1979) employs four points to approximate the first-order space derivative. For a constant wind velocity  $c \geq 0$  and grid spacing  $\Delta x$ , the advection equation (7.17) is first discretized as a centered-in-space, time-forward explicit scheme:

$$\Psi_j^{n+1} = \Psi_j^n - \alpha(\Psi_{j+1/2}^n - \Psi_{j-1/2}^n) \quad (7.83)$$

where the values of the advected quantity at the left ( $j - 1/2$ ) and right ( $j + 1/2$ ) edges of cell  $j$  are determined by a quadratic interpolation. One derives, for example:

$$\Psi_{j+1/2} = \frac{1}{2}[\Psi_j + \Psi_{j+1}] - \frac{1}{8}[\Psi_{j-1} - 2\Psi_j + \Psi_{j+1}] \quad (7.84)$$

so that

$$\Psi_j^{n+1} = \Psi_j^n - \frac{\alpha}{8}(\Psi_{j-2}^n - 7\Psi_{j-1}^n + 3\Psi_j^n + 3\Psi_{j+1}^n) \quad (7.85)$$

The scheme is second-order accurate in space, but it is unstable unless some dissipation is added to the advection equation. Other formulations of the QUICK

scheme (i.e., explicit, implicit, or semi-implicit approaches) are available (Chen and Falconer, 1992). A more elaborate algorithm, called QUICKEST, also proposed by Leonard (1979), is third-order accurate in time and space, and is stable for pure advection if  $\alpha \leq 1$ . In that scheme, the value of the function at the right edge is given by:

$$\Psi_{j+1/2} = \frac{1}{2} [\Psi_j + \Psi_{j+1}] - \frac{\alpha}{2} [\Psi_{j+1} - \Psi_j] - \frac{1}{8} (1 - \alpha^2) [\Psi_{j-1} - 2\Psi_j + \Psi_{j+1}] \quad (7.86)$$

The QUICK and QUICKEST schemes often generate overshoots and undershoots. They can therefore produce negative solutions. This problem is addressable by imposing flux-limiters in the integration scheme (see Section 7.5).

Finally, the algorithm proposed by Farrow and Stevens (1994), which can be regarded as an adaptation of the QUICK scheme, is expressed as a predictor-corrector integration scheme

$$\Psi_j^{n+1/2} = \Psi_j^n - \frac{\alpha}{4} (\Psi_{j+1}^n - \Psi_{j-1}^n) \quad (7.87)$$

$$\Psi_j^{n+1} = \Psi_j^n - \frac{\alpha}{2} \left[ \Psi_{j+1}^{n+1/2} - \Psi_{j-1}^{n+1/2} - \frac{1}{4} (\Psi_{j+1}^{n+1/2} - 3\Psi_j^{n+1/2} + 3\Psi_{j-1}^{n+1/2} - \Psi_{j-2}^{n+1/2}) \right] \quad (7.88)$$

It is third-order accurate in space and second-order in time. A von Neumann stability analysis indicates that it is stable for Courant numbers smaller than approximately 0.6.

### 7.3.3 Multilevel Algorithms

In the numerical schemes discussed in previous sections, the time derivatives are approximated by a two-level forward difference. We now consider methods in which information from several earlier time levels are used to calculate the value of function  $\Psi$  at time  $t_{n+1}$ .

The *regular leapfrog method* (Courant *et al.*, 1928), which is second-order accurate in time, is based on a centered-in-time and centered-in-space (CTCS) approximation of the advection equation:

$$\frac{\Psi_j^{n+1} - \Psi_j^{n-1}}{2\Delta t} = -c \frac{\Psi_{j+1}^n - \Psi_{j-1}^n}{2\Delta x} \quad (7.89)$$

or

$$\Psi_j^{n+1} = \Psi_j^{n-1} - \alpha (\Psi_{j+1}^n - \Psi_{j-1}^n) \quad (7.90)$$

In this *three-level* algorithm, the solution “leapfrogs” from time level  $(n-1)$  to time level  $(n+1)$  over the time level  $(n)$  at which the space derivative term is computed.

The von Neumann stability analysis provides a quadratic equation for the amplification coefficient, whose two roots are:

$$g(k) = \pm [1 - \alpha^2 \sin^2(k\Delta x)]^{1/2} - i \alpha \sin(k\Delta x) \quad (7.91)$$

If  $|\alpha \sin(k \Delta x)| > 1$ , the square root term is completely imaginary, and the modulus  $|g(k)|$  for one of the two roots is larger than 1, indicating instability. If  $|\alpha \sin(k \Delta x)| \leq 1$ , which is verified for all wavenumbers when  $|\alpha| \leq 1$  (CFL condition), the modulus is unity:

$$|g(k)| = \left\{ [1 - \alpha^2 \sin^2(k \Delta x)] + [\alpha \sin(k \Delta x)]^2 \right\}^{1/2} = 1 \quad (7.92)$$

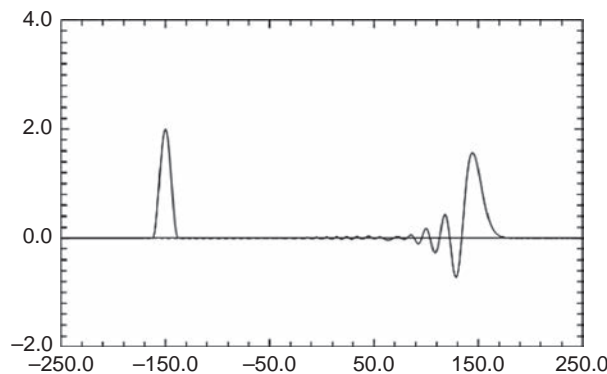
and the phase shifts for the two roots ( $\pm$ ) are respectively

$$\Phi_+ = -\sin^{-1}(\alpha \sin(k \Delta x)) \quad \text{and} \quad \Phi_- = \pi + \sin^{-1}(\alpha \sin(k \Delta x)) \quad (7.93)$$

For Courant stable conditions, the amplitude of all waves is preserved, not dissipated. This represents the major advantage of the method. When  $\alpha = 1$ , the method provides the exact solution (correct amplitude and phase). If  $\alpha < 1$ , computational dispersion occurs as phase errors, particularly for short waves, and leads to some spurious numerical oscillations.

Expression (7.91) with a  $\pm$  sign shows that the leapfrog algorithm generates two solutions with different amplification functions. One of them, called the *physical mode*, represents the meaningful solution, while the second one, referred to as the *computational mode*, is a mathematical artifact without any physical reality. This second solution propagates in the direction opposite to the flow and changes sign for every time step; it generates therefore noise that needs to be filtered with an appropriate method (see Section 4.15.4). The effect of the computational mode is visible in Figure 7.9, which shows the advection of a cosine-shaped function. Undesired oscillations with negative values of the function are produced upwind from large spatial gradients. The computational mode is most strongly excited when the initial conditions are characterized by sharp gradients.

The leapfrog algorithm tends to decouple odd and even grid points. Although, in principle, the solutions at these two types of grid point should not diverge, in practice they often do so as time progresses, causing *checkerboarding* of the solution. The problem can be addressed by adding a small dissipative term, by discarding



**Figure 7.9**

Advection by the leapfrog scheme of a cosine-shaped function with a half-width resolution of  $12\Delta x$ . The uniform grid is composed of 500 cells. The Courant number adopted in this example is 0.5. The solution is shown after 1600 time steps  $\Delta t$ . From Smolarkiewicz (2006).

the solutions at one of the two types of grid points, or by switching occasionally to an alternate advection scheme for just one time step.

Different improved leapfrog schemes have been proposed (Kim, 2003). The *upwind leapfrog scheme* introduced by Iserles (1986),

$$\frac{(\Psi_j^{n+1} - \Psi_j^n) + (\Psi_{j-1}^n - \Psi_{j-1}^{n-1})}{2\Delta t} = -c \frac{(\Psi_j^n - \Psi_{j-1}^n)}{\Delta x} \quad (7.94)$$

or

$$\Psi_j^{n+1} = \Psi_{j-1}^{n-1} + (1 - 2\alpha)(\Psi_j^n - \Psi_{j-1}^n) \quad (7.95)$$

is characterized by a considerably lower phase error than the regular leapfrog scheme. Accuracy can be increased by adopting a fourth-order accurate spatial discretization:

$$\frac{\Psi_j^{n+1} - \Psi_j^{n-1}}{2\Delta t} = -c \frac{\Psi_{j-2}^n - 8\Psi_{j-1}^n + 8\Psi_{j+1}^n - \Psi_{j+2}^n}{12\Delta x} \quad (7.96)$$

for which the von Neumann stability analysis leads to:

$$g(k) = -i \frac{\alpha}{6} [8 \sin(k\Delta x) - \sin(2k\Delta x)] \pm \left\{ 1 - \left[ \frac{\alpha}{6} [8 \sin(k\Delta x) - \sin(2k\Delta x)] \right]^2 \right\}^{1/2} \quad (7.97)$$

One derives easily that the scheme is stable if

$$\alpha < \frac{6}{8 \sin(k\Delta x) - \sin(2k\Delta x)} \quad (7.98)$$

For  $\alpha < 0.73$ , the scheme is stable for all harmonics. The use of higher orders for the calculation of the time derivative also improves the accuracy of the solution. For example, the *four-level* algorithm:

$$\Psi_j^{n+1} = \Psi_{j-1}^{n-2} + 2(1 - 3\alpha)(\Psi_j^n - \Psi_{j-1}^{n-1}) + \frac{(1 - 2\alpha)(1 - 3\alpha)}{1 + \alpha} (\Psi_{j-1}^n - \Psi_{j-1}^{n-1}) \quad (7.99)$$

is very accurate and leads to exact solutions when  $\alpha = 1/2$  or  $\alpha = 1/3$ . However, it is unstable for  $\alpha > 1/2$ .

Higher-order multi-stage methods are more accurate, but have the disadvantage of generating a larger number of computational modes. An interesting case is the third-order Adams–Bashforth scheme (see 4.197)):

$$\Psi_j^{n+1} = \Psi_j^n - \frac{\alpha}{24} \left[ 23(\Psi_{j+1}^n - \Psi_{j-1}^n) - 16(\Psi_{j+1}^{n-1} - \Psi_{j-1}^{n-1}) + 5(\Psi_{j+1}^{n-2} - \Psi_{j-1}^{n-2}) \right] \quad (7.100)$$

because the two undesired computational modes that are produced in this case are strongly damped if  $|\alpha| < 0.72$ . No filtering is required in most applications and this makes the algorithm particularly attractive, even though the solution is not positive

definite. For higher values of  $|\alpha|$ , the amplitude of one of the computational modes becomes larger than 1, and the scheme becomes unstable.

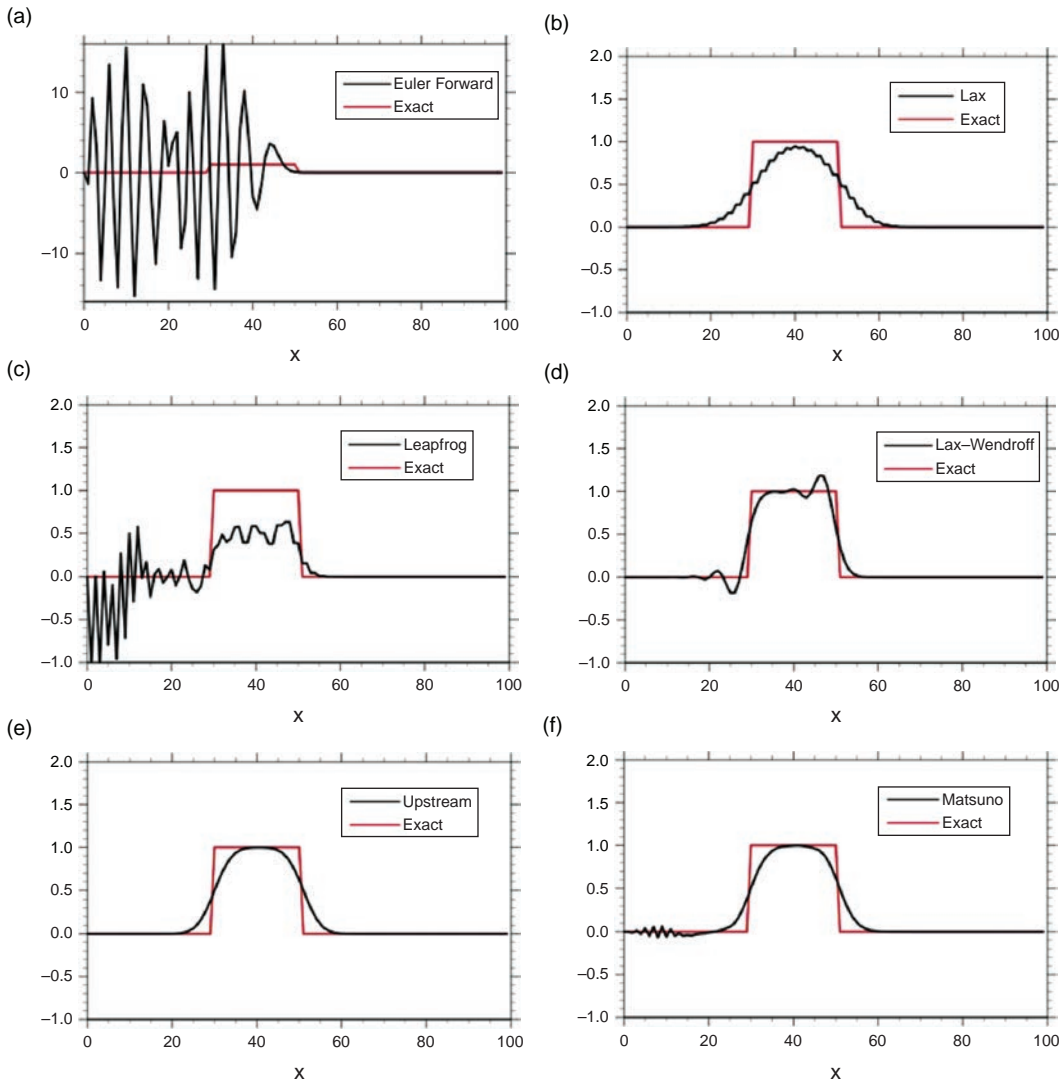
### 7.3.4 Performance of Elementary Finite Difference Algorithms

Table 7.1 summarizes the properties of the different algorithms presented previously. Figure 7.10 shows a comparison for the 1-D advection of an initial square function with constant wind speed. The instability of the Euler forward algorithm is manifested in the large oscillations. The upstream algorithm preserves sign and is free of

**Table 7.1** Elementary algorithms for solving the 1-D advection equation

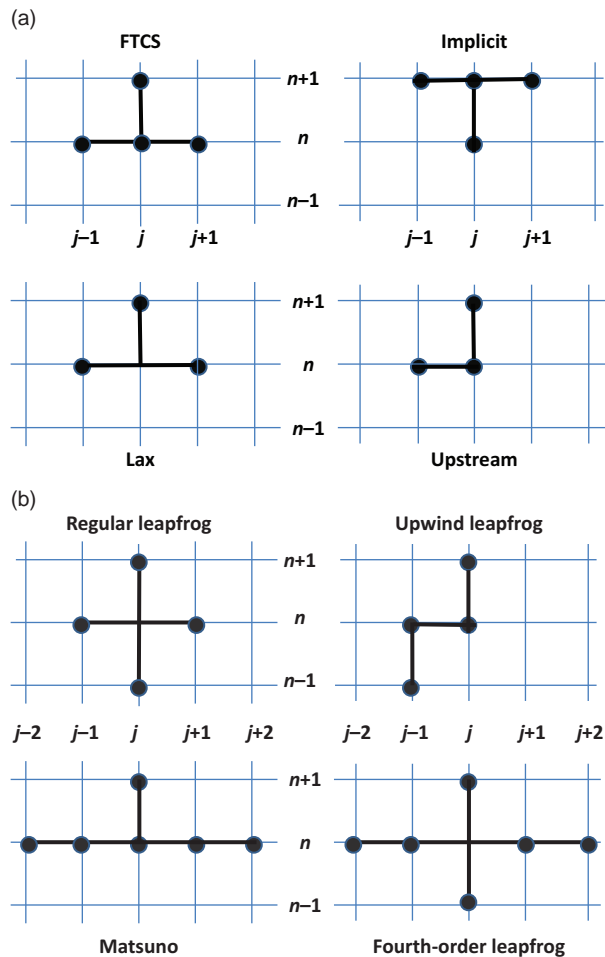
Method	Algorithm	Stability	Accuracy	Remarks
<b>Euler Forward</b>	$\Psi_j^{n+1} = \Psi_j^n - \alpha/2(\Psi_{j+1}^n - \Psi_{j-1}^n)$	Unconditionally unstable	$\Delta t, \Delta x^2$	
<b>Lax</b>	$\Psi_j^{n+1} = 1/2(\Psi_{j+1}^n + \Psi_{j-1}^n) - \alpha/2(\Psi_{j+1}^n - \Psi_{j-1}^n)$	Stable for $\alpha < 1$	$\Delta t, \Delta x^2$	Diffusive
<b>Leapfrog</b>	$\Psi_j^{n+1} = \Psi_j^{n-1} - \alpha(\Psi_{j+1}^n - \Psi_{j-1}^n)$	Stable for $\alpha < 1$	$\Delta t^2, \Delta x^2$	Dispersive
<b>Lax–Wendroff</b>	$\Psi_j^{n+1} = \Psi_j^n - \alpha/2(\Psi_{j+1}^n - \Psi_{j-1}^n) + \alpha^2/2(\Psi_{j+1}^n - 2\Psi_j^n + \Psi_{j-1}^n)$	Stable for $\alpha < 1$	$\Delta t^2, \Delta x^2$	
<b>Implicit</b>	$\Psi_j^{n+1} = \Psi_j^n - \alpha/2(\Psi_{j+1}^{n+1} - \Psi_{j-1}^{n+1})$	Unconditionally stable	$\Delta t, \Delta x^2$	
<b>Crank–Nicholson</b>	$\Psi_j^{n+1} = \Psi_j^n - \alpha/4[(\Psi_{j+1}^n - \Psi_{j-1}^n) + (\Psi_{j+1}^{n+1} - \Psi_{j-1}^{n+1})]$	Unconditionally stable	$\Delta t^2, \Delta x^2$	
<b>Matsuno</b>	$\Psi_j^{n+1} = \Psi_j^n - \alpha/2(\Psi_{j+1}^n - \Psi_{j-1}^n) + \alpha^2/4(\Psi_{j+2}^n - 2\Psi_j^n + \Psi_{j-2}^n)$	Stable for $\alpha < 1$	$\Delta t, \Delta x^2$	Diffusive
<b>Heun</b>	$\Psi_j^{n+1} = \Psi_j^n - \alpha/2(\Psi_{j+1}^n - \Psi_{j-1}^n) + \alpha^2/8(\Psi_{j+2}^n - 2\Psi_j^n + \Psi_{j-2}^n)$	Unconditionally unstable	$\Delta t^2, \Delta x^2$	
<b>Kurihara</b>	$\Psi_j^{n+1} = \Psi_j^n - \alpha/4[(\Psi_{j+1}^{n-1} - \Psi_{j-1}^{n-1}) + (\Psi_{j+1}^n - \Psi_{j-1}^n)] + \alpha^2/4(\Psi_{j+2}^n - 2\Psi_j^n + \Psi_{j-2}^n)$	Stable for $\alpha < 1$	$\Delta t^2, \Delta x^2$	Not diffusive
<b>Fourth-order (implicit)</b>	$\Psi_j^{n+1} = \Psi_j^n - \alpha/12[\Psi_{j-2}^{n+1} - 8\Psi_{j-1}^{n+1} + 8\Psi_{j+1}^{n+1} - \Psi_{j+2}^{n+1}]$	Unconditionally stable	$\Delta t, \Delta x^4$	
<b>Upstream (<math>\alpha &gt; 0</math>)</b>	$\Psi_j^{n+1} = \Psi_j^n - \alpha(\Psi_j^n - \Psi_{j-1}^n)$	Stable for $\alpha < 1$	$\Delta t, \Delta x$	Monotonic diffusive
<b>Upstream (<math>\alpha &lt; 0</math>)</b>	$\Psi_j^{n+1} = \Psi_j^n - \alpha(\Psi_{j+1}^n - \Psi_j^n)$	Stable for $\alpha < 1$	$\Delta t, \Delta x$	Monotonic diffusive





**Figure 7.10** Comparison between exact analytic (red) and numerical (black) solutions of the 1-D advection equation for a square function. The velocity  $c$  is constant. The different numerical algorithms are labeled in the panels. The original square function is centered at  $x = 20$  and the adopted Courant number is equal to 0.5. The periodic boundary condition for the advected field is zero at  $x = 0$  and  $x = 100$ . The results are shown after 40 time steps. The Euler forward algorithm is unstable (note the different scale used for the  $y$ -axis).

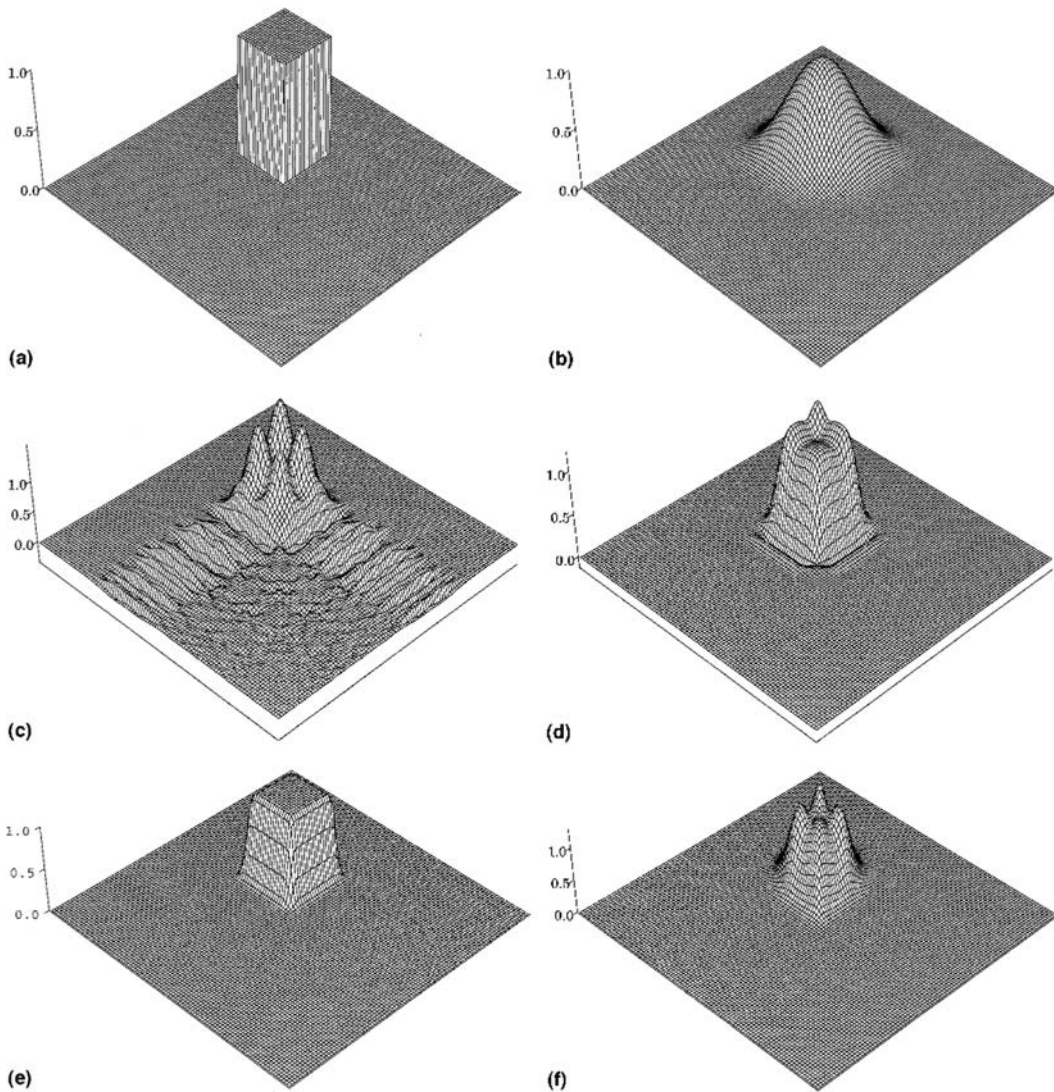
oscillations (negligible phase lag), but it is very diffusive. The Lax and Matsuno algorithms are also very diffusive. The leapfrog algorithm is not diffusive and conserves the concentration variance but it produces oscillations with undesirable negative values. The Lax–Wendroff method is slightly diffusive and produces small unwanted oscillations with negative values. Filters are generally applied to avoid unphysical negative values, but such filters may destroy the conservation properties



**Figure 7.11** Stencils describing several numerical algorithms for the approximate solution of the 1-D advection equation.

of the numerical algorithms. The stencils associated with some of the algorithms are shown in Figure 7.11. Finally, Figure 7.12 shows the performance of different algorithms in the case of the diagonal advection of a square wave in a 2-D domain. Again, one notes the strong numerical diffusion associated with the upwind scheme and the presence of large oscillations in the case of the leapfrog scheme. The multidimensional definite advection transport algorithm (MPDATA) (Smolarkiewicz, 1984; see Section 7.6) provides positive definite solutions, but with large overshoots. The Lax–Wendroff scheme with flux limiters (see Section 7.5) performs best and the QUICKEST algorithms exhibit significant oscillations (Gross *et al.*, 1999).

In summary, first-order methods such as the upstream algorithm are characterized by numerical diffusion in the solution and, as a result, tend to reduce the amplitude of peaks and to smooth spatial gradients that are present in the initial tracer distributions. High wavenumber components are also eliminated. Dispersion, which is common to the simple high-order methods described above, tends to distort the solution since it causes all waves, and specifically the small waves, to lag the true



**Figure 7.12** Two-dimensional advection of a square wave of width  $20\Delta x$  and initial concentration = 1 advected over 50 grid cells in each coordinate direction for a Courant number of 0.25. (a) Exact solution; (b) first-order upstream; (c) leapfrog; (d) QUICKEST; (e) Lax–Wendroff with flux limiters, (f) MPDATA. Reproduced from Gross *et al.* (1999) with permission from the American Society of Civil Engineers (ASCE).

solution. Numerical diffusion is generally viewed as a lesser evil because it merely causes loss of information, while dispersion generates spurious information.

### 7.3.5 Generalization to Variable Wind Speed and Grid Size

The previous discussion has highlighted some fundamental properties of different Eulerian algorithms. In practical applications, the wind in the  $x$ -direction may not be uniform ( $c$  is replaced by  $u(x, t)$ ), and the discretization interval  $\Delta x_j$  may vary

along the spatial dimension  $x$ . In that case, the space derivative in the conservative 1-D flux-form equation:

$$\frac{\partial \Psi}{\partial t} + \frac{\partial(u\Psi)}{\partial x} = 0 \quad (7.101)$$

is replaced by its second-order centered finite-difference approximation:

$$\frac{\partial(u\Psi)}{\partial x} = A_{j-1}(u\Psi)_{j-1} + B_j(u\Psi)_j + C_{j+1}(u\Psi)_{j+1} \quad (7.102)$$

where

$$A_j = \frac{\Delta x_{j+1} - 2\Delta x_j}{\Delta x_j(\Delta x_j + \Delta x_{j+1})} \quad B_j = \frac{\Delta x_{j+1} - \Delta x_j}{\Delta x_j \Delta x_{j+1}} \quad C_j = \frac{\Delta x_j}{\Delta x_j(\Delta x_j + \Delta x_{j+1})}$$

and  $\Delta x_j = x_j - x_{j-1}$ ,  $\Delta x_{j+1} = x_{j+1} - x_j$ . This center-difference scheme can be applied in the case of explicit, implicit, or Crank–Nicholson algorithms.

### 7.3.6 Mass Conservation

If we integrate the 1-D advection equation (7.17) over the spatial interval  $[A, B]$  and between time levels  $t_n$  and  $t_{n+1}$ , we find the conservation expression

$$\left[ \int_A^B \Psi dx \right]^{n+1} = \left[ \int_A^B \Psi dx \right]^n + c[\Psi_A - \Psi_B] \quad (7.103)$$

where we have again assumed  $c$  to be fixed. Condition (7.103) must be met for tracer mass to be conserved with  $\Psi_A$  and  $\Psi_B$  representing boundary conditions. In the simple case of the FCTS scheme, it is easy to evaluate the left-hand side of this integral relation:

$$\Delta x \left[ \sum_{j=j_A}^{j_B} \Psi_j^{n+1} \right] = \Delta x \sum_{j=j_A}^{j_B} \left( \Psi_j^n - c \frac{\Delta t}{2\Delta x} (\Psi_{j+1}^n - \Psi_{j-1}^n) \right) \quad (7.104)$$

or

$$\Delta x \left[ \sum_{j=j_A}^{j_B} \Psi_j^{n+1} \right] = \Delta x \left[ \sum_{j=j_A}^{j_B} \Psi_j^n \right] + c\Delta t (\Psi_{j_A-1/2}^n - \Psi_{j_B+1/2}^n) \quad (7.105)$$

if

$$\Psi_{j_A-1/2}^n = (\Psi_{j_A-1}^n + \Psi_{j_A}^n)/2 \quad \text{and} \quad \Psi_{j_B+1/2}^n = (\Psi_{j_B+1}^n + \Psi_{j_B}^n)/2$$

Thus, under the conditions adopted here, the accumulation of the conservative tracer  $\Psi$  in the domain  $[A, B]$  is proportional to the net flux ( $c\Psi$ ) at the boundaries  $A$  and  $B$ . The finite-difference analog has preserved the integral expressed by the continuum equation (7.103). If the fluxes across the external boundaries are zero or if the domain is periodic with  $\Psi_A = \Psi_B$ , mass is fully conserved in the domain. If the constant velocity  $c$  is replaced by a velocity  $u(x, t)$  that varies with space and time,

tracer conservation will be generally obtained if one considers the finite difference analog of the flux-form equation (7.2) but not its advective form (7.7).

### 7.3.7 Multidimensional Cases

The 1-D advection problem can be generalized to multiple spatial dimensions. In a 2-D Cartesian space  $(x, y)$ , the flux-conservative form of the advection equation is expressed by

$$\frac{\partial \Psi}{\partial t} + \frac{\partial(u\Psi)}{\partial x} + \frac{\partial(v\Psi)}{\partial y} = 0 \quad (7.106)$$

where  $u$  and  $v$  are the velocities of the wind components in the  $x$  and  $y$  directions, respectively. If we assume constant grid spacing, the discretization leads to the following expression

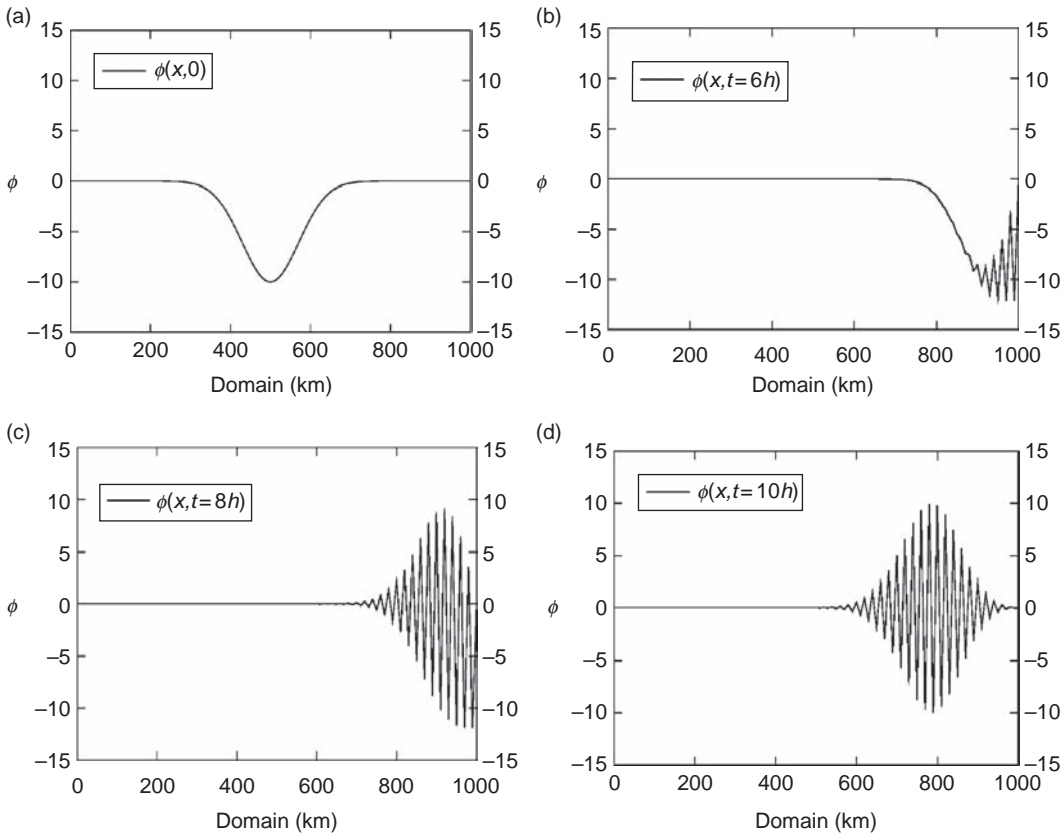
$$\frac{\Psi_{i,j}^{n+1} - \Psi_{i,j}^n}{\Delta t} + \frac{(u\Psi)_{i+1,j} - (u\Psi)_{i-1,j}}{2\Delta x} + \frac{(v\Psi)_{i,j+1} - (v\Psi)_{i,j-1}}{2\Delta y} = 0 \quad (7.107)$$

where  $\Delta x$  and  $\Delta y$  are the grid spacings in the  $x$  and  $y$  directions, and the indices  $i$  and  $j$  refer to these two directions respectively. If the differences in space are estimated at time  $t_n$  (explicit method), the algorithm is unconditionally unstable, but it can be made stable by adding numerical diffusion as in the 1-D case. If estimated at time  $t_{n+1}$  (implicit method), the algorithm is unconditionally stable. Expression (7.107) can be extended to three dimensions and solved for variable grid spacing. A Crank–Nicholson form of (7.107) can also be easily derived. The matrix of the system becomes rapidly very large and is not banded as in the 1-D case. Three-dimensional advection in practical applications is usually performed by operator splitting, with successive numerical solution of the 1-D advection equation over each dimension for individual time steps.

### 7.3.8 Boundary Conditions

The resolution of hyperbolic equations such as the advection equation applied to a limited spatial domain requires that a condition be imposed at the boundary through which material flows into the domain. In the 1-D case with a domain  $[a, b]$ , the condition must be specified at point  $x = a$  if the velocity is positive ( $c > 0$ ) and at  $x = b$  in the opposite situation ( $c < 0$ ). In some cases, the numerical algorithm requires that an additional condition be provided at the outflow boundary. By applying such a condition without precaution, the problem becomes ill-posed, and the algorithm may provide unstable solutions. This is the case when centered differences are used to represent space derivatives. Consider a 1-D flow on a domain  $[a, b]$  with a positive constant velocity  $c$  and an inflow boundary condition  $\Psi(a, t) = H(t)$  at location  $x = a$ . We approximate the spatial derivative by the leapfrog (CTCS) scheme ( $j = 1, J$ ):

$$\Psi_j^{n+1} = \Psi_j^{n-1} - \alpha(\Psi_{j+1}^n - \Psi_{j-1}^n) \quad (7.108)$$



**Figure 7.13** Advection of a bell-shaped function (arbitrary units) by a leapfrog scheme over a domain of 1000 km with a Courant number of 0.1 ( $c = 20 \text{ m s}^{-1}$ ,  $\Delta t = 50 \text{ s}$ ,  $\Delta x = 10 \text{ km}$ ) and boundary conditions of zero at the edges of the domain. Initial condition (a), solution after six hours (b), after eight hours (c) and after ten hours (d). From P. Termonia, Royal Meteorological Institute of Belgium.

At grid point  $j = 1$ , the solution is easily calculated:

$$\Psi_1^{n+1} = \Psi_1^{n-1} - \alpha(\Psi_2^n - \Psi_0^n) \quad (7.109)$$

since  $\Psi_0^n = H(t_n)$  is specified. The calculation of the solution at grid point  $j = J$ ,

$$\Psi_J^{n+1} = \Psi_J^{n-1} - \alpha(\Psi_{J+1}^n - \Psi_{J-1}^n) \quad (7.110)$$

is not straightforward because, in a well-posed problem, no value should be imposed at point  $J + 1$ . If a value is nevertheless imposed at this outflow boundary, e.g.,  $\Psi_{J+1}^n = \Psi_J^n$  or  $\Psi_{J+1}^n = 0$ , the scheme will produce unrealistic (unphysical) reflections that propagate upstream. An illustration is provided by Figure 7.13 that shows the advection of a bell-shaped function by a leapfrog scheme with  $\Psi = 0$  at both limits of the domain. As the signal reaches the downwind boundary, spurious wave reflections (saw-toothed artifacts) are produced and propagate upstream. This is generalizable to multidimensional problems: Spurious reflections often occur at the lateral boundaries of a limited-domain nested model driven by boundary conditions from a larger-domain model. Even if the imposed boundary conditions verify the analytic solution of the advection equation, some reflections are to be expected since the numerical solution is slightly different.



In practical atmospheric applications, the sign of the wind velocity and hence the direction of the flow at the boundary of the domain frequently change as the model simulation proceeds. It is therefore difficult to identify the boundary at which a condition should be specified. This problem is usually addressed by imposing some conditions along the entire boundary of the model domain (ill-posed condition), while adding in the equations a relaxation term that damps the high-frequency signal produced at the downwind boundaries. In this case, the original advection equation is modified as (Davies, 1983):

$$\frac{\partial \Psi}{\partial t} + c \frac{\partial \Psi}{\partial x} = -\lambda(x) (\Psi - \tilde{\Psi}) \quad (7.111)$$

where the relaxation coefficient  $\lambda(x)$  is different from zero only in the boundary zones (a few grid cells near the inflow and outflow boundaries, called *buffer zones*) and  $\tilde{\Psi}$  is an externally specified field chosen to be close to the expected solution. The value of  $\lambda(x)$  and the width of the relaxation zone (typically  $2\Delta x$  to  $6\Delta x$ ) need to be optimized to avoid the reflection of waves while minimizing perturbation to the solution. To ensure stability, the relaxation term should be estimated at time  $t_{n+1}$ .

An alternative damping scheme is to add a diffusion term to the advection equation:

$$\frac{\partial \Psi}{\partial t} + c \frac{\partial \Psi}{\partial x} = \frac{\partial}{\partial x} \left( K(x) \frac{\partial \Psi}{\partial x} \right) \quad (7.112)$$

where the diffusion coefficient  $K(x)$  is non-zero only near the boundary zones. Davies (1983) discusses the stability conditions for this approach.

## 7.4 Elementary Finite Volume Methods

In *finite volume* approaches, rather than considering the values of function  $\Psi$  at specified points of a model grid, one calculates the average of this function over defined grid cells. The grid points are now viewed as the centers of grid cells, often called gridboxes. The cell boundaries are called cell *edges*, *walls*, or *interfaces*.

### 7.4.1 One-Dimensional Formulation

In the 1-D problem ( $x$ -direction), the location of the cell center is noted  $x_j$ , while the locations of the cell interfaces are noted  $x_{j-1/2}$  (left side) and  $x_{j+1/2}$  (right side). For each grid cell ( $j$ ) (whose size is assumed here to be constant and equal to  $\Delta x$ ):

$$x_{j+1/2} = \frac{1}{2} (x_j + x_{j+1}) \quad (7.113)$$

except at the left ( $j = 1$ ) and right ( $j = N$ ) boundaries of the model, where we adopt

$$x_{1/2} = x_1 - \frac{(x_2 - x_1)}{2} \quad x_{N+1/2} = x_N + \frac{(x_N - x_{N-1})}{2} \quad (7.114)$$

The average value  $\Psi_j$  of the variable distribution  $\psi(x, t)$  inside the cell is

$$\Psi_j = \frac{1}{\Delta x} \int_{x_{j-1/2}}^{x_{j+1/2}} \psi(x, t) dx \quad (7.115)$$

If  $\psi(x, t)$  is a tracer concentration, then  $\Psi_j$  is the mean tracer concentration and  $\Psi_j \Delta x$  the tracer mass in grid cell ( $j$ ).

The exact (analytic) solution of the 1-D advection equation with fixed wind velocity  $c$ , when integrated over a time period  $\Delta t$ , is

$$\psi(x, t + \Delta t) = \psi(x - c\Delta t, t) \quad (7.116)$$

or, when the integral form is considered instead,

$$\frac{1}{\Delta x} \int_{x_{j-1/2}}^{x_{j+1/2}} \psi(x, t + \Delta t) dx = \frac{1}{\Delta x} \int_{x_{j-1/2}}^{x_{j+1/2}} \psi(x - c\Delta t, t) dx \quad (7.117)$$

Recognizing that the first term in (7.117) is equal to  $\Psi_j^{n+1}$ , and defining  $x' = x - c\Delta t$ , one can write:

$$\Psi_j^{n+1} = \frac{1}{\Delta x} \int_{x_{j-1/2}-c\Delta t}^{x_{j+1/2}-c\Delta t} \psi(x', t) dx' \quad (7.118)$$

Splitting this integral into different contributing parts, one writes equivalently:

$$\Psi_j^{n+1} = \frac{1}{\Delta x} \int_{x_{j-1/2}}^{x_{j+1/2}} \psi(x', t) dx' + \frac{1}{\Delta x} \int_{x_{j-1/2}-c\Delta t}^{x_{j-1/2}} \psi(x', t) dx' - \frac{1}{\Delta x} \int_{x_{j+1/2}-c\Delta t}^{x_{j+1/2}} \psi(x', t) dx' \quad (7.119)$$

or

$$\Psi_j^{n+1} = \Psi_j^n + \frac{1}{\Delta x} \int_{x_{j-1/2}-c\Delta t}^{x_{j-1/2}} \psi(x', t) dx' - \frac{1}{\Delta x} \int_{x_{j+1/2}-c\Delta t}^{x_{j+1/2}} \psi(x', t) dx' \quad (7.120)$$

The mean tracer concentration in grid cell  $j$  at time  $t_{n+1}$  is thus obtained by adding the mean value of the tracer concentration that enters grid cell ( $j$ ) to the existing mean value in that cell at time  $t_n$  and removing the mean value that is transported downstream from cell ( $j$ ) to cell ( $j + 1$ ). In this expression, the mass leaving the upwind donor cell equals the mass entering the neighboring downwind receptor cell. The finite volume method is therefore perfectly mass-conserving, which is its main advantage.

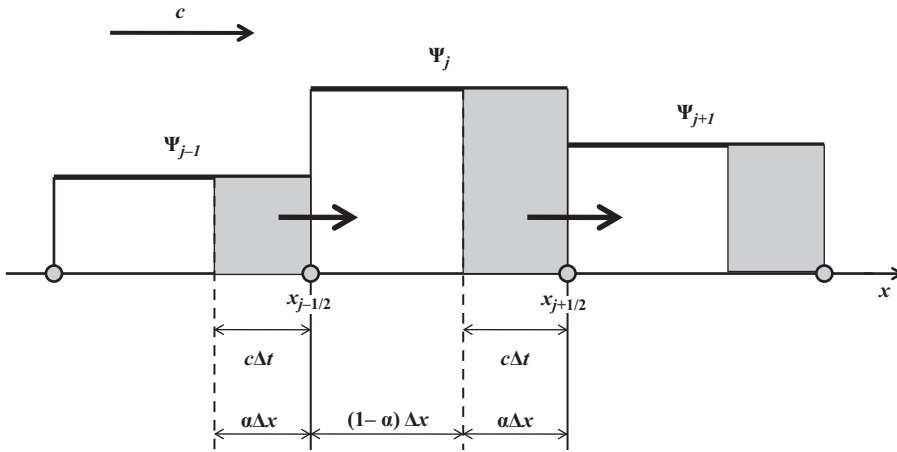
If  $F_{j-1/2}^{n+1/2}$  and  $F_{j+1/2}^{n+1/2}$  represent the mean fluxes through the left and right interfaces of grid cell  $j$ , respectively, averaged over time  $\Delta t = t_{n+1} - t_n$ , we write equivalently to (7.120):

$$\Psi_j^{n+1} = \Psi_j^n - \frac{\Delta t}{\Delta x} (F_{j+1/2}^{n+1/2} - F_{j-1/2}^{n+1/2}) \quad (7.121)$$

For the period  $\Delta t$  during which the subgrid function  $\psi(x)$  is assumed to remain unchanged, donor cell ( $j - 1$ ) transfers a mass to receptor cell ( $j$ ):

$$F_{j-1/2}^{n+1/2} \Delta t = \int_{x_{j-1/2}-c\Delta t}^{x_{j-1/2}} \psi(x) dx \quad (7.122)$$





**Figure 7.14** Representation of the *donor-cell scheme* in one dimension  $x$ . During time step  $\Delta t$ , the shaded area in cell  $j$  is transported in the  $x$ -direction to cell  $j + 1$ .

where the interval  $[x_{j-1/2} - c\Delta t, x_{j-1/2}]$  corresponds to the shaded area in cell  $(j - 1)$  (Figure 7.14). Similarly, the mass transferred from donor grid cell  $j$  to receptor grid cell  $(j + 1)$  is

$$F_{j+1/2}^{n+1/2} \Delta t = \int_{x_{j+1/2} - c\Delta t}^{x_{j+1/2}} \psi(x) dx \quad (7.123)$$

Different implementations of the finite volume method (i.e., different assumptions for the subgrid distribution of  $\psi(x)$  inside each cell) lead to different estimates of the fluxes at the interfaces of the grid cells. The simplest assumption is that  $\psi(x)$  is uniform inside each cell, so that the state  $\Psi_j$  at the grid center is identical to the state everywhere in the grid. This is the *donor cell* method. If the subgrid function  $\psi(x)$  is assumed to vary linearly with position  $x$  inside each grid cell, the algorithm is called *piecewise linear*. If  $\psi(x)$  is a second-order polynomial the algorithm is called *quadratic* or *piecewise parabolic*; see examples in Figure 7.15.

### Donor-cell algorithm

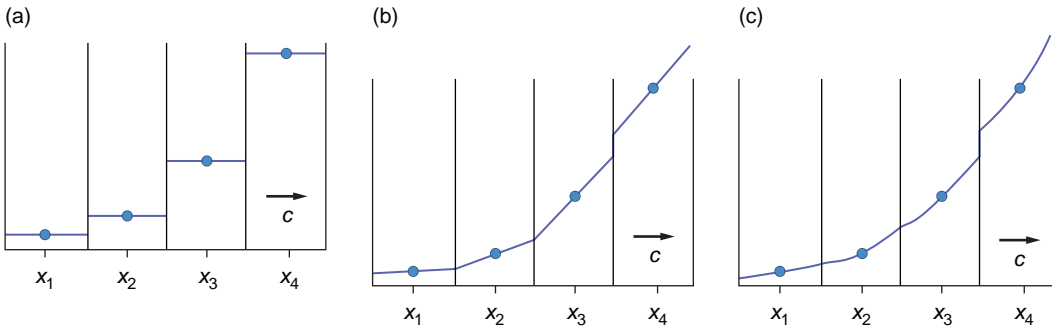
In the simple donor-cell scheme with  $c > 0$ , the subgrid function  $\psi(x)$  is uniform inside each cell, and the mass  $F_{j-1/2} \Delta t$  advected from cell  $(j - 1)$  to cell  $(j)$  over a time period  $\Delta t$  is equal to  $\Psi_{j-1} c \Delta t = \Psi_{j-1} \alpha \Delta x$ . Simultaneously, a mass equal to  $F_{j+1/2} \Delta t = \Psi_j c \Delta t = \Psi_j \alpha \Delta x$  is displaced from cell  $(j)$  to cell  $(j + 1)$ . From (7.121), we find:

$$\Psi_j^{n+1} = \Psi_j^n - \alpha (\Psi_j^n - \Psi_{j-1}^n) = \Psi_{j-1}^n \alpha + \Psi_j^n (1 - \alpha) \quad (7.124)$$

Similarly, if  $c < 0$ , we have

$$\Psi_j^{n+1} = \Psi_j^n - \alpha (\Psi_{j+1}^n - \Psi_j^n) = \Psi_j^n (1 - \alpha) - \Psi_{j+1}^n \alpha \quad (7.125)$$

As shown by Figure 7.14 displayed for  $c > 0$ , term  $\Psi_{j-1}^n \alpha \Delta x$  accounts for the mass transferred from the donor grid cell during the time step  $\Delta t$ , while term  $\Psi_j^n (1 - \alpha) \Delta x$



**Figure 7.15** Representation of the spatial distribution of a tracer within four grid cells: zeroth-, first-, and second-order polynomials.

represents the mass that remains in cell  $j$  during this time period. The addition of these two terms represents the resulting mass in gridbox ( $j$ ) at time  $t_{n+1}$ . This expression, derived in the simple case where  $\psi(x)$  is assumed uniform, is identical to the upstream formula (7.74). The method is first-order accurate and is therefore characterized by large numerical diffusion.

### Piecewise linear algorithm

In the case of the piecewise linear approach, we write for  $x_{j-1/2} < x < x_{j+1/2}$ :

$$\psi(x) = \Psi_j + b_j(x - x_j) \quad (7.126)$$

where  $\Psi_j$  denotes the value of linear function  $\psi(x)$  at the center of the cell (also the mean value of  $\psi(x)$  in the cell), and  $b_j$  is the *slope* of the function inside the cell. For a fixed velocity  $c > 0$  and an equally spaced grid, the flux at the left cell interface is:

$$F_{j-1/2}^{n+1/2} = \frac{c}{\Delta t} \int_{t_n}^{t_{n+1}} \psi(x_{j-1/2}, t) dt = \frac{c}{\Delta t} \int_{t_n}^{t_{n+1}} \Psi_{j-1}^n + b_{j-1}^n (x_{j-1/2} - x_{j-1} - c(t - t_n)) dt \quad (7.127)$$

or

$$F_{j-1/2}^{n+1/2} = c \left[ \Psi_{j-1}^n + \frac{1}{2} b_{j-1}^n (\Delta x - c\Delta t) \right] \quad (7.128)$$

Similarly, for  $c < 0$ , one finds

$$F_{j-1/2}^{n+1/2} = c \left[ \Psi_j^n - \frac{1}{2} b_j^n (\Delta x + c\Delta t) \right] \quad (7.129)$$

From expression (7.119), one can easily deduce that the average value of subgrid function  $\psi(x)$  in cell ( $j$ ) at time  $t_{n+1}$  is given by:

$$\Psi_j^{n+1} = \Psi_j^n - \alpha (\Psi_j^n - \Psi_{j-1}^n) - \frac{\alpha}{2} (b_j^n - b_{j-1}^n) (1 - \alpha) \Delta x \quad (7.130)$$

if  $c > 0$ , and

$$\Psi_j^{n+1} = \Psi_j^n - \alpha(\Psi_{j+1}^n - \Psi_j^n) + \frac{\alpha}{2}(b_{j+1}^n - b_j^n)(1 + \alpha)\Delta x \quad (7.131)$$

if  $c < 0$ . This algorithm can be viewed as an extension of the donor-cell scheme with a correction term that disappears if the slope of function  $\psi(x)$  is equal to zero.

The value of  $b_j$  is expressed as a function of the values of function  $\Psi$  at the center of adjacent cells. Different options are: (1) centered slope or Fromm method, (2) upwind slope or Beam–Warming method, and (3) downwind slope (equivalent to the Lax–Wendroff algorithm). The values of the  $b_j$  coefficients are respectively

$$b_j = \frac{\Psi_{j+1} - \Psi_{j-1}}{2\Delta x}, \quad b_j = \frac{\Psi_j - \Psi_{j-1}}{\Delta x}, \quad b_j = \frac{\Psi_{j+1} - \Psi_j}{\Delta x}$$

The resulting algorithm for  $c > 0$  is in the case of the Fromm scheme (which is upwind biased)

$$\Psi_j^{n+1} = \Psi_j^n - \frac{\alpha}{4}(\Psi_{j+1}^n + 3\Psi_j^n - 5\Psi_{j-1}^n + \Psi_{j-2}^n) + \frac{\alpha^2}{4}(\Psi_{j+1}^n - \Psi_j^n - \Psi_{j-1}^n + \Psi_{j-2}^n) \quad (7.132)$$

In the case of the Beam–Warming scheme (also upwind biased), it is (see also expression 7.82)

$$\Psi_j^{n+1} = \Psi_j^n - \frac{\alpha}{2}(3\Psi_j^n - 4\Psi_{j-1}^n + \Psi_{j-2}^n) + \frac{\alpha^2}{2}(\Psi_j^n - 2\Psi_{j-1}^n + \Psi_{j-2}^n) \quad (7.133)$$

and, in the case of the Lax–Wendroff scheme, we find the three-point stencil (spatially centered) expression that is identical to (7.52)

$$\Psi_j^{n+1} = \Psi_j^n - \frac{\alpha}{2}(\Psi_{j+1}^n - \Psi_{j-1}^n) + \frac{\alpha^2}{2}(\Psi_{j+1}^n - 2\Psi_j^n + \Psi_{j-1}^n) \quad (7.134)$$

Figure 7.19 in Section 7.5 shows the numerical solution for advection of a step function obtained with the second-order accurate Fromm and Beam–Warming methods. The solution from the Lax–Wendroff algorithm was previously shown in Figure 7.10. In all three cases, the solution is not monotonic.

Other implementations of the finite volume method with specific subgrid distributions of function  $\psi(x)$  (e.g., the algorithms of Russell and Lerner, 1981; Colella and Woodward, 1984; Prather, 1986) are discussed in Section 7.5.

## 7.4.2 Two-Dimensional Formulation

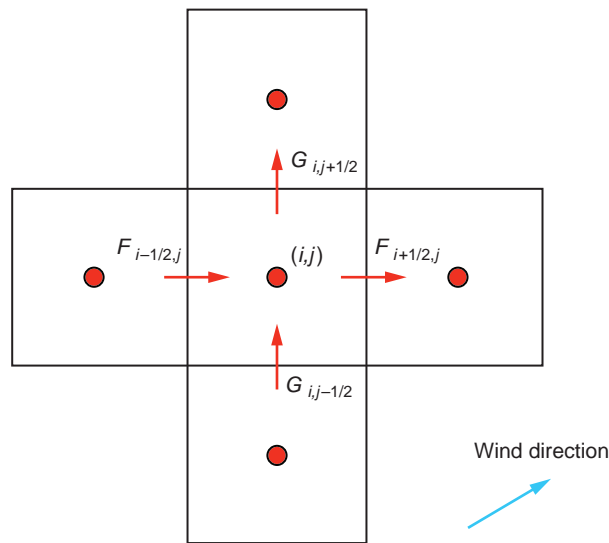
When extended to two dimensions (see Figure 7.16), the finite volume algorithm is expressed as:

$$\Psi_{i,j}^{n+1} = \Psi_{i,j}^n - \frac{\Delta t}{\Delta x}(F_{i+1/2,j}^{n+1/2} - F_{i-1/2,j}^{n+1/2}) - \frac{\Delta t}{\Delta y}(G_{i,j+1/2}^{n+1/2} - G_{i,j-1/2}^{n+1/2}) \quad (7.135)$$

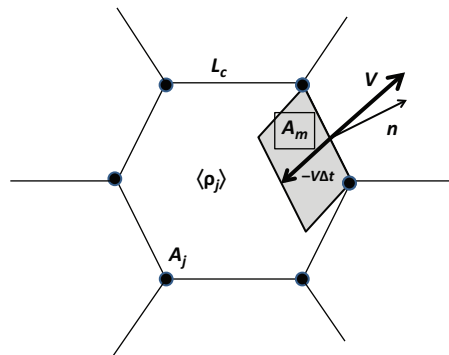
where  $F$  and  $G$  represent the mean fluxes in the  $x$ - and  $y$ -direction respectively. The challenge in defining accurate algorithms is to properly formulate the fluxes at the interfaces as a function of the dependent variables in the neighboring cells.

Another approach is to solve the discretized form (7.6) for the mean density  $\langle \rho \rangle_j$  inside grid cell  $j$  (Dukowicz and Baumgardner, 2000; Lipscomb and Ringler, 2005; Miura, 2007; Skamarock and Menchaca, 2010):

$$\langle \rho \rangle_j^{n+1} = \langle \rho \rangle_j^n - \frac{\Delta t}{A_j} \sum L_c(\mathbf{F}_j^{n+1/2} \mathbf{n}) \quad (7.136)$$



**Figure 7.16** Representation of orthogonal flux components  $F$  (in the  $x$ -direction) and  $G$  (in the  $y$ -direction) across cell interfaces in two dimensions. The flux form adopted for the algorithm ensures mass conservation.



**Figure 7.17** Schematic representation of the 2-D remapping algorithm of Miura (2007) in the case of a hexagonal cell. The shaded region represents the mass advected through the cell boundary over a time step  $\Delta t$ . Redrawn from Skamarock and Menchaca (2010).

Here,  $A_j$  is the area of the cell,  $\mathbf{F}_j$  the mass flux across the interfaces of the cell,  $\mathbf{n}$  a unit vector perpendicular to the cell boundaries, and  $L_c$  the length of each cell edge. The sum applies to all cell edges. The flux can be estimated through a remapping algorithm, as depicted in Figure 7.17 for a hexagonal cell where the fluid velocity  $\mathbf{v}$  at one point of each cell boundary (e.g., center of the cell edge) is projected backward to define the upstream flux-area  $A_m$  (shaded parallelogram). The mean density in area  $A_m$  is derived by a polynomial fit using the mean densities in the neighboring cells at time level  $t_n$ . The flux  $\mathbf{F}_j$  is then derived from the mass contained in area  $A_m$  that is displaced across the cell edge over a time interval  $\Delta t$  with velocity  $\mathbf{v}$ . The mass originating from all neighborhood cells is remapped onto cell  $j$  and provides the mean density in this cell at time level  $t_{n+1}$ . The accuracy of the scheme

depends on the order of the polynomial that is adopted. In the incremental remapping of Dukowicz and Baumgardner (2000) and of Lipscomb and Ringler (2005), all endpoints on the grid are tracked back, so that the upstream flux area is a polygon. The method bears some similarities with semi-Lagrangian schemes discussed in Section 7.8.

## 7.5 Preserving Monotonicity: Flux-Corrected Transport

As stated in Section 7.4, the solutions provided by high-order accurate algorithms are not monotonic. Preserving monotonicity in the solution of the advection equation is an important requirement for chemical transport models. The generation of new extrema or “ripples” in the vicinity of steep gradients (including shocks and discontinuities of the solution) is unacceptable in most applications. Correction techniques have therefore been proposed to eliminate these unphysical maxima or minima caused by numerical dispersion in high-order algorithms.

One-dimensional flux-corrected advection algorithms are based on the finite volume approximation equation (7.121)

$$\Psi_j^{n+1} = \Psi_j^n - \frac{\Delta t}{\Delta x} (F_{j+1/2}^{n+1/2} - F_{j-1/2}^{n+1/2})$$

in which  $F_{j-1/2}^{n+1/2}$  and  $F_{j+1/2}^{n+1/2}$  are again the flux averaged over the adopted time step at the edge of grid cell ( $j$ ). The presence of spurious oscillations in the solution is avoided by preventing the total variation ( $TV$ ) in the discrete representation of the solution

$$TV = \sum_j |\Psi_j - \Psi_{j-1}|$$

from increasing as the integration proceeds. This is accomplished by limiting the amplitude of the upstream and downstream fluxes, so that the following condition:

$$\sum_j |\Psi_j^{n+1} - \Psi_{j-1}^{n+1}| \leq \sum_j |\Psi_j^n - \Psi_{j-1}^n| \quad (7.137)$$

is fulfilled (total variation diminishing or *TVD* condition). An increase in the total variation ( $TV$ ) is a measure of the formation of oscillations in the solution.

Fluxes can be limited in the discrete form of the advection equation by specifying the fluxes at each edge of the finite volume cells [here for ( $j - 1/2$ )] as

$$F_{j-1/2} = F_{j-1/2}^L - \Phi(r_{j-1/2}) [F_{j-1/2}^L - F_{j-1/2}^H] \quad (7.138)$$

where  $F_{j-1/2}^L$  and  $F_{j-1/2}^H$  represent the fluxes calculated by a low-order and a high-order method, respectively. The flux limiter functions  $\Phi(r_{j-1/2})$  are expressed as a function of parameter  $r_{j-1/2}$  defined as

$$r_{j-1/2} = \frac{\Psi_{j-1} - \Psi_{j-2}}{\Psi_j - \Psi_{j-1}} \quad \text{for } c > 0 \quad (7.139)$$

$$r_{j-1/2} = \frac{\Psi_{j+1} - \Psi_j}{\Psi_j - \Psi_{j-1}} \quad \text{for } c < 0 \quad (7.140)$$

Similar expressions can be established for the flux at the other edge ( $j + 1/2$ ) of the cell. One can show that, to fulfill the TVD condition, the flux limiter function must be chosen such

$$\begin{aligned} r \leq \Phi(r) \leq 2r & \quad \text{for} \quad 0 \leq r \leq 1 \\ 1 \leq \Phi(r) \leq r & \quad \text{for} \quad 1 \leq r \leq 2 \\ 1 \leq \Phi(r) \leq 2 & \quad \text{for} \quad r > 2 \end{aligned}$$

In practical terms, the solution can be made monotonic by adopting for the  $\Phi(r)$  a value close to zero in the vicinity of sharp gradients (low-order method) and a value close to 1–2 (higher-order method) in regions where the solution is expected to be smooth.

To illustrate the flux limiter method, we consider the linear piecewise scheme discussed in Section 7.4. In this particular case, the flux at the left interface of a grid cell ( $j$ ) can be expressed by (7.128) and (7.129) or

$$F_{j-1/2}^{n+1/2} = c \left[ \Psi_{j-1}^n + \frac{1}{2}(1 - \alpha) b_{j-1}^n \Delta x \right] \quad \text{for } c > 0 \quad (7.141)$$

$$F_{j-1/2}^{n+1/2} = c \left[ \Psi_j^n - \frac{1}{2}(1 + \alpha) b_j^n \Delta x \right] \quad \text{for } c < 0 \quad (7.142)$$

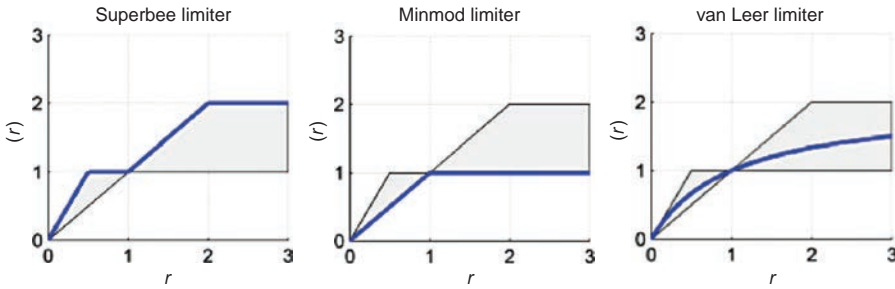
A flux limiter required to preserve monotonicity is introduced by adjusting the second term in expression (7.142). This is accomplished by replacing  $b_{j-1} \Delta x$  and  $b_j \Delta x$  in equations (7.141–7.142) by  $\Phi(r_{j-1/2}) (\Psi_j - \Psi_{j-1})$ . The concept of flux limiter is therefore similar to the concept of *slope limiter*, which is sometimes used to characterize the flux-corrected transport. The resulting “corrected flux” is therefore

$$F_{j-1/2}^{n+1/2} = c \left[ \Psi_{j-1}^n + \frac{1}{2}(1 - \alpha) \Phi(r_{j-1/2}^n) (\Psi_j^n - \Psi_{j-1}^n) \right] \quad \text{for } c > 0 \quad (7.143)$$

$$F_{j-1/2}^{n+1/2} = c \left[ \Psi_j^n - \frac{1}{2}(1 + \alpha) \Phi(r_{j-1/2}^n) (\Psi_j^n - \Psi_{j-1}^n) \right] \quad \text{for } c < 0 \quad (7.144)$$

with  $r_{j-1/2}$  defined by expressions (7.139) or (7.140), depending on the sign of the wind velocity  $c$ . By adjusting all indices in the above expression, one finds the value of the corrected flux at the right edge of cell ( $j$ ), and the solution  $\Psi_j^{n+1}$  at time  $t_{n+1}$  is derived by applying (7.121) with an appropriate choice for the limiter  $\Phi(r)$ . Note that the choice of  $\Phi(r) = 0$  and  $\Phi(r) = 1$  for all values of  $r$ , corresponds the donor cell and Lax–Wendroff algorithms, respectively. Similarly, the choice  $\Phi(r) = r$  and  $\Phi(r) = (1 + r)/2$  corresponds respectively to the Beam–Warming and the Fromm schemes discussed in Section 7.4. None of these four limiters satisfy the TVD condition, and as a result the corresponding schemes do not provide monotonic solutions.

Several formulations for flux limiters that satisfy the TVD condition and hence lead to monotonic solutions have been proposed (Roe, 1986). The following expressions



**Figure 7.18** Flux/slope limiter functions (blue curve) for the superbee, minmod and van Leer algorithms superimposed on the regions (shaded) in which the TVD condition is met. Courtesy of Graham W. Griffiths.

$$\Phi(r) = \max [0, \min (1, r)]$$

define the *minmod* method,

$$\Phi(r) = \max [0, \min (2r, 1), \min (r, 2)]$$

the *superbee* method, and

$$\Phi(r) = \frac{r + |r|}{1 + |r|}$$

the *van Leer* algorithm. Figure 7.18 shows as a function of parameter  $r$  the values of the three limiters  $\Phi(r)$ , as well as the domain in which the TVD condition is met. Figure 7.19 shows how the application of a minmod and superbee flux correction improves the solution. Several other limiters have been proposed to enforce monotonicity of the solution.

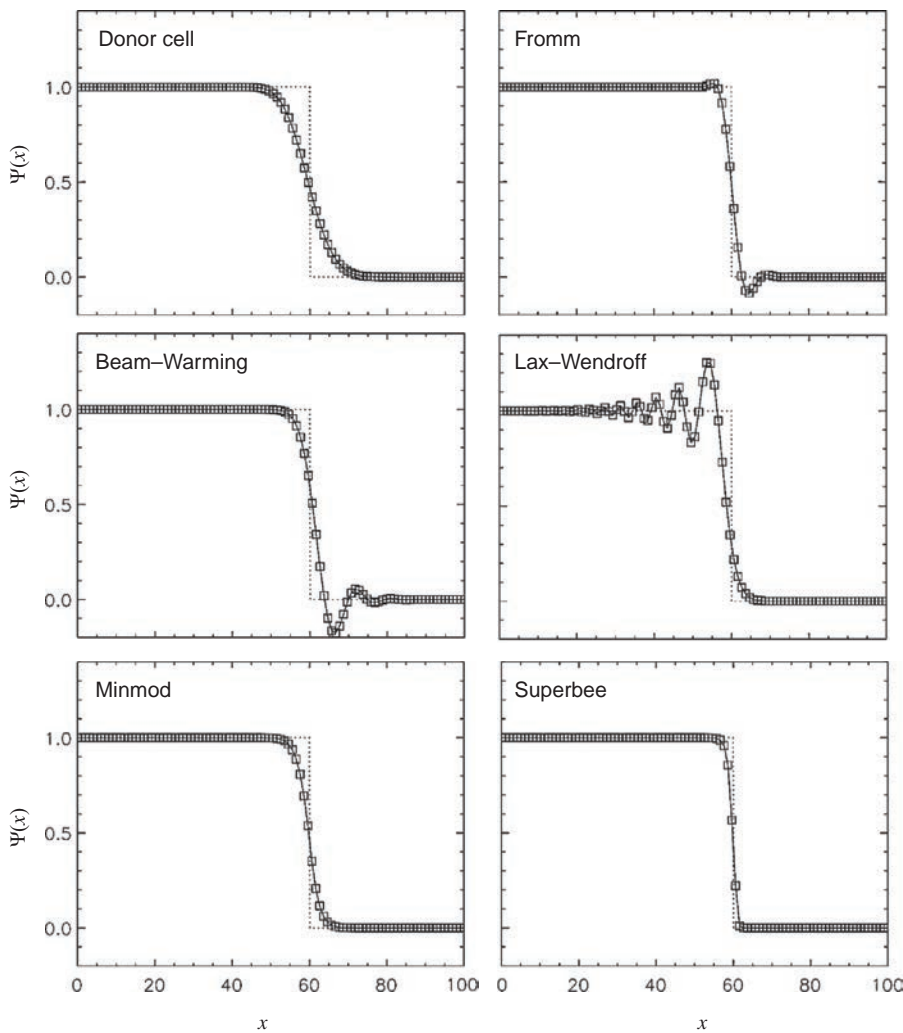
An interesting numerical method that overcomes the excessive diffusion of the upstream algorithm and provides monotonic solutions is the flux-corrected scheme developed by van Leer (1977, 1979). To describe this algorithm, we start again from the finite volume expression

$$\Psi_j^{n+1} = \Psi_j^n - \frac{\Delta t}{\Delta x} (F_{j+1/2} - F_{j-1/2}) \quad (7.145)$$

where  $\Psi_j$  applies to the center of cell  $j$  and  $F_{j+1/2}$  and  $F_{j-1/2}$  are the time-averaged flux across boundaries  $j + 1/2$  and  $j - 1/2$ , respectively. We assume here that the wind field is not uniform, and  $F_{j+1/2}$  is therefore computed as the product of the velocity  $u_{j+1/2}$  by an estimate of  $\Psi$  at the grid cell boundary. This estimate is obtained from a Taylor's series expansion on the gridded  $\Psi$  field:

$$F_{j-1/2}^{n+1/2} = u_{j-1/2}^n \left[ \Psi_{j-1}^n + \frac{1}{2} \left( 1 - u_{j-1/2}^n \frac{\Delta t}{\Delta x} \right) \Delta_{j-1}^n \Psi \right] \quad \text{for } u_{j-1/2}^n > 0 \quad (7.146)$$

$$F_{j-1/2}^{n+1/2} = u_{j-1/2}^n \left[ \Psi_j^n - \frac{1}{2} \left( 1 + u_{j-1/2}^n \frac{\Delta t}{\Delta x} \right) \Delta_j^n \Psi \right] \quad \text{for } u_{j-1/2}^n < 0 \quad (7.147)$$



**Figure 7.19** Advection of a sharp discontinuity (step function) using six different algorithms: the first-order diffusive *donor-cell* scheme; the second-order non-monotonic *Fromm* and *Beam-Warming* algorithms; and the flux-corrected *minmod* and *superbee* methods. The results are obtained after 300 time steps with  $\Delta t = 0.1$  over a grid of 100 points with a spacing  $\Delta x = 1$ . From C. P. Dullemond with permission.

where  $\Delta_j^n \Psi / \Delta x$  corresponds to the slope  $b_j$  of the subgrid function  $\psi(x)$ . Van Leer proposes for  $\Delta_j^n \Psi$  an expression that limits the flux at the edge of the grid cells (see also Allen *et al.*, 1991):

$$\Delta_j^n \Psi = 2 \frac{(\Psi_j^n - \Psi_{j-1}^n)(\Psi_{j+1}^n - \Psi_j^n)}{(\Psi_{j+1}^n - \Psi_{j-1}^n)} \quad (7.148)$$

if  $(\Psi_j - \Psi_{j-1})(\Psi_{j+1} - \Psi_j) > 0$ , and by  $\Delta_j^n \Psi = 0$  otherwise. The algorithm is easily extended to two dimensions by using expression (7.135) rather than (7.145) as the



initial step. Although considerably less diffusive than the upstream scheme, this algorithm still contains scale-dependent diffusion. Once the spatial distribution of the transported quantity has diffused to a preferred shape, the diffusion decreases considerably.

## 7.6 Advanced Eulerian Methods

Several advanced schemes for solving the advection equation have been developed with the purpose of avoiding the spurious oscillations found in high-order methods and the excessive numerical diffusion characteristic of low-order methods. They can be viewed as an extension of some of the fundamental methods discussed in the previous sections.

### The MPDATA Scheme of Smolarkiewicz

The *Multidimensional Definite Advection Transport Algorithm* (MPDATA) proposed by Smolarkiewicz (1983, 1984) focuses on compensating the first-order error of the upstream scheme by reducing the implicit numerical diffusion. Starting from the upstream scheme:

$$\Psi_j^* = \Psi_j^n - \alpha(\Psi_j^n - \Psi_{j-1}^n) \quad (7.149)$$

which provides a first guess  $\Psi_j^*$  for the solution at time  $t_{n+1}$ , the algorithm uses a second step in which the velocity  $c$  is replaced by a compensatory “anti-diffusion velocity”  $u^A$  defined as

$$u^A = \frac{K_{visc}}{\Psi} \frac{\partial \Psi}{\partial x} \quad \text{for } \Psi > 0 \quad \text{and} \quad u^A = 0 \quad \text{for } \Psi = 0 \quad (7.150)$$

with  $K_{visc} = 0.5(c\Delta x - c^2\Delta t) = 0.5 c\Delta x(1 - \alpha)$ . Here, the ratio  $(1/\Psi) \partial \Psi / \partial x$  is calculated iteratively using the latest estimate  $\Psi_j^{n+1}$  of the solution ( $\Psi_j^*$  at the first iterative step). The value of the anti-diffusion velocity at half mesh point  $j + \frac{1}{2}$  is therefore

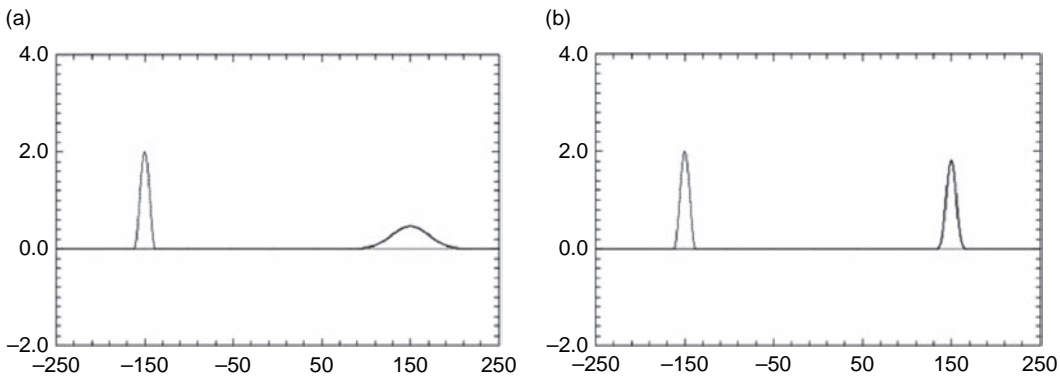
$$u^A = \frac{K_{visc}}{\Psi} \frac{\partial \Psi}{\partial x} = \frac{2K_{visc}}{\Delta x} \left( \frac{\Psi_{j+1}^{n+1} - \Psi_j^{n+1}}{\Psi_{j+1}^{n+1} + \Psi_j^{n+1} + \varepsilon} \right) \quad (7.151)$$

where  $\varepsilon$  is a small value that ensures that  $u^A$  is equal to zero when  $\Psi_j^*$  and  $\Psi_{j+1}^*$  are equal to zero. The “anti-diffusion” step becomes:

$$\begin{aligned} \Psi_j^{n+1} = & \Psi_j^* - \frac{\Delta t}{2\Delta x} \left[ \left( u_{j+1/2}^A + |u_{j+1/2}^A| \right) \Psi_j^* + \left( u_{j+1/2}^A - |u_{j+1/2}^A| \right) \Psi_{j+1}^* \right] \\ & + \frac{\Delta t}{2\Delta x} \left[ \left( u_{j+1/2}^A + |u_{j+1/2}^A| \right) \Psi_j^* + \left( u_{j+1/2}^A - |u_{j+1/2}^A| \right) \Psi_{j-1}^* \right] \end{aligned} \quad (7.152)$$

Several iterations can be performed to improve accuracy.

This simple and computationally efficient algorithm is positive definite (if the initial condition is positive) with considerably less implicit diffusion than in the



**Figure 7.20** Comparison between two advection algorithms applied to a cosine-shaped function (resolved with 12 intervals). (a) first-order upwind scheme; (b) second-order accurate Smolarkiewicz scheme in which the numerical diffusion that characterizes the upwind scheme is compensated by the introduction of an “anti-diffusion” velocity. As in Figure 7.9, the adopted grid is uniform with 500 cells. The Courant number is 0.5 and the solution is shown after 1600 time steps. From Smolarkiewicz (2006).

upstream method (Figure 7.20). It is stable under the CFL condition. It does not preserve monotonicity of the transported quantities and, in general, the solutions are not free from small oscillations. The algorithm can easily be extended to multiple dimensions, with an anti-diffusion pseudo velocity defined in each direction. Smolarkiewicz (2006) expanded his MPDATA algorithm to arbitrary finite volume frameworks.

### The SHASTA Scheme of Boris and Book

The *Sharp and Smooth Transport Algorithm* (SHASTA) proposed by Boris and Book (1973) is an Eulerian finite difference algorithm that makes use of the flux-corrected transport (FCT) technique described in Section 7.5. It ensures monotonicity of the solution, conserves mass, and handles steep gradients and shocks particularly well. The scheme includes an advection step followed by a corrective step that reduces the effect of the diffusion produced by the first step. We consider here the 1-D case and assume a variable velocity  $u(x)$ .

**Advection step.** The SHASTA algorithm first defines fluid elements formed by connecting linearly adjacent values ( $\Psi_j$  and  $\Psi_{j+1}$  in Figure 7.21). Each resulting trapezoidal element is displaced by the distance  $u \Delta t$ . Since the wind velocity  $u(x)$  is variable in space, the advection is not a simple translation of the initial element; contraction or dilatation along  $x$  can take place. We assume a Courant number less than 0.5, so that the function at grid point  $j$  can never be advected further than the grid cell boundaries. After the displacement of the function is completed, the displaced elements are interpolated back onto the original Eulerian grid (see Figure 7.21).

In their algorithm, Boris and Book (1973) prescribe the wind velocities at the grid points  $x_{j-1}$ ,  $x_j$ ,  $x_{j+1}$ , and at the intermediate time level  $t_{n+1/2}$ . They deduce at each grid point  $j$  a first approximate value for the function  $\Psi$  and time level  $t_{n+1}$ :

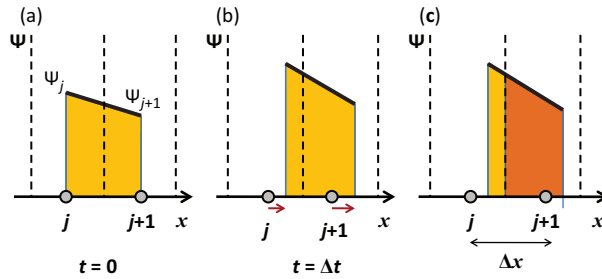


Figure 7.21

Advection of a fluid element. (a) Initial condition. (b) Location and shape of the fluid element after the advection step. During a time step  $\Delta t$  the two boundaries of the fluid element at location  $j$  and  $j + 1$  are displaced by a distance  $u_j \Delta t$  and  $u_{j+1} \Delta t$ , respectively. Here, the wind velocities are provided at the intermediate time  $t_{n+1/2}$ . At the end of the advection step, the fluid element is deformed if the velocity  $u(x)$  is not uniform. (c) Interpolation of the fluid element onto the grid. The light orange fraction remains in cell  $j$  while the darker orange fraction goes into cell  $j + 1$ . Adapted from Boris and Book (1973).

$$\Psi_j^* = \frac{1}{2} \mathcal{Q}_-^2 (\Psi_{j-1}^n - \Psi_j^n) + \frac{1}{2} \mathcal{Q}_+^2 (\Psi_{j+1}^n - \Psi_j^n) + (\mathcal{Q}_- + \mathcal{Q}_+) \Psi_j^n \quad (7.153)$$

where

$$\mathcal{Q}_- = \frac{\frac{1}{2} + u_j^{n+1/2} \frac{\Delta t}{\Delta x}}{1 - \frac{\Delta t}{\Delta x} (u_{j-1}^{n+1/2} - u_j^{n+1/2})} \quad (7.154)$$

and

$$\mathcal{Q}_+ = \frac{\frac{1}{2} - u_j^{n+1/2} \frac{\Delta t}{\Delta x}}{1 + \frac{\Delta t}{\Delta x} (u_{j+1}^{n+1/2} - u_j^{n+1/2})} \quad (7.155)$$

For a uniform velocity  $c$ , the displacement of the trapezoid corresponds to a translation without deformation, and expression (7.153) becomes

$$\Psi_j^* = \Psi_j^n - \frac{\alpha}{2} (\Psi_{j+1}^n - \Psi_{j-1}^n) + \left( \frac{1}{8} + \frac{\alpha^2}{2} \right) (\Psi_{j+1}^n - 2\Psi_j^n + \Psi_{j-1}^n) \quad (7.156)$$

with  $\alpha = c \Delta t / \Delta x$ . This expression includes a two-sided differencing expression that approximates the advection, and a diffusion approximation where the diffusion coefficient is the sum of an independent term ( $1/8$ ) and a velocity-dependent term ( $\alpha^2/2$ ). This second term is smaller since  $c \Delta t / \Delta x$  is chosen to be less than 0.5. Without the velocity-independent diffusivity, (7.156) is identical to the Lax–Wendroff algorithm. Following the von Neumann analysis, the amplification coefficient associated with the advective step is

$$g(k) = 1 + \left( \frac{1}{4} + \alpha^2 \right) [1 - \cos(k\Delta x)] - i\alpha \sin(k\Delta x) \quad (7.157)$$

and the corresponding amplification factor is

$$|g(k)| = \left\{ \left( 1 - \frac{1}{4}(1 - \cos(k\Delta x)) \right)^2 - \frac{\alpha^2}{2}(1 - 2\alpha^2) \left( (1 - \cos(k\Delta x))^2 \right) \right\}^{1/2} \quad (7.158)$$

The value of this factor is smaller than one for all wave harmonics if the Courant number is less than 0.5.

**Correction (anti-diffusion) step.** Assuming that the diffusivity in (7.156) is only weakly velocity-dependent ( $\alpha \ll 0.5$ ), a second step is applied to remove the excessive diffusion produced by the advection step. Thus, we write:

$$\Psi_j^{n+1} = \Psi_j^* - \frac{1}{8} (\Psi_{j+1}^* - 2\Psi_j^* + \Psi_{j-1}^*) \quad (7.159)$$

where  $\Psi_j^*$  is the approximation for the transported function derived by the first (advective) step. This can be rewritten as

$$\Psi_j^{n+1} = \Psi_j^* - (f_{j+1/2} - f_{j-1/2}) \quad (7.160)$$

where

$$f_{j\pm 1/2} = \pm \frac{1}{8} (\Psi_{j\pm 1}^* - \Psi_j^*) \quad (7.161)$$

represents the amount of material (“flux”) crossing the boundaries of grid cell  $j$  during the time step  $\Delta t$ . The amplification coefficient associated with the anti-diffusion step

$$g(k) = 1 + \frac{1}{4}(1 - \cos(k\Delta x)) \quad (7.162)$$

is real, so that the anti-diffusion step does not affect the phase properties of the solution.

The overall amplification factor for the two consecutive steps is:

$$|g(k)| = \left\{ \left( 1 - \frac{1}{16}(1 - \cos(k\Delta x))^2 \right)^2 - \frac{\alpha^2}{2}(1 - 2\alpha^2)(1 - \cos(k\Delta x))^2 \left( 1 + \frac{1}{4}(1 - \cos(k\Delta x)) \right)^2 \right\}^{1/2} \quad (7.163)$$

in which the velocity-dependent (or  $\alpha$ -dependent) term is generally small. Again, the method is stable when the amplification factor is less than or equal to 1.

The anti-diffusion correction step can introduce spurious extrema and negative values, which can again be avoided by applying an FCT constraint. The monotonicity of the solution is indeed preserved if the anti-diffusion flux  $f$  never produces values for  $\Psi$  at any grid point  $j$  that are larger than the values at the neighboring points. This is achieved if, rather than using (7.161), the value of the flux is replaced by the following FCT condition:

$$f_{j+1/2} = \text{sign}(\Delta_{j+1/2}) \max \left\{ 0, \min \left[ \Delta_{j-1/2} \text{sign}(\Delta_{j+1/2}), \frac{1}{8} |\Delta_{j+1/2}|, \Delta_{j+3/2} \text{sign}(\Delta_{j+1/2}) \right] \right\} \quad (7.164)$$

where

$$\Delta_{j+1/2} = \Psi_{j-1} - \Psi_j \quad (7.165)$$

The FCT step can be improved by replacing the factor 1/8 in (7.164) with a factor that accounts for the velocity dependence of the diffusivity. Further improvements to the SHASTA method that lead to more accurate solutions have been introduced by Boris and Book (1976).

### The Piecewise Parabolic Method

In their piecewise parabolic method (PPM), Colella and Woodward (1984) assume that the subgrid distribution  $\psi(x)$  of the tracer concentration inside cell  $j$  can be represented by a quadratic function:

$$\psi(x) = \psi_{j-1/2} + y(x) \left[ \psi_{j+1/2} - \psi_{j-1/2} + d_j(1 - y(x)) \right] \quad (7.166)$$

where  $y = (x - x_{j-1/2})/\Delta x$ . Coefficients  $\psi_{j-1/2}$  and  $\psi_{j+1/2}$  are the values of  $\psi(x)$  at the boundaries  $x_{j-1/2}$  and  $x_{j+1/2}$  of cell  $j$ , and

$$d_j = 6 \left[ \Psi_j^n - \frac{1}{2} (\psi_{j+1/2} + \psi_{j-1/2}) \right] \quad (7.167)$$

where  $\Psi_j^n$  is the mean concentration in grid cell  $j$  at time  $t_n$ . For constant spacing  $\Delta x$ , it can be shown through interpolation from  $\Psi_j^n$  in nearby zones that the value of  $\psi(x)$  at for example  $x_{j+1/2}$  can usually be expressed as:

$$\psi_{j+1/2} = \frac{7}{12} (\Psi_j^n + \Psi_{j+1}^n) - \frac{1}{12} (\Psi_{j+2}^n + \Psi_{j-1}^n) \quad (7.168)$$

When the grid cells are unequally spaced, the expressions for  $\psi_{j+1/2}$  and  $\psi_{j-1/2}$  are more complicated. Colella and Woodward (1984) propose a slightly modified interpolation procedure in the presence of sharp discontinuities (shocks) to ensure that these discontinuities remain sharp during the advection step. The effect of this “steepening” process (Carpenter *et al.*, 1990) is shown in Figure 7.22. Large discontinuities can still produce oscillations in the post-shock flow. In this case, it is advised to introduce some dissipation in the neighborhood of the shock. This can be achieved, for example, by flattening the interpolation profile in the vicinity of the discontinuity, which is equivalent to reducing locally the order of the method. In this case, coefficient  $\psi_{j+1/2}$  can be replaced, for example, by

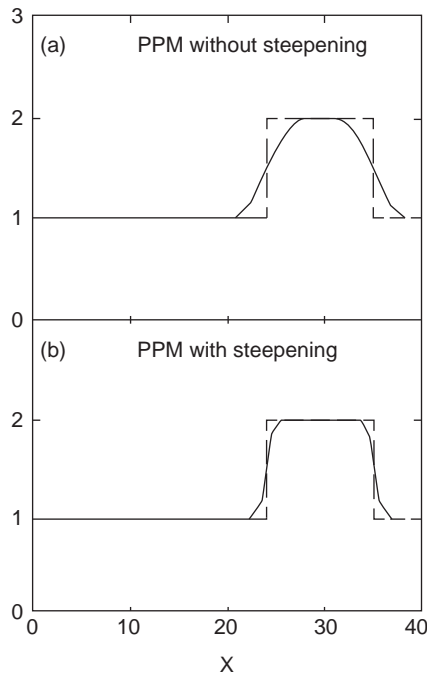
$$\psi_{j+1/2}^{flat} = \Psi_j f_j + \psi_{j+1/2} (1 - f_j) \quad (7.169)$$

where  $f_j \in [0, 1]$  is an adjustable factor. Far away from discontinuities or if the shock profile is sufficiently broad, coefficient  $f_j$  should be set to zero.

The solution at time  $t_{n+1}$  is obtained by:

$$\Psi_j^{n+1} = \Psi_j^n - \alpha (a_{j+1/2}^n - a_{j-1/2}^n) \quad (7.170)$$

where, for example,



**Figure 7.22** Advection of a square function using the piecewise parabolic method (PPM) (Carpenter *et al.*, 1990) without steepening (a) and with steepening (b). The spatial domain extends over  $40 \Delta x$  with cyclic boundary conditions. The Courant number  $\alpha = 0.5$ . The numerical solution is shown after 1000 time steps (12.5 revolutions). From Müller (1992). Copyright © American Meteorological Society, used with permission.

$$a_{j+1/2}^n = \psi_{j+1/2} - \frac{\alpha}{2} \left[ \psi_{j+1/2} - \psi_{j-1/2} - \left( 1 - \frac{2\alpha}{3} \right) d_j \right] \quad (7.171)$$

Other algorithms for rendering the PPM shape-preserving are presented by Colella and Sekora (2008).

### The Crowley–Tremback–Bott Scheme

To improve the accuracy of the first-order upstream method, Crowley (1968), Tremback *et al.* (1987), and Bott (1989a, 1989b) have also proposed to represent the transported quantity within each grid cell  $j$  by a polynomial  $\psi_{j,\ell}$  of order  $\ell$  (with  $\ell$  assumed to be an even integer number). Thus, at time level  $n$ , we write

$$\psi_{j,\ell}^n(y) = \sum_{k=0}^{\ell} a_{j,k}^n y^k \quad (7.172)$$

where  $y = (x - x_j)/\Delta x$  is a dimensionless variable such that  $-1/2 \leq y \leq 1/2$ . We assume again that the grid spacing  $\Delta x$  is uniform. Coefficients  $a_{j,k}^n$  are determined from the requirement that the value of  $\psi_{j,\ell}^n(y)$  agree with the value of  $\psi_j^n$  at grid points

**Table 7.2** Coefficients  $a_{j,k}$  for the  $\ell = 2$  and  $\ell = 4$  versions of the Bott's area preserving flux form algorithm (after Bott, 1989a and Chlond, 1994)

	$\ell = 2$	$\ell = 4$
$a_{j,0}$	$-\frac{1}{24} [\Psi_{j+1} - 26\Psi_j + \Psi_{j-1}]$	$\frac{1}{1920} [9\Psi_{j+2} - 116\Psi_{j+1} + 2134\Psi_j - 116\Psi_{j-1} + 9\Psi_{j-2}]$
$a_{j,1}$	$\frac{1}{2} [\Psi_{j+1} - \Psi_{j-1}]$	$\frac{1}{48} [-5\Psi_{j+2} + 34\Psi_{j+1} - 34\Psi_{j-1} + 5\Psi_{j-2}]$
$a_{j,2}$	$\frac{1}{2} [\Psi_{j+1} - 2\Psi_j + \Psi_{j-1}]$	$\frac{1}{48} [-3\Psi_{j+2} + 36\Psi_{j+1} - 66\Psi_j + 36\Psi_{j-1} - 3\Psi_{j-2}]$
$a_{j,3}$	—	$\frac{1}{12} [\Psi_{j+2} - 2\Psi_{j+1} + 2\Psi_{j-1} - \Psi_{j-2}]$
$a_{j,4}$	—	$\frac{1}{12} [\Psi_{j+2} - 4\Psi_{j+1} + 6\Psi_j - 4\Psi_{j-1} + \Psi_{j-2}]$

( $i = j - \ell/2, \dots, j, \dots, j + \ell/2$ ), and that the area covered by  $\Psi_{j,\ell}^n(y)$  in grid cell  $j$  equals  $\Psi_j^n \Delta x$ . Thus, the coefficients  $a_{j,k}^n$  are expressed as a function of the values of  $\Psi_j, \Psi_{j+1}, \dots$  at the  $(\ell + 1)$  neighboring points. Table 7.2 provides the values of the coefficients derived for second-order and fourth-order polynomials, while Tremback *et al.* (1987) also considers higher order schemes.

As before, the solution for grid cell  $j$  at time  $t_{n+1}$  is provided by the finite-volume approximation:

$$\Psi_j^{n+1} = \Psi_j^n - \frac{\Delta t}{\Delta x} (F_{j+1/2} - F_{j-1/2})$$

In the Crowley–Tremback–Bott scheme, the fluxes  $F_{j+1/2}$  and  $F_{j-1/2}$  at the right and left boundaries of grid cell  $j$  are estimated from (7.122) and (7.123) in which  $\psi_j(y, t)$  is replaced by its polynomial approximation (7.172) of order  $\ell$ .

In the more general case where the velocity  $u$  in the  $x$ -direction is spatially variable, (7.123) is replaced by (see Bott, 1989a, 1989b; Chlond, 1994)

$$F_{j+1/2} = \frac{\Delta t}{\Delta x} (I_{j+1/2}^+ - I_{j+1/2}^-) \quad (7.173)$$

where

$$I_{j+1/2}^+ = \int_{1/2-\alpha^+}^{1/2} \psi_j(z, t) dz \quad \text{and} \quad I_{j+1/2}^- = \int_{-1/2}^{1/2-\alpha^+} \psi_j(z, t) dz$$

are area integrals in which

$$z = (y - x_j)/\Delta x$$

$$\alpha^+ = \alpha_{j+1/2}^+ = \max \left( 0, u_{j+1/2}^n \Delta t / \Delta x \right) \quad \text{and} \quad \alpha^- = \alpha_{j+1/2}^- = \max \left( 0, u_{j+1/2}^n \Delta t / \Delta x \right)$$

Again, function  $\psi_j(z, t)$  can be approximated by an area-preserving polynomial (7.172). Substitution of this polynomial into the above expressions yields the following expressions for integrals  $I_{j+1/2}^+$  and  $I_{j+1/2}^-$ :

$$I_{j+1/2, \ell}^+ = \sum_{k=0}^{\ell} a_{j,k} \frac{1 - (1 - 2\alpha^+)^{k+1}}{(k+1)2^{k+1}} \quad (7.174)$$

and

$$I_{j+1/2, \ell}^- = \sum_{k=0}^{\ell} a_{j,k} (-1)^k \frac{1 - (1 - 2\alpha^-)^{k+1}}{(k+1)2^{k+1}} \quad (7.175)$$

This polynomial fitting method is not exempt from localized unphysical oscillations near sharp spatial gradients. To address the problem, Bott (1989a, 1989b) introduces nonlinear flux limiters and imposes that the total amount of outflow from gridbox  $j$  during a time step  $\Delta t$  be limited to  $\Psi_j^n \Delta x / \Delta t$ . In addition, the flux  $F_{j+1/2}$  is set to zero if it does not have the same sign as the velocity  $u_{j+1/2}$ . These two conditions are fulfilled if, rather than using (7.173), the flux  $F_{j+1/2}$  is expressed as

$$F_{j+1/2} = \frac{\Delta x}{\Delta t} \left( \beta_{j+1/2} \tilde{I}_{j+1/2, \ell}^+ - \beta_{j+3/2} \tilde{I}_{j+1/2, \ell}^- \right) \quad (7.176)$$

where

$$\tilde{I}_{j+1/2, \ell}^+ = \max \left( I_{j+1/2, \ell}^+, 0 \right), \quad \tilde{I}_{j+1/2, \ell}^- = \max \left( I_{j+1/2, \ell}^-, 0 \right)$$

and

$$\beta_{j+1/2} = \min \left\{ 1, \frac{\Psi_j^n}{\max \left( \tilde{I}_{j+1/2, \ell}^+ + \tilde{I}_{j+1/2, \ell}^-, \varepsilon \right)} \right\}$$

Here  $\varepsilon$  is a small value added to avoid numerical unstable situations if  $\tilde{I}_{j+1/2, \ell}^+ + \tilde{I}_{j+1/2, \ell}^- = 0$ .

Spatial functional distributions other than polynomials have been used to interpolate the dependent variables between grid points. Spalding (1972), for example, uses an exponential fitting technique, which prevents the spurious oscillations associated with the Bott scheme near sharp gradients, and ensures positivity of the solution. The method, however, is diffusive and computationally expensive. Chlond (1994) uses a hybrid scheme in which the polynomial and exponential interpolation methods are combined. A switch is used so that the polynomial scheme is applied in regions where the distribution of the transported quantity is smooth, and the exponential fitting technique is applied near sharp gradients.

### The Prather Scheme

The algorithm presented by Prather (1986) is an extension of the Russell and Lerner scheme in which the advected function inside a grid cell is represented by a second-order polynomial. In each grid cell, we represent the 3-D  $(x, y, z)$  distribution of the tracer mixing ratio  $\psi$  by



$$\psi(x, y, z) = a_0 + a_x x + a_{xx} x^2 + a_y y + a_{yy} y^2 + a_z z + a_{zz} z^2 + a_{xy} xy + a_{yz} yz + a_{xz} xz \quad (7.177)$$

within a rectangular gridbox of volume  $V = \Delta x \Delta y \Delta z$  with  $0 \leq x \leq \Delta x$ ,  $0 \leq y \leq \Delta y$ , and  $0 \leq z \leq \Delta z$ . The same function can also be expressed by a linear combination of orthogonal second-order polynomials  $K_k$ :

$$\psi(x, y, z) = m_0 K_0 + m_x K_x + m_y K_y + m_{yy} K_{yy} + m_z K_z + m_{zz} K_{zz} + m_{xy} K_{xy} + m_{yz} K_{yz} + m_{xz} K_{xz} \quad (7.178)$$

where  $m_k$  are moment coefficients. By definition, the orthogonal functions satisfy the conditions:

$$\int_V K_i K_j dV = 1 \quad (i \neq j) \quad (7.179)$$

where  $dV = dx dy dz$ . Prather (1986) provides ten orthogonal polynomials that apply to the algorithm in three dimensions:

$$\begin{aligned} K_0 &= 1 \\ K_x(x) &= x - \frac{\Delta x}{2} & K_{xx}(x) &= x^2 - x\Delta x + \frac{(\Delta x)^2}{6} \\ K_y(y) &= y - \frac{\Delta y}{2} & K_{yy}(y) &= y^2 - y\Delta y + \frac{(\Delta y)^2}{6} \\ K_z(z) &= z - \frac{\Delta z}{2} & K_{zz}(z) &= z^2 - z\Delta z + \frac{(\Delta z)^2}{6} \\ K_{xy}(x, y) &= \left(x - \frac{\Delta x}{2}\right) \left(y - \frac{\Delta y}{2}\right) \\ K_{yz}(y, z) &= \left(y - \frac{\Delta y}{2}\right) \left(z - \frac{\Delta z}{2}\right) \\ K_{xz}(x, z) &= \left(x - \frac{\Delta x}{2}\right) \left(z - \frac{\Delta z}{2}\right) \end{aligned}$$

with appropriate normalization factors. He also provides linear expressions that relate coefficients  $a_k$  and  $m_k$ . An upstream method is used to transport simultaneously the zeroth-order (mass), first-order (slope), and second-order (curvature) moments of the tracer distribution in each grid cell. The moments  $S_i$  are defined by

$$S_0 = \int_V \psi(x, y, z) K_0 dV = m_0 V \quad (7.180)$$

$$S_x = \frac{6}{\Delta x} \int_V \psi(x, y, z) K_x(x) dV = \frac{m_x V \Delta x}{2} \quad (7.181)$$

$$S_{xx} = \frac{30}{(\Delta x)^2} \int_V \psi(x, y, z) K_{xx}(x) dV = \frac{m_{xx} V (\Delta x)^2}{6} \quad (7.182)$$

$$S_{xy} = \frac{36}{\Delta x \Delta y} \int_V \psi(x, y, z) K_{xy}(x, y) dV = \frac{m_{xy} V \Delta x \Delta y}{4} \quad (7.183)$$

Parallel expressions are derived for  $S_y$ ,  $S_{yy}$ ,  $S_{xy}$ ,  $S_z$ ,  $S_{zz}$ , and  $S_{xz}$ .

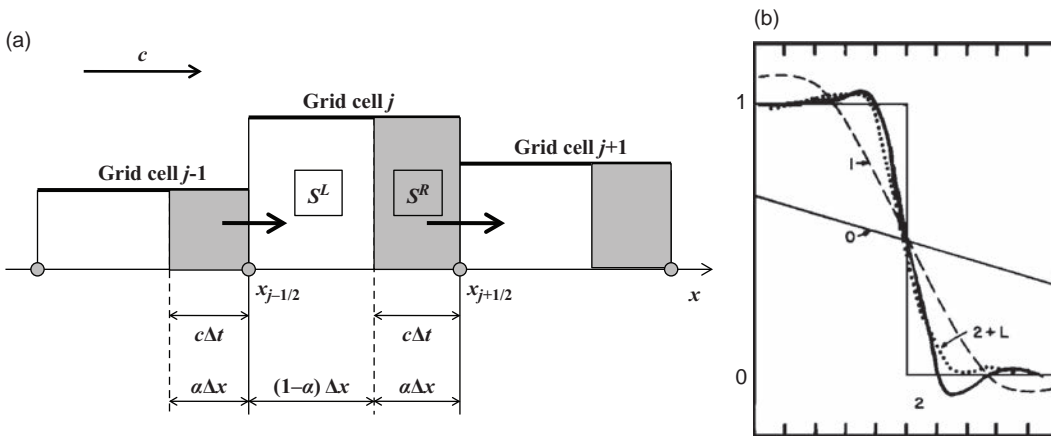


Figure 7.23

(a) Advection along direction  $x$  of moments during a time step  $\Delta t$  from grid cell  $j$  to  $j + 1$  after decomposition of moment  $S$  into sub-moments  $S^L$ , which remains in gridbox  $j$ , and  $S^R$ , which is transferred from gridbox  $j$  to adjacent downwind gridbox  $j + 1$ . Coefficient  $\alpha = V^R/(V^R + V^L) = c\Delta t/\Delta x$  is the Courant number. (b) Comparison of the distribution of a tracer advected across 200 gridboxes by the basic upstream scheme (0), the first-order moments method (1) and the second-order moments method (2). Note the presence of overshoots and undershoots in the second-order scheme. Positivity is ensured by placing limits on the high-order moments (curve  $2 + L$ ). From Prather (1986).

To illustrate the method, we consider a 2-D problem and assume that the velocity  $c > 0$  of the flow is uniform and directed in the  $x$ -direction. The distribution of the tracer mixing ratio inside a 2-D gridbox is expressed by

$$\psi(x, y) = m_0 K_0 + m_x K_x(x) + m_{xx} K_{xx}(x) + m_y K_y(y) + m_{yy} K_{yy}(y) + m_{xy} K_{xy}(x, y) \quad (7.184)$$

Within a cell  $j$  of volume  $V$ , we define the sub-volume  $V^R$  of the fluid that will be removed from this cell and added to the neighboring cell  $j + 1$  over a time step  $\Delta t$  (Figure 7.23):

$$V^R = c\Delta t \Delta y \Delta z \quad (7.185)$$

The volume  $V^L$  of the fluid remaining in the original cell  $j$  is

$$V^L = (\Delta x - c\Delta t) \Delta y \Delta z \quad (7.186)$$

The method involves two consecutive steps:

**First step:** Decomposition of the moments  $S_k$  for each grid cell into sub-moments  $S_k^R$  and  $S_k^L$  associated with the fraction of the tracer that is advected to the downwind grid cell and the fraction of the tracer that remains in the grid cell during time step  $\Delta t$ . We have

$$\begin{aligned} S_0^R &= \alpha[S_0 + (1 - \alpha)S_x + (1 - \alpha)(1 - 2\alpha)S_{xx}] \\ S_x^R &= \alpha^2[S_x + 3(1 - \alpha)S_{xx}] & S_y^R &= \alpha[S_y + (1 - \alpha)S_{xy}] \\ S_{xx}^R &= \alpha^3 S_{xx} & S_{yy}^R &= \alpha S_{yy} & S_{xy}^R &= \alpha^2 S_{xy} \end{aligned}$$

where the Courant number  $\alpha = c\Delta t/\Delta x = V^R/V$  is assumed to be smaller than 1. The fraction of the tracer remaining in the cell  $j$  during time step  $\Delta t$  is located in the sub-box of volume  $V^L$  extending from bounds  $x_{j-1/2}$  to  $x_{j+1/2} - c\Delta t$ . The corresponding sub-moments  $S_K^L$  are:

$$\begin{aligned} S_0^L &= (1 - \alpha)[S_0 - \alpha S_x - \alpha(1 - 2\alpha)S_{xx}] \\ S_x^L &= (1 - \alpha)^2[S_x - 3\alpha S_{xx}] \quad S_y^L = (1 - \alpha)[S_y - \alpha S_{xy}] \\ S_{xx}^L &= (1 - \alpha)^3 S_{xx} \quad S_{yy}^L = (1 - \alpha)S_{yy} \quad S_{xy}^L = (1 - \alpha)^2 S_{xy} \end{aligned}$$

**Second step: Advection step and reconstruction of the moments for the complete grid cell.** The advection is performed by transporting moments  $S_K^R$  from grid cell  $j$  to adjacent cell  $j + 1$ , while maintaining moments  $S_K^L$  in their original box. For time  $t_{n+1}$ , the moments for the entire gridbox  $j$  can be reconstructed by calculating the updated moments:

$$\begin{aligned} S_0 &= S_0^R + S_0^L \\ S_x &= \alpha S_x^R + (1 - \alpha)S_x^L + 3[(1 - \alpha)S_0^R - \alpha S_0^L] \\ S_{xx} &= \alpha^2 S_{xx}^R + (1 - \alpha)^2 S_{xx}^L + 5\{\alpha(1 - \alpha)(S_x^R - S_x^L) + (1 - 2\alpha)[(1 - \alpha)S_0^R - \alpha S_0^L]\} \\ S_y &= S_y^R + S_y^L \quad S_{yy} = S_{yy}^R + S_{yy}^L \\ S_{xy} &= \alpha S_{xy}^R + (1 - \alpha)S_{xy}^L + 3[(1 - \alpha)S_y^R - \alpha S_y^L] \end{aligned}$$

From these new moments derived on the full grid cell, one derives the coefficients  $m_k$  using (7.180)–(7.183), and from there the spatial distribution of the tracer mixing ratio inside the grid cell.

The Prather method is less diffusive than the slope scheme (Figure 7.23b), but it adds to the computational and memory requirements because at each time step ten moments must be computed and stored for each grid cell in the 3-D case. As with other high-order methods, the scheme produces overshoots and undershoots that can lead to negative solutions. Placing a limit on the magnitude of the higher-order moments can ensure positivity of the solution (Figure 7.23b). The scheme is absolutely stable for Courant numbers ranging from 0.2764 to 0.7236 (Prather, 1986), but is marginally unstable over the rest of the domain  $[0, 1]$ . Phase errors are extremely small.

## 7.7 Lagrangian Methods

Lagrangian advection methods divide the atmosphere into a large number of air parcels and follow the displacement of their centroids as a function of time. Tracer mixing ratios are conserved in these displacements. The trajectory of the centroid is determined from the wind velocity at the centroid location. Because the wind velocities are generally provided at the discrete points where observations are performed or at the grid points of an Eulerian meteorological model, their values must be interpolated at the location of each centroid. Although the motion of the air parcels does not follow any grid, model results can still be provided at regularly spaced grid points by interpolation from the randomly located air parcels situated in the vicinity of these grid points.

In the general case of a multidimensional model with variable wind velocity  $\mathbf{v}(\mathbf{r}, t)$ , where  $\mathbf{r}$  is the parcel location, the parcel trajectories are calculated from:

$$\frac{d\mathbf{r}(t)}{dt} = \mathbf{v}(\mathbf{r}(t), t) \quad (7.187)$$

A second-order accurate solution of this differential equation is obtained by solving, for example, the implicit expression (4.270):

$$\mathbf{r}(t + \Delta t) = \mathbf{r}(t) + \frac{1}{2} \Delta t [\mathbf{v}(\mathbf{r}(t), t) + \mathbf{v}(\mathbf{r}(t + \Delta t), t + \Delta t)] \quad (7.188)$$

where  $\mathbf{r}(t)$  is the location of the air parcel at time  $t$  (departure point) and  $\mathbf{r}(t + \Delta t)$  is the position of the parcel at time  $t + \Delta t$  (arrival point). This implicit equation can be solved by an iteration and interpolation procedure (Kida, 1983; Stohl, 1998).

The use of computationally expensive iterative implicit methods can be avoided if the position vector  $\mathbf{r}(t)$  is expanded by a Taylor series in which terms of order higher than  $(\Delta t)^2$  are neglected:

$$\mathbf{r}(t + \Delta t) = \mathbf{r}(t) + \left(\frac{d\mathbf{r}}{dt}\right)_t \Delta t + \frac{1}{2} \left(\frac{d^2\mathbf{r}}{dt^2}\right)_t (\Delta t)^2 + O[(\Delta t)^3] \approx \mathbf{r}(t) + \mathbf{v}\Delta t + \frac{1}{2} \boldsymbol{\gamma}(\Delta t)^2 \quad (7.189)$$

Here, the wind velocity  $\mathbf{v} = (d\mathbf{r}/dt)_t$  and the acceleration  $\boldsymbol{\gamma} = (d\mathbf{v}/dt)_t = (d^2\mathbf{r}/dt^2)_t$  are calculated at the departure point (time level  $t$ ). The acceleration  $\boldsymbol{\gamma}$  is easily obtained from the spatial variation of the velocity field if the local rate of change in the velocity  $\Delta\mathbf{v}/\Delta t$  can be neglected over the time interval  $\Delta t$ . Thus:

$$\boldsymbol{\gamma} = \frac{d\mathbf{v}}{dt} = \frac{\partial\mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} \approx \mathbf{v} \cdot \nabla \mathbf{v} \quad (7.190)$$

In the above equations, a random wind velocity  $\mathbf{v}'$  is often added to the mean wind velocity  $\mathbf{v}$  to account for small-scale turbulent motions. An example of a simulation produced over a period of three weeks by a Lagrangian particle dispersion model following a volcanic eruption is shown in Figure 7.24.

In global Lagrangian models, the position of an air parcel is often defined by its longitude  $\lambda$ , latitude  $\phi$ , and altitude  $z$  (or pressure  $p$ ). Thus, after a time step  $\Delta t$ , an air parcel originally located at point  $[\lambda(t), \phi(t), z(t)]$  is displaced to a new point whose position  $[\lambda(t + \Delta t), \phi(t + \Delta t), z(t + \Delta t)]$  is derived from

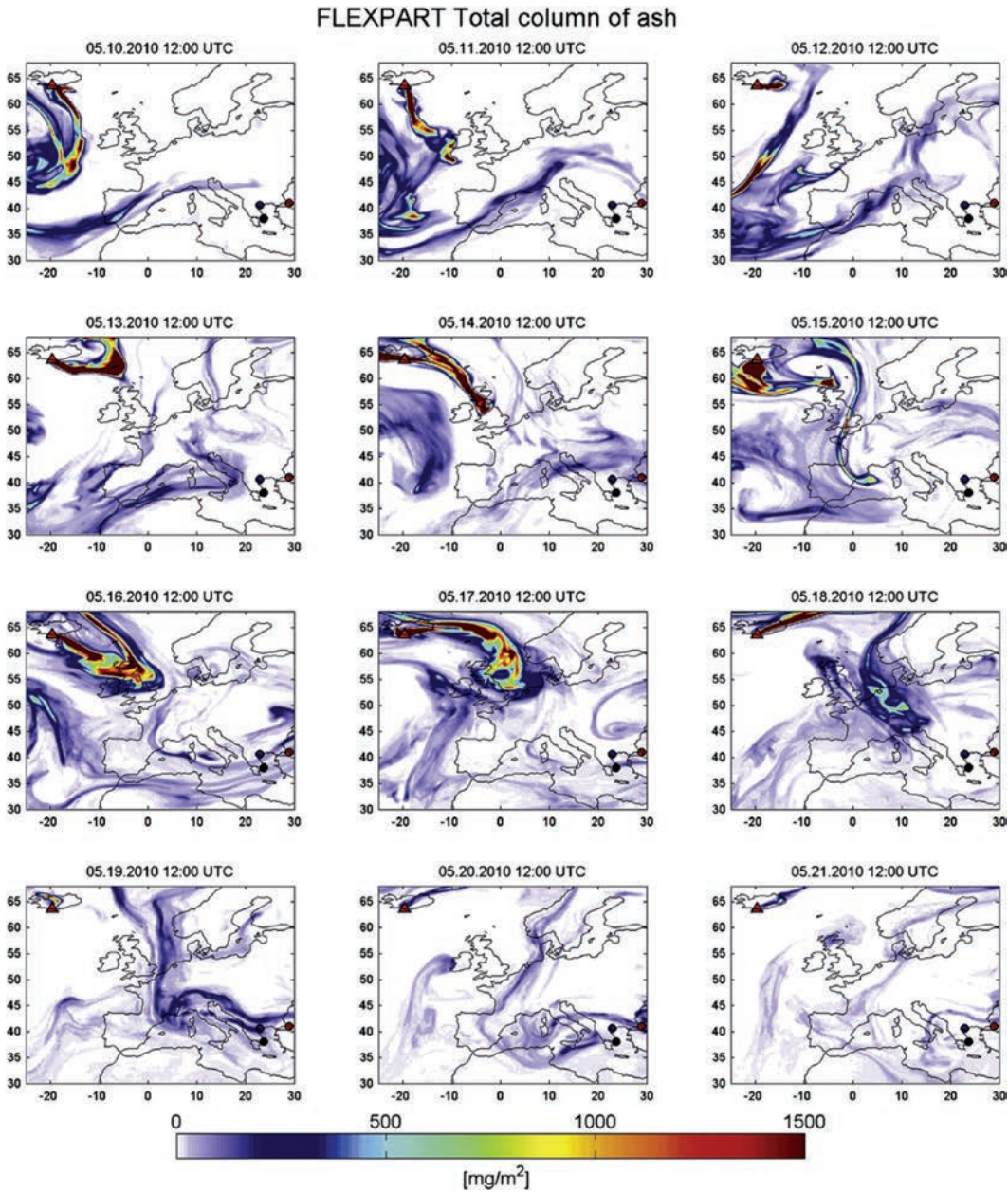
$$\lambda(t + \Delta t) = \lambda(t) + \frac{\Delta x \Delta y}{a^2 [\sin(\phi + \Delta\phi) - \sin(\phi)]} \quad (7.191)$$

$$\phi(t + \Delta t) = \phi(t) + \frac{\Delta y}{a} \quad (7.192)$$

$$z(t + \Delta t) = z(t) + \Delta z \quad (7.193)$$

where  $a$  is the Earth's radius. The geometric displacements  $\Delta x$ ,  $\Delta y$ , and  $\Delta z$  are expressed as a function of the components  $(u, v, w)$  of the wind velocity  $\mathbf{v}$  and the components  $(\gamma_x, \gamma_y, \gamma_z)$  of the wind acceleration  $\boldsymbol{\gamma}$  by

$$\Delta x = u\Delta t + \frac{1}{2} \gamma_x (\Delta t)^2 \quad \Delta y = v\Delta t + \frac{1}{2} \gamma_y (\Delta t)^2 \quad \Delta z = w\Delta t + \frac{1}{2} \gamma_z (\Delta t)^2 \quad (7.194)$$



**Figure 7.24** Simulation for the period May 10–21, 2010 of the vertically integrated concentration [ $\text{mg m}^{-2}$ ] of atmospheric ash resulting from the eruption of the Eyjafjalljokull volcano in southern Iceland. Transport was simulated using the Lagrangian particle dispersion model FLEXPART (Stohl *et al.*, 1998), which traces the displacement of a large number of particles by the mean winds to which random motions representing turbulence and convection are superimposed. The model is driven by assimilated meteorological data on a  $0.18^\circ \times 0.18^\circ$  grid with 91 levels in the vertical. Reproduced from Papayannis *et al.* (2012).

From (7.190), the three components of the acceleration ( $\gamma_x, \gamma_y, \gamma_z$ ) are expressed by

$$\gamma_x = \frac{u}{a \cos \varphi} \frac{\partial u}{\partial \lambda} + \frac{v}{a} \frac{\partial u}{\partial \varphi} + w \frac{\partial u}{\partial z} \quad (7.195)$$

$$\gamma_y = \frac{u}{a \cos \varphi} \frac{\partial v}{\partial \lambda} + \frac{v}{a} \frac{\partial v}{\partial \varphi} + w \frac{\partial v}{\partial z} \quad (7.196)$$

$$\gamma_z = \frac{u}{a \cos \varphi} \frac{\partial w}{\partial \lambda} + \frac{v}{a} \frac{\partial w}{\partial \varphi} + w \frac{\partial w}{\partial z} \quad (7.197)$$

Lagrangian methods have several advantages over Eulerian methods. First, since all tracers follow the same trajectory, a single calculation of the air parcel displacement can be used to immediately infer the transport of all tracers. Second, the stability of the algorithm is not constrained by the value of the Courant number as in the explicit Eulerian methods, so the adopted time step is limited by accuracy rather than by stability considerations. Other desirable requirements are met: during the parcel displacement, mass is conserved and the sign of the transported function is maintained. Thus, unless interpolation procedures are not carefully performed, the method also guarantees monotonicity, transportivity and locality of the solution.

The Lagrangian methods also have several disadvantages. First, inaccuracies in the interpolation of the wind velocities lead to errors in the calculation of the parcel trajectories, and these errors can accumulate as the time integration proceeds. Second, the initially uniform distribution of air parcels may become highly irregular over time as a result of errors in the wind interpolation. As a result, the tracer concentration may become under-determined in certain parts of the domain while being over-determined in others. Third, Lagrangian transport does not allow for mixing between air parcels even when they are closely located. As a result, contrary to the Eulerian algorithms that often produce excessive diffusion, Lagrangian methods require that some diffusive mixing be added to account for interactions between air parcels. This is critical in particular for the treatment of nonlinear chemistry and aerosol microphysics.

## 7.8 Semi-Lagrangian Methods

Semi-Lagrangian transport (SLT) methods combine important advantages of the Lagrangian and Eulerian methods. The upstream SLT method (Figure 7.25) consists of using Lagrangian back-trajectories to relate concentrations on a regular Eulerian grid at the end of a model time step to the concentrations at the beginning of the time step. We first consider a version of the SLT scheme in which *points* in the atmosphere with given tracer mixing ratios are displaced with the flow during a model time step to reach locations coincident with the model grid points. We then consider a finite volume version of the SLT scheme in which *volumes* of air are displaced with the flow (with no mass transfer through their boundaries) to reach at the end of the model time step a volume of air that is coincident with a model grid cell.



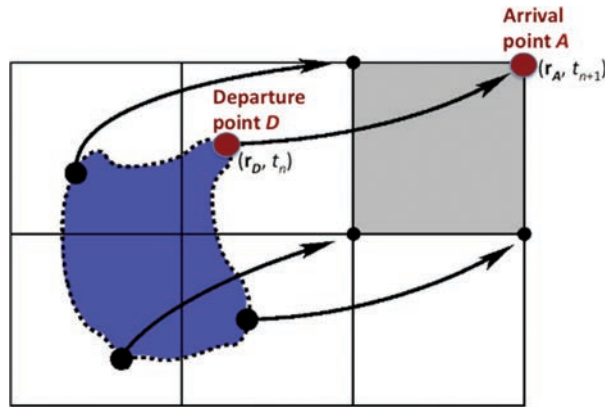


Figure 7.25

Schematic representation of the semi-Lagrangian method in two dimensions. A parcel located at the arrival point  $A$  at time level  $t_{n+1}$  was located at the departure point  $D$  at time level  $t_n$ . The value of a conserved quantity such as the mixing ratio of a passive tracer is unchanged as the parcel is displaced from point  $D$  to point  $A$  during the time step. The value of the quantity at the departure point  $D$  is derived by interpolation from the neighboring grid points at time  $t_n$ . In the finite volume version of the SLT scheme, one considers the displacement of a given volume of air (a surface in two dimensions) from its departure position (blue area) to its arrival position (gray area) coincident with a Eulerian grid cell of the model. Reproduced with permission from Peter Hjort Lauritzen (personal communication), National Center for Atmospheric Research.

### 7.8.1 Grid Point Based SLT Schemes

During a time step  $\Delta t = t_{n+1} - t_n$ , one determines the backward trajectory of the atmospheric points that reach the different Eulerian grid points of the model at time level  $t_{n+1}$ . The trajectories are calculated using interpolated gridded velocities. Consider an arrival point  $A$  at time  $t_{n+1}$  on the Eulerian grid (Figure 7.25). The location  $\mathbf{r}_D$  of the upwind departure point  $D$  at time  $t_n$  is determined by backward integration of (7.187). For example, we write the equation

$$\mathbf{r}_D = \mathbf{r}_A - \frac{\Delta t}{2} [\mathbf{v}_D + \mathbf{v}_A] \quad (7.198)$$

which has to be solved iteratively since the velocity  $\mathbf{v}_D$  depends on the location of the departure point, which is not known a priori. In general, the departure points do not coincide with model grid points. The tracer mixing ratio at departure point  $D$  and time level  $t_n$  is determined by the interpolation of the surrounding values of the mixing ratio at the closest regular Eulerian grid points. For an inert tracer, the mixing ratio  $\mu$  remains constant along the trajectory between  $D$  and  $A$ . Thus:

$$\mu(\mathbf{r}_A, t_{n+1}) = \mu(\mathbf{r}_D, t_n) \quad (7.199)$$

The starting point of the backward trajectory can be derived by replacing the 3-D advection problem with three successive 1-D problems (Seibert and Morariu, 1991). For the advection along the  $x$ -axis, the trajectory is determined by integrating

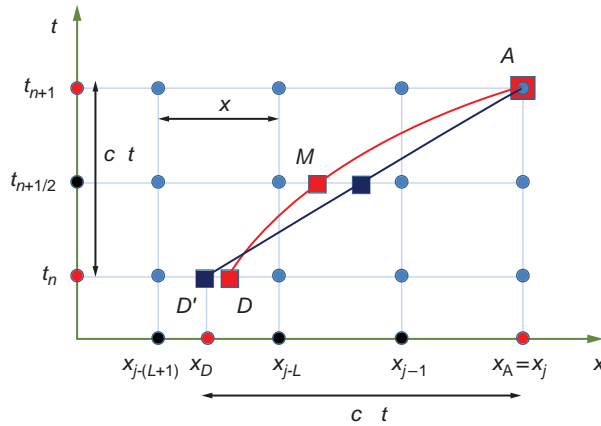


Figure 7.26

Representation of the semi-Lagrangian scheme in the 1-D case for a variable wind speed  $c(x, t)$ . The trajectory is represented by the curve  $DA$  (in red). The arrival point  $A$  at time  $t_{n+1}$  is coincident with grid point  $x_j$  of the Eulerian model grid. The location of the departure point  $D$  is derived from a back-trajectory calculation and is not coincident with a model grid point. It is located between grid points  $x_{j-(L+1)}$  and  $x_{j-L}$ . (In this figure,  $L = 2$ , but it could be larger for longer time steps). By approximating the variable wind velocity by its value at the midpoint  $M$  (or by the average between the wind speeds at the departure and arrival points), the curve  $DA$  (in red) can be approximated by the straight line  $D'A$  (in blue), and the departure point  $D$  by point  $D'$ . The approximation can be improved by iteration.

$$\frac{dx}{dt} = u(x) \quad (7.200)$$

between the departure point  $x_D$  and the arrival point (coincident with grid point  $x_A = x_j$ ; see Figure 7.26). Thus,

$$\int_{x_D}^{x_j} \frac{dx}{u(x)} = \int_{t_n}^{t_n + \Delta t} dt \quad (7.201)$$

Approximating the wind along the trajectory by

$$u(x) = u(x_j) + (x - x_j) \frac{\partial u}{\partial x} \quad (7.202)$$

(7.201) can be solved analytically and the departure point  $x_D$  is found to be

$$x_D = x_j - \left[ 1 - \exp \left( -\Delta t \frac{\partial u}{\partial x} \right) \right] u(x_j) \left( \frac{\partial u}{\partial x} \right)^{-1} \quad (7.203)$$

where the partial derivative  $\partial u / \partial x$  is numerically calculated as

$$\frac{\partial u}{\partial x} = \frac{(u_{j+1} - u_j)}{(x_{j+1} - x)} \quad \text{for } u_j \leq 0 \quad \text{and} \quad \frac{\partial u}{\partial x} = \frac{(u_j - u_{j-1})}{(x_j - x_{j-1})} \quad \text{for } u_j > 0 \quad (7.204)$$

In the simple 1-D case in which the wind velocity  $u = c$  is constant and positive (see Figure 7.26), the departure point is given by



$$x_D = x_A - c\Delta t \quad (7.205)$$

and the mixing ratio at the departure point  $D$  is provided for example by a linear interpolation between the values at the closest grid points (indices  $m - 1 = j - (L + 1)$  and  $m = j - L$  in Figure 7.26)

$$\mu(x_D, t_n) = \mu(x_{j-(L+1)}, t_n) + \frac{(x_D - x_{j-(L+1)})}{\Delta x} [\mu(x_{j-L}, t_n) - \mu(x_{j-(L+1)}, t_n)] \quad (7.206)$$

When the Courant number  $\alpha = c\Delta t/\Delta x$  is smaller than 1, the departure point  $D$  is located between grid point  $x_{j-1}$  and the arrival point  $x_j(A)$ , and it is straightforward to show that

$$\mu(x_j, t_{n+1}) = \mu(x_{j-1}, t_n) + \frac{(\Delta x - c\Delta t)}{\Delta x} [\mu(x_j, t_n) - \mu(x_{j-1}, t_n)] \quad (7.207)$$

because the mixing ratio remains constant during the displacement of the parcel. Adopting the more classic notation, we write

$$\mu_j^{n+1} = \alpha \mu_{j-1}^n + (1 - \alpha) \mu_j^n \quad (7.208)$$

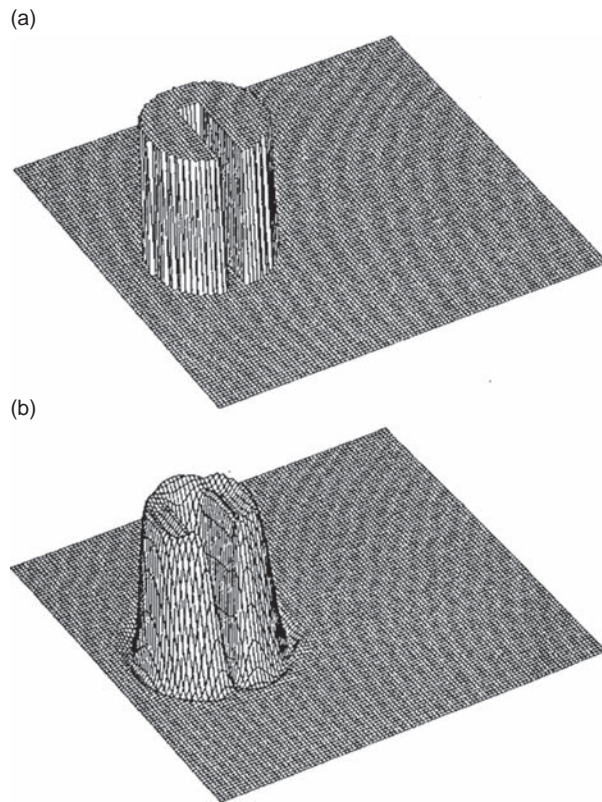
More generally, if the Courant number  $\alpha$  has a *ceiling* (smallest larger integer) of  $L$ , such that  $\alpha' = L - \alpha$  is positive and smaller than unity, then the departure point  $D$  is located in the grid cell  $[j - L, j - L + 1]$ . The interpolation formula (7.208) then becomes:

$$\mu_j^{n+1} = \alpha' \mu_{j-L}^n + (1 - \alpha') \mu_{j-L+1}^n \quad (7.209)$$

Numerical approximation (7.209) is equivalent to expression (7.75) that describes the notoriously dissipative Eulerian upstream method. Thus, if one uses a linear interpolation as implemented here, the semi-Lagrangian method is excessively diffusive. By using higher-order interpolation schemes, the intensity of the diffusion can be considerably reduced. Cubic spline functions (Bermejo, 1990) or biquadratic polynomials (Lauritzen *et al.*, 2010) are often adopted. Williamson and Rasch (1989) and Rasch and Williamson (1990b) examine several possible interpolators and assess their ability to preserve the shape of the advected fields. Accurate interpolation schemes add to the computational costs of the method.

Non-interpolating semi-Lagrangian schemes have also been developed (see, e.g., Ritchie, 1986). In this case, the vector that defines the back-trajectory is decomposed in the sum of a vector that reaches the grid point  $G$  closest to the departure point  $D$  and a residual vector pointing from this grid point to the departure point  $D$ . To determine the value of the transported function at grid point  $G$ , no interpolation is needed. The transport along the second vector is performed using a classic Eulerian method, for which the Courant number is always smaller than 1. The overall advection for a given time step is thus always stable, but the advection for the second substep has the dispersive/diffusive properties of the Eulerian scheme that is adopted.

As in the case of the pure Lagrangian methods, SLT algorithms are stable for relatively large time steps. The stability condition is provided by the Lipschitz criterion (trajectories may not cross each other), which is considerably less severe than the CFL condition. To illustrate the performance of the semi-Lagrangian



**Figure 7.27** Advection (solid-body rotation) of a “slotted” cylinder using a semi-Lagrangian method with a constant uniform angular velocity about the center of the domain. The adopted Courant number is 4.2. A cubic-spline interpolator is adopted: (a) shows the initial condition and (b) the shape of the cylinder after six revolutions. From Staniforth and Côté (1991). Copyright © American Meteorological Society, used with permission.

method, Figure 7.27 shows the evolution of a “slotted” cylinder after six revolutions of solid-body rotation at uniform angular velocity about the center of the domain (Staniforth and Côté, 1991). The adopted Courant number of 4.2 is considerably larger than required for the stability of Eulerian schemes. For the same accuracy, the SLT method is more computationally efficient than Eulerian methods.

A major disadvantage of SLT grid point schemes is that they do not conserve mass. Numerical adjustment is necessary and different methods can be used for this purpose (Rasch and Williamson, 1990b; see also Section 7.9). The simplest is to compare the total mass  $M_i(t_{n+1})$  of tracer  $i$  at time  $t_{n+1}$  over the model domain to the total mass  $M_i(t_n)$  at time  $t_n$ , and apply a uniform multiplicative correction  $M_i(t_n)/M_i(t_{n+1})$  to the mixing ratios at  $t_{n+1}$ .

## 7.8.2 Finite Volume Based SLT Schemes

To avoid the artificial mass correction process required by SLT grid point schemes, conservative finite volume SLT schemes have been developed (Nair and

Machenhauer, 2002; Nair *et al.* 2003; Zerroukat *et al.*, 2002, 2007; Lauritzen *et al.*, 2006). Rather than transporting the tracer mixing ratio at specific points in the model domain, these schemes advect variable finite volume elements that contain a specified mass of the tracer. By this process, the total mass (or the averaged mass density) of the tracer in the “departure” volume is equal to its mass in the “arrival” volume, ensuring mass conservation. In a 2-D formulation, the conservation of the total mass inside a Lagrangian grid cell of area  $A$  that moves and is distorted with the fluid motion is expressed as:

$$\frac{\partial}{\partial t} \int_{A(t)} \rho dA = 0 \quad \text{or equivalently} \quad \int_{A(t)} \rho dA = \int_{A(t+\Delta t)} \rho dA \quad (7.210)$$

After discretization in time, we have

$$\langle \rho_j^{n+1} \rangle A_j^{n+1} = \langle \rho_j^n \rangle A_j^n \quad (7.211)$$

where  $A_j^n$  and  $A_j^{n+1}$  are the surface areas of the cell  $j$  at the departure and arrival time levels, respectively, and

$$\langle \rho_j \rangle = \frac{1}{A_j} \int_{A_j} \rho_j dA \quad (7.212)$$

is the mean density in grid cell  $j$ . The surface  $A_j^{n+1}$  at time  $t_{n+1}$  coincides with a Eulerian grid cell of the model. The value of the mean density  $\langle \rho_j^n \rangle$  in the departure area  $A_j^n$  is derived by interpolation of the solution at time  $t_n$ . The solution at time  $t_{n+1}$  is obtained by a remapping process. The method is illustrated in Figure 7.28 in a 1-D configuration. In this simple case, one defines the mean density of a tracer in a grid cell of size  $\Delta x$ :

$$\langle \rho_j \rangle = \frac{1}{\Delta x_j} \int_{\Delta x_j} \rho_j(x) dx \quad (7.213)$$

The solution at the arrival time  $t_{n+1}$  is

$$\langle \rho_j^{n+1} \rangle = \frac{\langle \rho_j^n \rangle \Delta x_j^D}{\Delta x_j} \quad (7.214)$$

If the wind velocity  $u$  varies as a function of space and time, the size of the gridbox (noted  $\Delta x$  at the arrival location) varies during the back-trajectory step and becomes  $\Delta x^D \neq \Delta x$  at the departure location.

Flux-form finite volume SLT schemes are conservative for tracer transport if the winds originate from a general circulation model (GCM) where the exact same scheme was applied to solve the Navier–Stokes equation for momentum. This requirement can be achieved in online GCM simulations of chemical tracers, but is generally not achievable in offline chemical transport models (CTMs) (Jöckel *et al.*, 2001; Horowitz *et al.*, 2003). This problem in offline CTMs often results from inconsistencies between the advection scheme and the surface pressure tendency provided by the dynamical model. A mass fixer is often applied to alleviate this

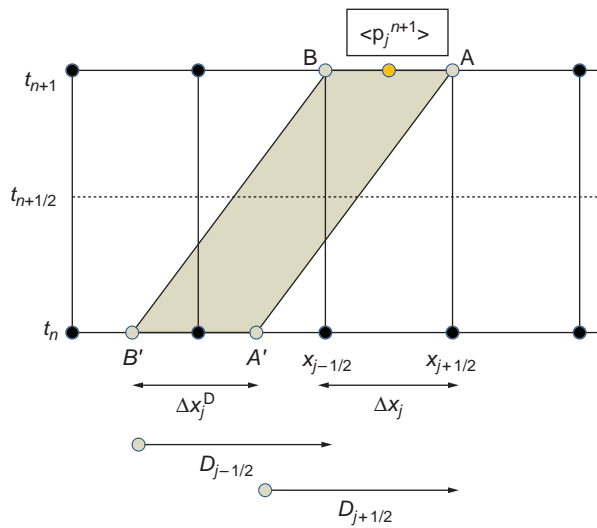


Figure 7.28

Schematic representation of the 1-D finite volume mass conserving Lagrangian method of Lin and Rood (1996). The “volume”  $\Delta x$  at the arrival time  $t_{n+1}$  is represented by the grid cell  $AB$ , and its value  $\Delta x^D$  at departure time  $t_n$  is represented by the distance  $A'B'$ . If the wind velocity  $u$  is not constant in space, the displacements at the cell interfaces  $D_{j+1/2}$  and  $D_{j-1/2}$  for points  $A'$  and  $B'$  are not equal and  $\Delta x$  is different from  $\Delta x^D$  (Lin and Rood, 1996). Copyright © American Meteorological Society, used with permission.

problem (Horowitz *et al.*, 2003), but may violate monotonicity requirements and generate non-physical transport.

## 7.9 Spectral, Finite Element, and Spectral Element Methods

The *spectral method* is widely used to solve the dynamical equations in GCMs. In this approach, the fields such as the temperature or the winds are represented by a series of continuous basis functions such as spherical harmonics in the horizontal plane (see Chapter 4) and by a finite difference formulation in the vertical direction. Spectral methods have also been used in these models to represent the advection of moisture. To illustrate the methodology, we consider again the 1-D case in the  $x$ -direction and we approximate the solution of the advection equation by the expansion

$$\psi(x, t) \approx \sum_{k=1}^K \alpha_k(t) \Phi_k(x) \quad (7.215)$$

in which  $\Phi_k(x)$  represents a set of orthogonal functions (e.g., elementary trigonometric functions) and  $\alpha_k$  are unknown coefficients that depend on time  $t$ . Expression (7.215) is introduced in the advection equation (7.17), leading to a system of  $K$  differential equations for the coefficients  $\alpha_k(t)$  that can be solved by standard

methods. When spherical geometry is used, the solution is instead expressed as a function of spherical harmonics (see equation (4.252)).

With the proper choice of parameters, the spectral method provides accurate and stable results and can be mass-conservative. However, it is not shape-preserving (monotonic, positive definite); overshoots and undershoots can be produced (see Box 4.7). Negative values of the transported fields can be eliminated by using appropriate filtering and filling schemes, but these corrupt the correlations between tracers which is critical for nonlinear chemistry. In addition, the algorithm does not satisfy the criterion of locality. Because of these limitations, spectral methods are generally not well suited for chemical applications.

In the *finite element method*, the solution  $\psi(x, t)$  is also provided by expansion (7.215), but with the basis functions  $\Phi_k(x)$  defined only on a small region of space  $[A, B]$  called finite elements (see Section 4.10). In the Galerkin approach, the coefficients  $a_k$  are derived by requiring that the error arising from representing function  $\psi(x, t)$  by expansion (7.215),

$$e_K = \frac{\partial}{\partial t} \sum_{k=1}^K a_k(t) \Phi_k(x) + c \frac{\partial}{\partial x} \sum_{k=1}^K a_k(t) \Phi_k(x) \quad (7.216)$$

be orthogonal to the basis functions. This condition is expressed by the integral over the domain  $[A, B]$ :

$$\int_A^B e_K \Phi_i(x) dx = 0 \quad (7.217)$$

for all values of  $i \in [0, K]$ . If in (7.217) one replaces the error  $e_K$  by expression (7.216), one obtains the system of  $K$ -coupled ODEs:

$$\sum_{k=1}^K \frac{\partial a_k(t)}{\partial t} \int_A^B \Phi_i(x) \Phi_k(x) dx + c \sum_{k=1}^K a_k \int_A^B \Phi_i(x) \frac{\partial \Phi_k(x)}{\partial x} dx = 0 \quad (i = 1, \dots, K) \quad (7.218)$$

which is solved to obtain the coefficients  $a_k(t)$ . For this purpose, the time derivatives are usually approximated by finite differences.

The *spectral element method* (Patera, 1984) is a finite element technique in which a high-degree spectral method is applied within each element. As discussed by Nair *et al.* (2011), the spectral element method combines the geometric flexibility of the traditional finite element methods with the high accuracy, rapid convergence, and weak numerical dispersion and dissipation of the classical spectral methods. It is not inherently conservative, but can be engineered to ensure a user-required level of mass conservation.

Although finite element and spectral element methods have so far mainly been used for engineering applications, they are now emerging as promising methods for atmospheric problems. Their local domain decomposition property makes them particularly well suited for massively parallel computer architectures, and they can be easily applied when the model domain is geometrically complex or when grid refinement is needed in specified atmospheric regions.

## 7.10 Numerical Fixers and Filters

An important requirement for the numerical method applied to solve the advection equation is that the solution be positive and that mass be conserved. This is not always the case, and if so one can apply a-posteriori corrections to restore these desired properties. Jablonowski and Williamson (2011) provide an extensive review of the use of fixers and filters in atmospheric models.

**Fixers.** Negative values in the transported variables (e.g., tracer mixing ratio) can be eliminated by introducing “*numerical fixers*” that borrow mass from surrounding (or downstream) grid points. Rasch and Williamson (1990a) describe a possible implementation of local and global fixers. When a negative value is encountered during a point-by-point scan of the calculated quantities on each horizontal surface, the immediate neighboring points are examined, and if sufficient mass is available to “fill the hole,” the negative value is set to zero and the values at the neighboring points are reduced proportionally. If there is not sufficient mass available, no action is taken. The application of this local *filling method* does conserve mass, but may not eliminate all negative values. In a second step, a global filter is applied in which the remaining negative values are set to zero; this violates mass conservation but a renormalization can be applied by scaling to the global masses in the domain, as described in Section 7.8.1 to enforce mass conservation in SLT schemes. This filling process produces diffusion and does not ensure monotonicity of the corrected fields; it can also be computationally expensive.

**Filters.** In several of the algorithms described previously, short waves may grow excessively in the solution of the advection equation, producing undesired noise and even catastrophic instability. These waves can be eliminated by appropriate *smoothing* or *filtering*. One option is to add a small diffusive term in the advection equation, which will smooth the solution and suppress high wavenumbers. In spectral models, numerical noise can easily be suppressed by omitting wavenumbers larger than a specified value and highlighting only the scales of interest. Spectral filtering can also be used in grid point models by applying a Fourier transform to the solution, which eliminates the high frequencies in the signal, and applying an inverse Fourier transform. Such filtering is often applied in polar regions where the meridians converge and a longitudinal grid point spacing becomes so small that, without filtering, the solution would become unstable. Finally, numerical filters such as the linear Shapiro filter presented in Section 4.15.2 are often applied to eliminate two-grid interval waves completely while having little effect on longer waves (Shapiro, 1971). Other high-frequency filters have been developed by Asselin (1972) and Forester (1977).

## 7.11 Concluding Remarks

In this chapter we have reviewed different numerical algorithms used to approximate the solution of the linear advection equation. No existing method fully addresses modelers’ requirements. The examination of simple numerical schemes reveals that



high-order algorithms are generally not monotonic and occasionally produce undesired negative values. Low-order algorithms such as the upstream method preserve the sign of the solution, but are excessively diffusive. Thus, practical applications must adopt more elaborate schemes that address some of the drawbacks that characterize the simple methods. Modern schemes are often upstream-based Eulerian finite volume methods that are mass conservative, positive definite, and possess good phase-error characteristics. They may use adaptive time steps to meet CFL stability requirements or semi-Lagrangian options to get around these requirements. Specific, often complex nonlinear algorithms are developed to reduce the numerical diffusion that characterizes upstream methods. These complex schemes can yield significant improvement in accuracy, but often with enhanced computational costs.

Finite volume Eulerian methods, in which a subgrid distribution of the transported quantity is specified, provide highly accurate solutions and are popular in global CTMs. In many respects, they are superior to classic grid point methods. Computational cost depends on the user tolerance for numerical diffusion. The Prather scheme is regarded as a reference among Eulerian models. It produces accurate solutions with little diffusion. However, it has large computing time and storage requirements. A van Leer scheme may enable higher grid resolution, compensating for the lower accuracy.

Lagrangian methods are popular for source-oriented and receptor-oriented transport problems in which one is concerned with transport from a point source or transport contributing to concentrations at a receptor point. However, they do not provide the regular full-domain solution achievable by Eulerian methods and cannot properly represent nonlinear chemistry or aerosol microphysics. Semi-Lagrangian methods are very popular in global CTMs because their numerical stability is not as severely constrained by choice of time step as in the case of Eulerian schemes. They are sometimes used as a back-up scheme in cases where the regular Eulerian solver violates the CFL criterion.

In summary, there is no single advection scheme that is universally best. The choice of scheme depends on the type of problem being solved, the tolerance for different kinds of error, and the computational demands. For any scheme, it is important to verify that basic criteria of stability and mass conservation are met. The material in this chapter should enable readers to understand the issues associated with different advection schemes and to make informed choices in selecting appropriate schemes for their applications.

## References

- Allen D. J., Douglass A. R., Rood R. B., and Guthrie P. D. (1991) Application of a monotonic upstream-biased transport scheme to three-dimensional constituent transport calculations, *Mon. Wea. Rev.*, **119**, 2456–2464.
- Asselin R. (1972) Frequency filter for time integrations, *Mon. Wea. Rev.*, **100**, 487–490.
- Bermejo R. (1990) Notes and correspondence on the equivalence of semi-Lagrangian schemes and particle-in-cell finite element methods, *Mon. Wea. Rev.*, **118**, 979–987.

- Boris J. P. and Book D. L. (1973) Flux-corrected transport. I. SHASTA, a fluid transport algorithm that works, *J. Comput. Phys.*, **11**, 38–69.
- Boris J. P. and Book D. L. (1976) Flux-corrected transport. III. Minimal-error FCT algorithms, *J. Comput. Phys.*, **20**, 397–431.
- Bott A. (1989a) A positive definite advection scheme obtained by non-linear renormalization of the advective fluxes, *Mon. Wea. Rev.*, **117**, 1006–1015.
- Bott A. (1989b) Reply, *Mon. Wea. Rev.*, **117**, 2633–2636.
- Carpenter R. L., Droegemeier K. K., Woodward P. R., and Hane C. E. (1990) Application of the piecewise parabolic method (PPM) to meteorological modeling, *Mon. Wea. Rev.*, **118**, 586–612.
- Chen Y. and Falconer R. A. (1992) Advection–diffusion modelling using the modified QUICK scheme, *Int. J. Numer. Meth. Fluids*, **15**, 1171–1196.
- Chlond A. (1994) Locally modified version of Bott’s advection scheme, *Mon. Wea. Rev.*, **122**, 111–125.
- Colella P. and Sekora M. D. (2008) A limiter for PPM that preserves accuracy at smooth extrema, *J. Comput. Phys.*, **227** (15), 7069–7076.
- Colella P. and Woodward P. R. (1984) The piecewise parabolic method (PPM) for gasdynamical simulations, *J. Comput. Phys.*, **54**, 174–201.
- Courant R., Friedrichs K., and Lewy H. (1928) Über die partiellen Differenzengleichungen der mathematischen Physik, *Math. Ann.*, **100**, 32–74.
- Courant R., Isaacson E., and Rees M. (1952) On the solution of nonlinear hyperbolic differential equations by finite difference, *Commun. Pure Appl. Math.*, **5**, 243–255.
- Crowley W. P. (1968) Numerical advection experiments, *Mon. Wea. Rev.*, **96**, 1–11.
- Davies H. C. (1983) Limitations of some common lateral boundary schemes used in regional NWP models, *Mon. Wea. Rev.*, **111**, 1002–1012.
- Dukowicz J. K. and Baumgardner J. R. (2000) Incremental remapping as a transport/advection algorithm, *J. Comput. Phys.*, **160**, 318–335.
- Dullemond C. P. (2009) *Numerical Fluid Dynamics*, Lecture Notes, Zentrum für Astronomie, Ruprecht-Karls Universität, Heidelberg.
- Farrow D. E. and Stevens D. P. (1994) A new tracer advection scheme for Bryan Cox type ocean general circulation models, *J. Phys. Oceanogr.*, **25**, 1731–1741.
- Forester C. K. (1977) Higher order monotonic convective difference schemes, *J. Comput. Phys.*, **23**, 1–22.
- Godunov S. K. (1959) A finite difference method for the computation of discontinuous solutions of the equations of fluid dynamics, *Mat. Sb.*, **47**, 357–393.
- Gross E. S., Koseff J. R., and Monismith S. G. (1999) Evaluation of advective schemes for estuarine salinity simulations, *J. Hydr. Engrg., ASCE*, **125** (1), 32–46.
- Horowitz L. W., Walters S., Mauzerall D. L., *et al.* (2003) A global simulation of tropospheric ozone and related tracers: Description and evaluation of MOZART, version 2, *J. Geophys. Res.*, **108** (D24), 4784, doi: 10.1029/2002JD002853.
- Iserles A. (1986) Generalised leapfrog methods, *IMA J. Numer. Anal.*, **6**, 381–392.
- Jablonowski C. and Williamson D. L. (2011) The pros and cons of diffusion, filters and fixers in atmospheric general circulation models, In *Numerical Techniques for*



- Global Atmospheric Models, Lecture Notes in Computational Science and Engineering* (Lauritzen P. H., Jablonowski C., Taylor M. A., and Nair R. D., eds.), Springer, Berlin.
- Jöckel P., von Kuhlmann R., Lawrence M. G., *et al.* (2001) On a fundamental problem in implementing flux-form advection schemes for tracer transport in 3-dimensional general circulation and chemistry transport models, *Q. J. R. Meteorol. Soc.*, **127**, 1035–1052.
- Kida H. (1983) General circulation of air parcels and transport characteristics derived from a hemispheric GCM: Part 1. A determination of advective mass flow in the lower stratosphere, *J. Meteor. Soc. Japan*, **61**, 171–187.
- Kim C. (2003) Accurate multi-level schemes for advection, *Int. J. Numer. Meth. Fluids*, **41**, 471–494, doi: 10.1002/flid.443.
- Kurihara Y. (1965) On the use of implicit and iterative methods for the time integration of the wave equation, *Mon. Wea. Rev.*, **93**, 33–46.
- Lauritzen P. H., Kaas E., and Machenhauer B. (2006) A mass-conservative semi-implicit semi-Lagrangian limited-area shallow-water model on a sphere, *Mon. Wea. Rev.*, **134**, 2588–2606.
- Lauritzen P. H., Nair R. D., and Ullrich P. A. (2010) A conservative semi-Lagrangian multi-tracer transport scheme (CSLAM) on the cubed-sphere grid, *J. Comput. Phys.*, **229** (5), doi: 10.1016/j.jcp.2009.10.036
- Lauritzen P. H., Ullrich P. A., and Nair R. D. (2011) Atmospheric transport schemes: Desirable properties and a semi-Lagrangian view on finite-volume discretizations. In *Numerical Techniques for Global Atmospheric Models, Lecture Notes in Computational Science and Engineering* (Lauritzen P. H., Jablonowski C., Taylor M. A., and Nair R. D., eds.), Springer, Berlin.
- Lax P. D. (1954) Weak solutions of nonlinear equations and their numerical computation, *Comm. Pure Appl. Math.*, **7**, 159–193.
- Lax P. D. and Wendroff B. (1960) Systems of conservation laws, *Comm. Pure and Appl. Math.*, **13**, 217–237.
- Lax P. D. and Wendroff B. (1964) Difference schemes for hyperbolic equations with high order of accuracy, *Comm. Pure Appl. Math.*, **17** (3), 281–398.
- Leith C. E. (1965) Numerical simulations of the earth's atmosphere, *Meth. Comput. Phys.*, **4**, 1–28.
- Leonard B. P. (1979) A stable and accurate convective modelling procedure based on quadratic upstream interpolation, *Comput. Methods in Appl. Mech. Eng.*, **19**, 59–98.
- Lin J. C. (2012) Lagrangian modeling of the atmosphere: An introduction. In *Lagrangian Modeling of the Atmosphere* (Lin J., Brunner D., Gerbig C., *et al.*, eds.), American Meteorological Union, Washington, DC.
- Lin S. J. and Rood R. B. (1996) Multidimensional flux-form semi-Lagrangian transport schemes, *Mon. Wea. Rev.*, **124**, 2046–2070.
- Lipscomb W. H. and Ringler T. D. (2005) An incremental remapping transport scheme on a spherical geodesic grid, *Mon. Wea. Rev.*, **133**, 2335–2350.
- Miura H. (2007) An upwind-biased conservative advection scheme for spherical hexagonal–pentagonal grids, *Mon. Wea. Rev.*, **135**, 4038–4044.

- Müller R. (1992), The performance of classical versus modern finite-volume advection schemes for atmospheric modeling in a one-dimensional test-bed, *Mon. Wea. Rev.*, **120**, 1407–1415.
- Nair R. D., and Machenhauer B. (2002) The mass-conservative cell-integrated semi-Lagrangian advection scheme on the sphere, *Mon. Wea. Rev.*, **130**, 647–667.
- Nair R. D., Scroggs J. S., and Semazzi F. H. M. (2003) A forward-trajectory global semi-Lagrangian transport scheme, *J. Comput. Phys.*, **193**, 275–294.
- Nair R. D., Levy M., and Lauritzen P. H. (2011) Emerging methods for conservation laws. In *Numerical Techniques for Global Atmospheric Models, Lecture Notes in Computational Science and Engineering* (Lauritzen P. H., Jablonowski C., Taylor M. A., and Nair R. D., eds.), Springer, Berlin.
- Papayannis A., Mamouri R. E., Amiridis V., *et al.* (2012) Optical properties and vertical extension of aged ash layers over the Eastern Mediterranean as observed by Raman lidars during the Eyjafjallajökull eruption in May 2010, *Atmos. Env.*, **48**, 56–65.
- Patera A. T. (1984) A spectral element method for fluid dynamics: Laminar flow in a channel expansion, *J. Comput. Phys.*, **54**, 468–488.
- Prather M. J. (1986) Numerical advection by conservation of second order moments, *J. Geophys. Res.*, **91**, 6671–6681.
- Press W. H., Teukolsky S. A., Vetterling W. T., and Flannery B. P. (2007) *Numerical Recipes: The Art of Scientific Computing*, 3rd edition, Cambridge University Press, Cambridge.
- Rasch P. J. and Williamson D. L. (1990a) Computational aspects of moisture transport in global models of the atmosphere, *Quart. J. Roy. Meteor. Soc.*, **116**, 1071–1090.
- Rasch P. J. and Williamson D. L. (1990b) On shape-preserving interpolation and semi-Lagrangian transport, *SIAM J. Sci. Stat. Comput.*, **11**, 656–687.
- Ritchie H. (1986) Eliminating the interpolation associated with the semi-Lagrangian scheme, *Mon. Wea. Rev.*, **114**, 135–146.
- Roe P. L. (1986) Characteristic-based schemes for the Euler equations, *Ann. Rev. Fluid Mech.*, **18**, 337–365.
- Russell G. L. and Lerner J. A. (1981) A new finite-differencing scheme for the tracer transport equation, *J. Appl. Meteor.*, **20**, 1483–1498.
- Seibert P. and Morariu B. (1991) Improvements of upstream, semi-Lagrangian numerical advection schemes, *J. Appl. Meteor.*, **30**, 117–125.
- Shapiro R. (1971) The use of linear filtering as a parameterization of atmospheric motion, *J. Atmos. Sci.*, **28**, 523–531.
- Skamarock W. and Menchaca M. (2010) Conservative transport schemes for spherical geodesic grids: High-order reconstruction for forward-in-time schemes, *Mon. Wea. Rev.*, **138**, 4497–4508.
- Slingerland R. and Kump L. (2011) *Mathematical Modeling of Earth's Dynamical Systems: A Primer*, Princeton University Press, Princeton, NJ.
- Smolarkiewicz P. K. (1983) A simple positive definite advection scheme with small implicit diffusion, *Mon. Wea. Rev.*, **111**, 479–486.
- Smolarkiewicz P. K. (1984) A fully multidimensional positive definite advection transport algorithm with small implicit diffusion, *J. Comput. Phys.*, **54**, 325–362.

- Smolarkiewicz P. K. (2006) Multidimensional positive definite advection transport algorithm: An overview, *Int. J. Numer. Meth. Fluids*, **50** (10), 1123–1144.
- Spalding D. B. (1972) A novel finite difference formulation for differential expressions involving both first and second derivatives, *Int. J. Numer. Meth. Eng.*, **4**, 551–559.
- Staniforth A. and Côté J. (1991) Semi-Lagrangian integration schemes for atmospheric models: A review, *Mon. Wea. Rev.*, **119**, 2206–2223.
- Stohl A. (1998) Computation, accuracy and applications of trajectories: A review and bibliography, *Atmos. Environ.*, **32** (6), 947–966.
- Stohl A., Hittenberger M., and Wotawa G. (1998) Validation of the Lagrangian particle dispersion model FLEXPART against large scale tracer experiment data, *Atmos. Environ.*, **32**, 4245–4262.
- Tremback C. J., Powell J., Cotton W. R., and Pielke R. A. (1987), The forward-in-time upstream advection scheme: Extension to higher order, *Mon. Wea. Rev.*, **115**, 894–902.
- van Leer B. (1977) Toward the ultimate conservative difference scheme IV: A new approach to numerical convection, *J. Comp. Phys.*, **23**, 276–299.
- van Leer B. (1979) Toward the ultimate conservative difference scheme V: A second order sequel to Godunov's method, *J. Comp. Phys.*, **32**, 101–136.
- Warming R. F. and Beam R. M. (1976) Upwind second-order difference schemes and applications in aerodynamic flows, *AIAA Journal*, **14** (9), 1241–1249.
- Williamson D. L. (1992) Review of the numerical approaches for modeling global transport. In *Air Pollution Modelling and its Applications* (Dop H. V. and Kallow G., eds.), Plenum Press, New York.
- Williamson D. L. and Rasch P. J. (1989) Two-dimensional semi-Lagrangian transport with shape preserving interpolation, *Mon. Wea. Rev.*, **117**, 102–129.
- Zerroukat M., Wood N., and Staniforth A. (2002) SLICE: A semi-Lagrangian inherently conserving and efficient scheme for transport problems, *Q. J. R. Meteorol. Soc.*, **128**, 2801–2820.
- Zerroukat M., Wood N., and Staniforth A. (2007) Application of the Parabolic Spline Method (PSM) to a multi-dimensional conservative transport scheme (SLICE), *J. Comput. Phys.*, **225**, 935–948.