Chapter 3. Finite Difference Methods for Hyperbolic Equations

1. Introduction

Most hyperbolic problems involve the transport of fluid properties. In the equations of motion, the term describing the transport process is often called convection or advection.

E.g., the 1-D equation of motion is

$$\frac{du}{dt}=\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}=-\frac{1}{r}\frac{\partial p}{\partial x}+v\nabla^2 u\ . \qquad (1)$$

Here the advection term $u\frac{\partial u}{\partial x}$ term is nonlinear.

We will focus first on linear advection problem, and move to nonlinear problems later.

From (1), we can see the transport process can be expressed in the Lagrangian form (in which the change of momentum u along a particle, du/dt, is used) and the Eulerian form. With the former, advection term does not explicitly appear. Later in this course, we will also discuss semi-Lagrangian method for solving the transport problems. In this chapter, we discuss only the Eulerian advection equation.

2. Linear convection – 1-D wave equation

2.1. The wave equations

The classical 2nd-order hyperbolic wave equation is

$$\frac{\partial^2 u}{\partial t^2}=c^2\frac{\partial^2 u}{\partial x^2}\ . \qquad (2)$$

The equation describes wave propagation at a speed of c in two directions.

The 1st-order equation that has properties similar to (2) is

$$\frac{\partial u}{\partial t}+c\frac{\partial u}{\partial x}=0\ , \qquad c>0. \qquad (3)$$

Note that Eq.(2) can be obtained from Eq.(3), by taking a time derivative of (3) and resubstituting (3) into the new equation.

For a pure initial value problem with initial condition

u(x, 0) = F(x), $-\infty < x < \infty$,

the exact solution to (3) is u(x,t) = F(x-ct), which we have obtained earlier using the method of characteristics. We know that the solution represents a signal propagating at speed c.

2.2. Centered in time and space (CTCS) FD scheme for 1-D wave equation

We apply the centered in time and space (CTCS) scheme to Eq.(2):

$$\frac{u_i^{n+1} - 2u_i^n + u_i^{n-1}}{\Delta t^2} - c^2 \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2} = 0 . \qquad (4)$$

We find for this scheme,

$$t = O(\Delta t^2 + \Delta x^2) .$$

Performing von Neumann stability analysis, we can obtain a quadratic equation for amplification factor λ:

$$l_{\pm} = 1 - 2p^2 \sin^2\left(\frac{k\Delta x}{2}\right) \pm 2p \sin\left(\frac{k\Delta x}{2}\right)\left[p^2 \sin^2\left(\frac{k\Delta x}{2}\right) - 1\right]^{1/2}$$

where

$$p = \frac{c\Delta t}{\Delta x}$$

which is the fraction of zone distance moved in Δt at speed c.

Let $q = p \sin\left(\frac{k\Delta x}{2}\right)$, we have

$$l_{\pm} = 1 - 2q^2 \pm 2q[q^2 - 1]^{1/2} .$$

We want to see under what condition, if any, $| l_{\pm} | \leq 1$.

We consider two possible cases.

Case I: If θ ≤1, then λ is complex:

$$l_{\pm} = 1 - 2q^{2} \pm i2q[1-q^{2}]^{1/2}$$

→ $$|\, l_{\pm} |^{2} = (1-2q^{2})^{2} + 4q^{2}[1-q^{2}] = 1$$

Therefore, when θ ≤ 1, the amplification factor is always 1, which is what we want to pure advection!

$$\theta \le 1 \rightarrow p^{2} \sin^{2}\left(\frac{k\Delta x}{2}\right) \le 1$$

We want the above to be true for all k, therefore $p^2 \le 1$ has to be satisfied for all value of $\sin^2()$.

$$p^{2} \le 1 \rightarrow p = \frac{c\Delta t}{\Delta x} \le 1,$$

which is the same as the condition we obtained earlier using energy method for FTUS scheme.

Case II:

If θ ≥ 1, λ is real:

$$l_{\pm} = 1 - 2q^{2} \pm 2q[q^{2}-1]^{1/2} \rightarrow$$

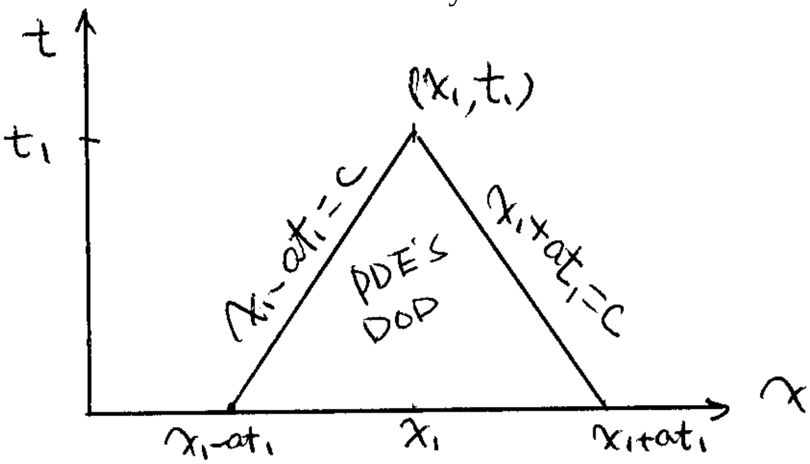
$$|\, l_{\pm} |^{2} = (1-2q^{2} \pm 2q[q^{2}-1]^{1/2})^{2},$$

you can show for yourself that $|\, l_{\pm} | > 1$ therefore the scheme is unstable.

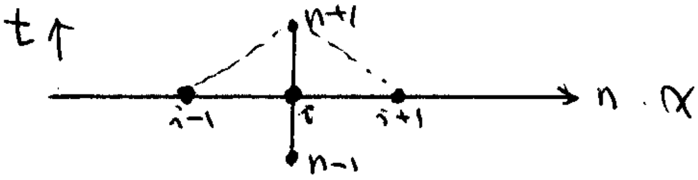
2.3. Courant-Friedrichs-Lewy (CFL) Stability Criterion

Let's consider the stability condition obtained above using the concept of domain of dependence.

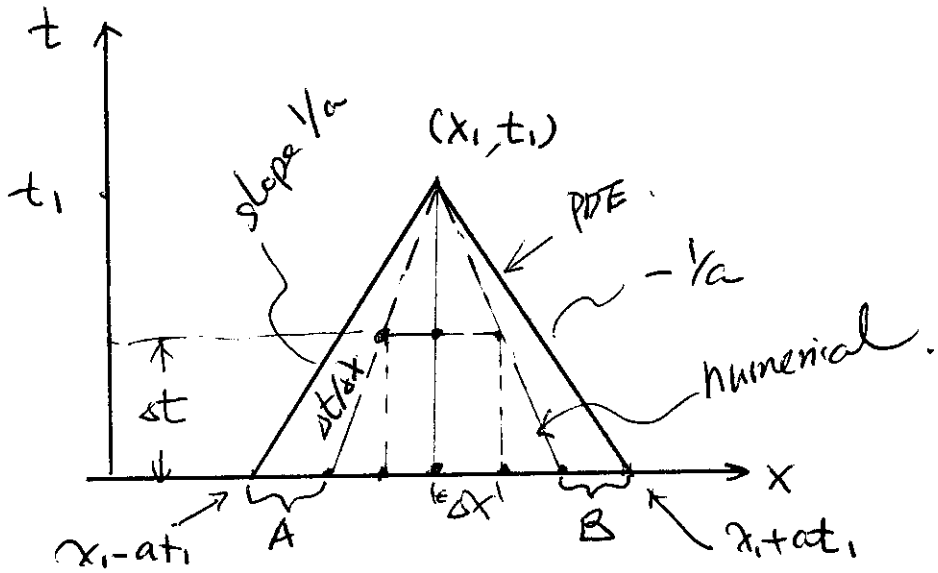
Recall from earlier discussion, the solution at (x_1, t_1) depends on data in the interval [$x_1 - at_1$, $x_1 + at_1$], and the D.O.D. is the area enclosed by the two characteristics lines.



Based on the following discretization stencil,



we can construct a numerical domain of dependence below:



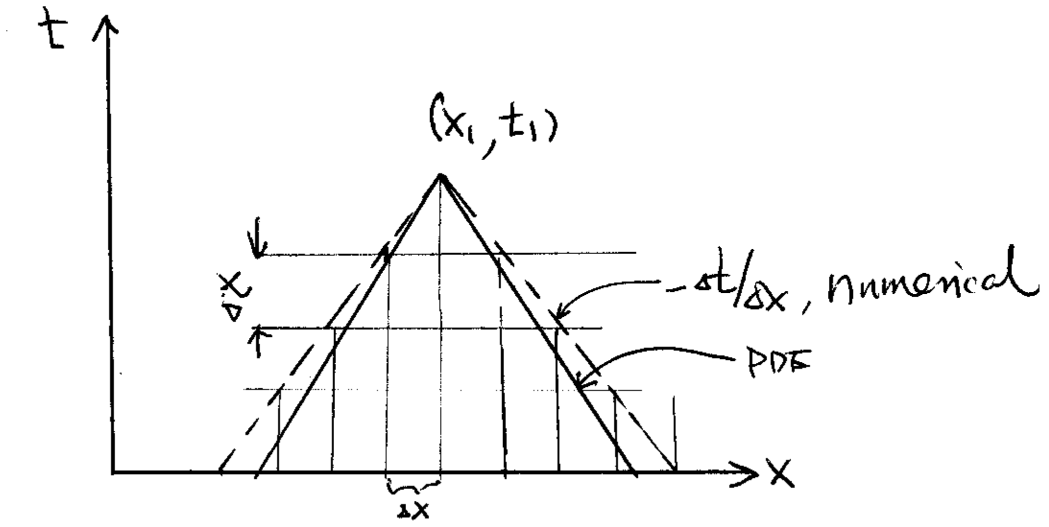
Case I: When the numerical DOD is smaller than the PDE's DOD (which usually happens when Δt is large), the numerical solution cannot be expected to converge to the true solution, because the numerical solution is not using part of the initial condition, e.g., the initial values in the intervals of A and B. The true solution, however, is definitely dependent on the initial values in these intervals. Different initial values there will result in different true solutions, while the numerical solution remain unaffected by their values. We therefore cannot expect the solutions to match.

The numerical solution must then be unstable. Otherwise, the Lax's Equivalence theorem is violated.

The above situation occurs when $\Delta t / \Delta x > 1/ c$ → unstable solution. This agrees with the result of our stability analysis.

Case II: When $\Delta t / \Delta x = 1/c$, the PDE DOD coincides with the numerical DOD, the scheme is stable.

Case III: When $\Delta t / \Delta x < 1/c$, the PDE DOD is contained within the numerical DOD:



the numerical solution now fully depends on the initial condition. It is possible for the scheme to be stable. In the case of CTCS scheme, it is indeed stable.

Definition: $\frac{c\Delta t}{\Delta x} = \sigma =$ Courant number

The condition that $\sigma \leq 1$ for stability is known as the Courant-Friedrichs-Lewy (CFL) stability criterion.

The CFL condition requires that the numerical domain of dependence of a finite difference scheme include the domain of dependence of the associated partial differential equation.

Satisfaction of the CFL condition is a necessary, not a sufficient condition for stability.

E.g., the second-order centered-in-time and fourth-order centered-in-space scheme for a 1-D advection equation requires $\sigma \le 0.728$ for stability whereas the D.O.D condition requires that $\sigma \le 2$.

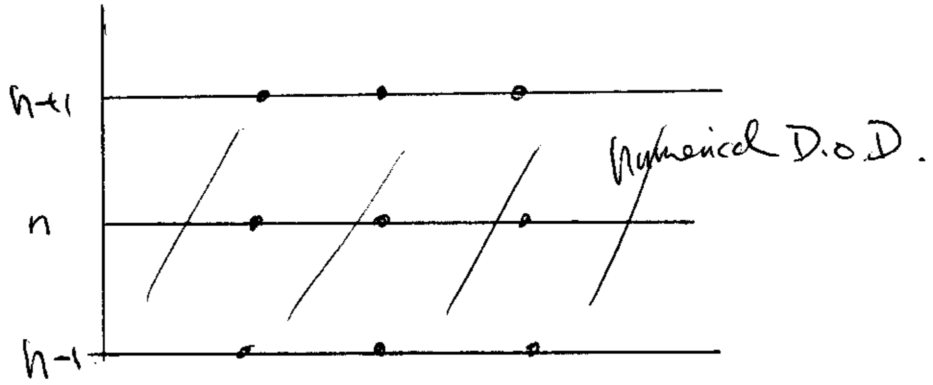
The DOD concept explains why implicit schemes can be unconditionally stable – it is because their numerical DOD always contains the PDE's DOD

e.g., the second-order in time and space implicit scheme for wave equation (2):

$$d_{tt}u^n = \frac{c^2}{4}[d_{xx}u^{n+1} + 2d_{xx}u^n + d_{xx}u^{n-1}].$$

is stable for all σ.

The numerical DOD is:



The numerical DOD covers the PDE's DOD.

Read Durran sections 2.2.2 and 2.2.3, which discuss the CFL criterion using the forward-in-time upstream-in-space (also called upwind) scheme.

3. Phase and Amplitude Errors of 1-D Advection Equation

Reading: Duran section 2.4.2. Tannehill et al section 4.1.2.

The following example F.D. solutions of a 1D advection equation show errors in both the wave amplitude and phase.

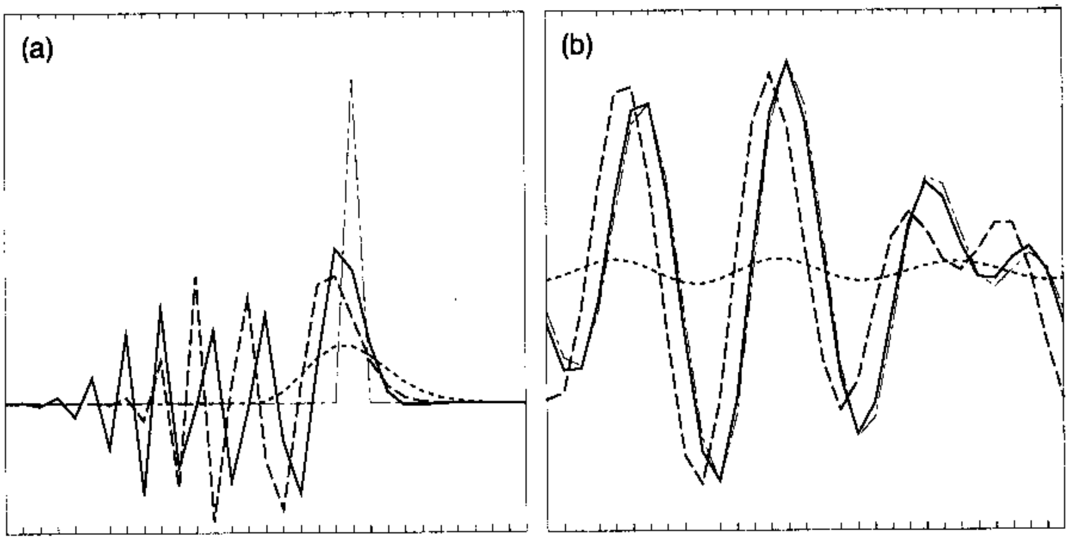


FIGURE 2.13. Exact solution and differential–difference solutions for (a) advection of a spike over a distance of five grid points, and (b) advection of the sum of equal-amplitude $7.5\Delta x$ and $10\Delta x$ sine waves over a distance of twelve grid points. Exact solution (dot-dashed), one-sided first-order (short-dashed), centered second-order (long-dashed), and centered fourth-order (solid). The distribution is translating to the right. Grid-point locations are indicated by the tick marks at the top and bottom of the plot.

In this section, we will examine the truncation errors and try to understand their behaviors.

3.1. Modified equation

The 1D advection equation is

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0 . \qquad (5)$$

Upwind or Donor-Cell Approximation

We have discussed earlier the stability of the forward-in-time upstream-in-space approximation to the 1D advection equation, using the energy method. The FDE is

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + c\frac{u_i^n - u_{i-1}^n}{\Delta x} = 0 \tag{6}$$

Here we assume c>0, therefore the scheme is upstream in space.

We can find from (6) that

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = -\frac{\Delta t}{2}u_{tt} + \frac{c\Delta x}{2}u_{xx} - \frac{(\Delta t)^2}{6}u_{ttt} - \frac{c(\Delta x)^2}{6}u_{xxx} + O(\Delta x^3 + \Delta t^3) \tag{7}$$

and the right hand side is the truncation error.

An analysis of τ can reveal a lot about the expected behavior of the numerical solution, and to investigate, we develop what is known as the Modified Equation. In this method, we write τ so as to illustrate the anticipated error types.

Dispersion Error – occurs when the leading terms in τ have odd-order derivatives. They are characterized by oscillations or small wiggles in the solution, mostly in the wave of a moving wave.

It's called dispersion error because waves of different wavelengths propagate at different speed (i.e., wave speed = a function of k) due to numerical approximations – causing dispersion of waves. For the PDE, all Fourier components described by Eq.(5) should move at the same speed, c.

Dissipation Error – occurs when the leading terms inτ have even-order derivatives. They are characterized by a loss of wave amplitude. The effect is also called artificial viscosity and is implicit in the numerical solution.

The combined effect of dissipation and dispersion is often called diffusion.

To isolate these errors, we derive the Modified Eqution, which is the PDE that is actually solved when a FD scheme is applied to the PDE.

The modified equation is obtained by replacing time derivatives in the truncation error by the spatial derivatives.

Let's do this for Eq.(7).

To replace u_{tt} in right hand side of (7), we perform (Eq.(7) $)_t$ →

$$u_{tt} + cu_{xt} = -\frac{\Delta t}{2}u_{ttt} + \frac{c\,\Delta x}{2}u_{xxt} - \frac{(\Delta t)^2}{6}u_{tttt} - \frac{c(\Delta x)^2}{6}u_{xxxt} + \ldots \qquad (8)$$

and perform – c (Eq(7) $)_x$

$$-cu_{tx} - c^2u_{xx} = \frac{c\Delta t}{2}u_{ttx} - \frac{c^2\Delta x}{2}u_{xxx} + \frac{c(\Delta t)^2}{6}u_{tttx} + \frac{c^2(\Delta x)^2}{6}u_{xxxx} + \ldots \qquad (9)$$

and add (8) and (9) →

$$u_{tt} = c^2u_{xx} + \Delta t\left(\frac{-u_{ttt}}{2} + \frac{c}{2}u_{ttx} + O(\Delta t)\right) + \Delta x\left(\frac{c}{2}u_{xxt} - \frac{c^2}{2}u_{xxx} + O(\Delta x)\right). \qquad (10)$$

Similary, we can obtain other time derivatives, u_{ttt} found in (7) and (10) and u_{ttx} and u_{xxt} found in (10). They are (see Table 4.1 of Dannehill et al):

$$\begin{aligned} u_{ttt} &= -c^3u_{xxx} + O(\Delta x + \Delta t) \\ u_{ttx} &= c^2u_{xxx} + O(\Delta x + \Delta t) \\ u_{xxt} &= -cu_{xxx} + O(\Delta x + \Delta t) \end{aligned} \qquad (11)$$

Combining (7), (10) and (11) →

$$u_t + cu_x = \frac{c\Delta t}{2}\left(1- m\right)u_{xx} - \frac{c(\Delta x)^2}{6}\left(2m^2 - 3m + 1\right)u_{xxx} + O(\Delta x^3, \Delta x^2\Delta t, \Delta t^2\Delta x, \Delta t^3) \qquad (12)$$

where $m = \frac{c\Delta t}{\Delta x}$.

Eq.(12) is the modified equation, which clearly shows the error terms relative to the original PDE.

Note that the leading term has as form of $K\frac{\partial^2 u}{\partial x^2}$ which, for 1- μ > 0, represent the dissipation (or diffusion as we often call it) process and therefore the dominant error is of a dissipation nature.

Note that we had used CTCS scheme δ_{2t} + c δ_{2x}u = 0, then the leading error term in the modified equation would be

$$\frac{c(\Delta x)^2}{6}\left[\frac{c^2(\Delta t)^2}{(\Delta x)^2} - 1\right]\frac{\partial^3 u}{\partial x^3}. \qquad (12)$$

It contains the third (odd) order derivative, and the dominant error is of the dispersive nature.

Returning to the upstream scheme, we find that when μ = c Δt/Δx = 1, the scheme is exact!! The coefficient of the leading error term, $\frac{c\Delta t}{2}\left(1 - m\right)$, is called the artificial viscosity, and when μ ≠ 1, causes server damping of the computational solution (see figure shown earlier). In fact, the Doner-Cell is well known for its strong damping.

3.2. Quantitative Estimate of Phase and Amplitude Errors

Reading: Sections 4.1.2 – 4.1.12 of Tannehill et al.
Sections 2.5.1 and 2.5.2 of Durran.

By examining the leading order error in the modified equation, we can find the basic nature of the error. To estimate the error quantitatively, we use either analytical (as part of the stability analysis) or numerical method. We will first look at the former.

With the stability analysis, we were already examining the amplitude of waves in the numerical solution. For a linear advection equation, we want the amplification factor to be 1, so that the wave does not grow or decay in time. The von Neumann stability analysis actually also provides the information about propagation (phase) speed of the waves. Any difference between the numerical phase speed and true phase speed is the phase error.

Going back to the figure we showed at the beginning of this section (section 3), we can see that the first-order FTUS scheme has strong amplitude error but little phase error, while the 2nd-order CTCS scheme has large phase error but small amplitude error. The 4th-order CTCS scheme has a smaller phase error than its 2nd-order counterpart.

Amplitude Error

Recall that in the Neumann stability analysis, the frequency ω can be complex, and if it is, there will be either decay or growth in amplitude – which is entirely computational for a pure advection problem. This is so because

if ω is real,

$$|l| = |\exp(-iw\Delta t)| = |\cos(w\Delta t) + i\sin(w\Delta t)| = 1$$ → no change in amplitude.

if ω is complex, i.e., $\omega = \omega_R + i\,\omega_I$,

$$|l| = |\exp(-iw\Delta t)| = |\exp(-iw_R\,\Delta t)\exp(w_I\,\Delta t)| = |\exp(w_I\,\Delta t)| \neq 1$$ most of the time.

When $\omega_I > 0$, $|\lambda| > 1$, the wave grows and when $\omega_I < 0$, $|\lambda| < 1$ the wave decays (is damped). $|\lambda|$ is the amplitude change per time step and $|\lambda|^N$ is the total amplitude change after N steps.

If, e.g., $|\lambda| = 0.95$, then after 100 steps, the amplitude becomes 5.92×10^{-3}!

Remember for PDE $u_x + c\, u_x = 0$, the frequency ω is always real. Assuming wave solution $u = U \exp[\, i(kx-\omega t)\,]$, you can find $\omega = kc$, which is called the dispersion relation in wave dynamics. For the current problem the phase speed of waves is $\omega/k = c$ which is the same for all wave components. Therefore the analytic waves are non-dispersive.

Phase Error

For convinence, let's define

ω_a = frequency of the analytical solution (PDE)
ω_d = frequency of discrete solution (FDE)

then

$$l_a \equiv \exp(-iw_a \Delta t), \quad l_d \equiv \exp(-iw_d \Delta t).$$

Recall that if $z = x + i\, y$ ($i = \sqrt{-1}$), we can use the polar form and write

$$z = |\, z\,| \exp(\, i\, \theta\,) \quad \text{or} \quad z = |\, z\,| (\cos\theta + i \sin\theta\,)$$

where $|\, z\,| = (\, x^2 + y^2\,)^{1/2}$ is called the modulus of z.

Thus, $\lambda_a = |\, \lambda_a\,| \exp(\, i\, \theta_a\,) = \exp(\, i\, \theta_a\,)$ (because $|\, \lambda_a\,| = 1$).

We define θ_a = the phase change per time step of the analytic solution

$= -\, \omega_a \Delta t$ ~ frequency × time step size

For the finite difference solution, ω will, in general, be complex, i.e., ω has an imaginary part:

$$\omega_d = (\omega_d)_R + i\, (\omega_d)_I$$

$$\therefore \qquad \lambda_d = \exp[\, (\omega_d)_I \Delta t\,]\ \exp[\, -i\, (\omega_d)_R \Delta t\,] = |\, \lambda_d\,| \exp(\, i\, \theta_d\,) \rightarrow \qquad (13)$$

$\theta_d \equiv -\, (\omega_d)_R \Delta t$ is the phase change per time step associated with the F.D. scheme.
$|\lambda_d|$ is not necessarily 1 here.

From (13), we can see that $q_d = \operatorname{atan} \dfrac{\operatorname{Im}\{\, l_d\,\}}{\operatorname{Re}\{\, l_d\,\}}$. (14)

Taking the radio, $$\frac{q_d}{q_a} = \frac{-(w_d)_R \Delta t}{-w_a \Delta t} = \frac{kc_d}{kc_a} = \frac{c_d}{c_a}$$

for the same wave number, and the ratio tells us about the relative phase error.

If c_d / c_a < 1, the F.D. solution lags the analytic solution (moves slower)
If c_d / c_a > 1, the F.D. solution leads the analytic solution (moves faster)
If c_d / c_a = 1, the F.D. solution has no phase error

3.2.1. First-order upwind scheme

Let's now apply these notations of phase and amplitude error to the first-order upwind (donor-cell) scheme.

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + c\frac{u_i^n - u_{i-1}^n}{\Delta x} = 0 \tag{15}$$

Using Neumann method, assume that $u_i^n = U\, l_d^n e^{ikx_i}$, you can show for yourself that

$$\lambda_d = 1 - \mu + \mu \cos(k\Delta x) - i\, \mu \sin(k\Delta x) \tag{16}$$

where $\mu = c\Delta t/\Delta x$ is the Courant number.

$$|\, \lambda_d |^2 = 1 + 2\mu(\mu - 1)\, [1- \cos(k\Delta x)\,]$$

since $1-\cos(k\Delta x) \geq 0$,

$$|\, \lambda_d\, | \leq 1 \text{ when } \mu \leq 1.$$

Look at the $2\Delta x$ wave, $k\Delta x = 2\pi/(2\Delta x)\, \Delta x = \pi$

$$|\, \lambda_d |^2 = 1 + 2\mu(\mu - 1)\, [1+1] = 1 + 4\mu(\mu - 1). \tag{17}$$

When $\mu = 1$, $|\, \lambda_d\, | = 1$, there is no amplitude error.

When $\mu = 0.5$, $|\, \lambda_d\, | = 0$, the 2 Δx wave is completed damped in one time step!!

For a 4Δx wave and when $\mu = 0.5$, $|\, \lambda_d\, | = 0.5$ → 4Δx waves are damped by half in one time step!

Therefore, the upwind advection scheme is strongly damping. It should not be used except for some special reason.

The following figure shows the amplification modulus for the upwind scheme plotted for different values of μ (υ in the figure), the Courant number.

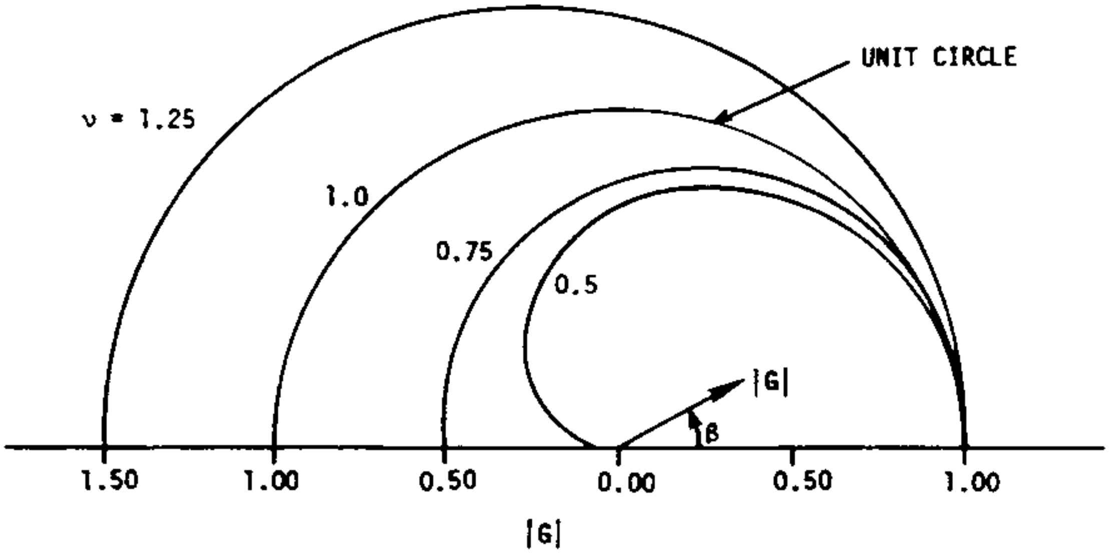


Figure 4.2 Amplification factor modulus for upstream differencing scheme.

β in the figure is our $k\Delta x$, and $\pi\Delta x/L \le k\Delta x \le \pi$. The lower and upper limits of $k\Delta x$ correspond to 2L and $2\Delta x$ waves, respectively. L is the length of the computational domain.

This is so because the shortest wave supported is $2\Delta x$ in wavelength, →

$$k\Delta x = 2\pi \;/\; (2\Delta x)\; \Delta x = \pi$$

The longest wave supported by a domain of length L is 2L in wavelength →

$$k\Delta x = 2\pi/(2L)\; \Delta x = \pi\Delta x/L$$

$k\Delta x \rightarrow 0$ when $L \rightarrow \infty$.

For example, for a $4\Delta x$ wave, $k\Delta x = \pi/2$

From the figure, we see that:

When $\mu = 1$, the amplification factor is 1, there is <u>no amplitude error</u> for all values of $k\Delta x$ (β), i.e., for all waves.

When μ > 1, the amplification factor is > 1 for all kΔx except for wave number zero. The amplification factor is the largest for the shortest wave (kΔx = π), implying that the 2Δx wave will grow the fastest when μ >1, in another word, the 2Δx wave is most unstable.

This is why we see grid-scale noises when the solution blows up!

When μ < 1, all waves are stable but are significantly damped. Again, the amplitude error is larger for shorter waves (larger kΔx). For Courant number of 0.75, the amplitude of the 2Δx wave is reduced by half after one single time step. The error is even bigger when μ = 0.5.

From the above, we can see that the numerical solution is poorest for the shortest waves, and as the wavelength increases, the solution becomes increasingly accurate. This is so because longer waves are sampled by a large number of grid points and are, therefore, well resolved.

2Δx wave is special in that it is often the most unstable when stability criterion is violated, and when the solution is stable it tends to be most inaccurate.

For generally cases, it is impossible to ensure μ =1 everywhere unless the advection speed is constant. Therefore, strong damping is inevitable with the upwind scheme. You will see severely smoothed solution when using this scheme.

The damping behavior of the upwind scheme can also be understood from the modified equation (13) discussed earlier. The leading error term is of the form $K\frac{\partial^2 u}{\partial x^2}$ which represents diffusion.

Now, let's examine the dispersion (phase) error of the upwind scheme.

According to earlier definitions and (16)

$$\theta_a = -\ \omega_a\ \Delta t\ \ = -\ c\ k\Delta t = -\ \mu\ k\Delta x$$

$$q_d = \text{atan}\frac{\text{Im}\{l_d\}}{\text{Re}\{l_d\}} = \text{atan}\frac{-m\sin(k\Delta x)}{1-m+m\cos(k\Delta x)}$$

From the above, the ratio of the numerical to analytic phase speed, $\frac{q_d}{q_a}$, can be calculated.

In the following figure this ratio is plotted as a function of β (kΔx) for μ (υ in the figure) = 0.25, 0.5 and 0.75.

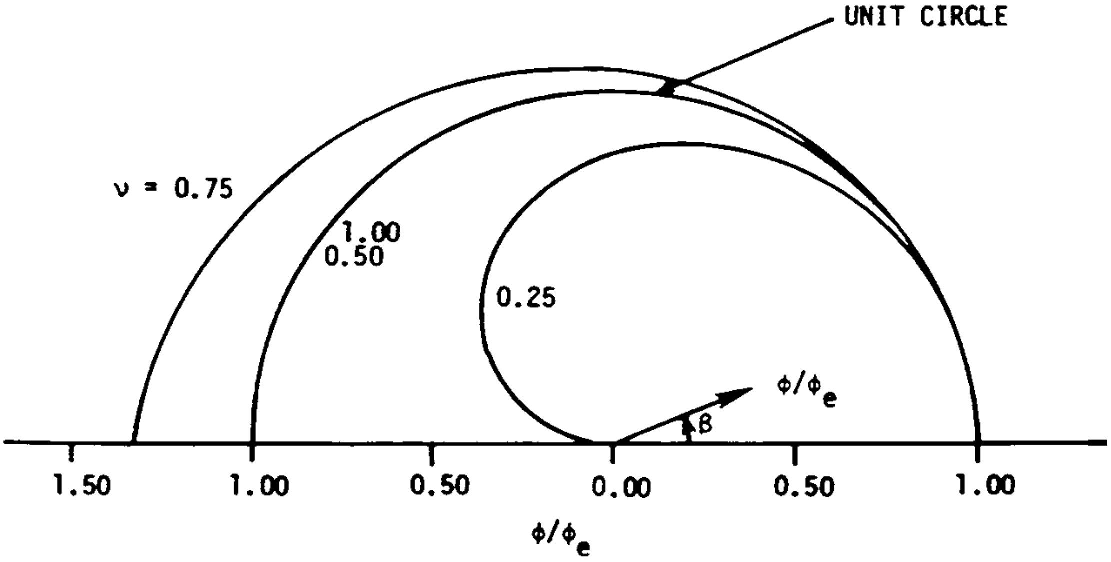


Figure 4.3 Relative phase error of upstream differencing scheme.

We can see that there is no phase error (corresponding to the unit circle) when μ =0.5.

All waves are slowed down when μ < 0.5. All waves are accelerated when 0.5 < μ < 1.0.

Again the phase error is larger for short waves (larger β, i.e., κΔx). The error is greatest for the 2Δx wave.

For a fixed μ, θ_d → 0 when kΔx → π, i.e., 2Δx does not move at all!

Because the F.D. phase speed is dependent on wavenumber k, the numerical solution is dispersive, whereas the analytical solution is not.

From the above discussions, we see that when using the upwind scheme the waves that move too slow are also strongly damped.

The upwind advection scheme is actually a monotonic scheme – it does not generate new extrema (minimum or maximum) that are not already in the field. For a positive field such as density, it will not generate negative values. We will discuss more about the monotonicity of numerical schemes later.

Note that for practical problems, c can change sign in a computational domain. In that case, which point to use in the spatial difference depends on the local sign of c:

$$\frac{u_i^{n+1}-u_i^n}{\Delta t}+c\frac{u_i^n-u_{i-1}^n}{\Delta x}=0 \qquad c>0$$
$$\frac{u_i^{n+1}-u_i^n}{\Delta t}+c\frac{u_{i+1}^n-u_i^n}{\Delta x}=0 \qquad c<0 \qquad (18)$$

Using the following definitions:

$c^+ = (c + |c|)/2, \; c^- = (c - |c|)/2,$

the upwind scheme in (19) can be written into a single expression

$$u_i^{n+1}=u_i^n+\frac{\Delta t}{\Delta x}\left[c^+(u_i^n-u_{i-1}^n)+c^-(u_{i+1}^n-u_i^n)\right] \qquad (19)$$

Substituting c^+ and c^- into (18) yields

$$u_i^{n+1}=u_i^n+\Delta t\frac{c(u_{i+1}^n-u_{i-1}^n)}{2\Delta x}+\frac{\Delta t\Delta x\,|\,c\,|}{2}\frac{(u_{i+1}^n-2u_i^n-u_{i-1}^n)}{\Delta x^2}. \qquad (20)$$

One can see that the 2nd term on RHS is the advection term in centered difference form and the 3rd term has a form of diffusion. If one uses forward-in-time centered-in-space scheme to discretize equation (5), one will get a FDE like (2) except for the 3rd term on RHS. The scheme is known as the Euler explicit scheme, and the stability analysis tells us that it is absolutely unstable. So it should never be used. Apparently, the 'diffusion term' included in the upwind scheme stabilizes the upwind scheme – it is achieved by damping the otherwise growing short waves.

The included 'diffusion term' also introduces excessively damping to the short waves, as seen earlier. One possible remedy is to attempt to remove this excessive diffusion through a corrective step and several corrective steps. This is exactly what is done in the Smolarkiewicz (1983, 1984) scheme, which is rather popular in the field of meteorology.

Because Smolarkiewicz scheme is based on the upwind scheme, it maintains the positive definiteness of the advected fields therefore is a good choice for advecting mass and water variables.

References:

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.

3.2.2. Leapfrog scheme for advection

In this section, we examine a perhaps most commonly used scheme in atmospheric models – the leapfrog centered advection scheme.

Here leapfrog refers to finite difference in time – the frog leaps over time level n from n-1 to n+1 – it is a name for the second-order centered difference in time.

Leapfrog scheme is usually used together with centered difference in space – and the latter can be of 2nd or higher order.

The leapfrog scheme gives us second order accuracy in time.

The PDE is

$$\frac{u_i^{n+1} - u_i^{n-1}}{2\Delta t} + c\frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x} = 0 \qquad (21)$$

$$t = O(\Delta x^2, \Delta t^2)$$

and (show it yourself)

$$l_{\pm} = -im\sin(k\Delta x) \pm [1 - m^2\sin^2(k\Delta x)]^{1/2} \qquad (22)$$

If $1 - m^2\sin^2(k\Delta x) \ge 0$, then $|\, l_{\pm} | \equiv 1$ and there is no amplitude error for all waves. This is the most attractive property of the leapfrog scheme.

In (22), we see that there are two roots for λ - one of them is acutally non-physical and is known as the computational mode.

Which one is computational and how does it behave?

Let's look at the positive root λ_+ first:

$$\lambda_+ = |\lambda_+| \exp(-i\,\beta_+)$$

where $\beta_+ = -\theta_d$ (θ_d is the phase change in one time step for the discretized scheme, as defined ealier).

If $\mu \le 1$, $|\lambda_+| = 1$.

$$l_+ = \cos(b_+) - i\sin(b_+) = -ia + [1 - a^2]^{1/2}$$

where a = $m\sin(k\Delta x)$, therefore

$$b_+ = \sin^{-1}[m\sin(k\Delta x)].$$

Now consider the <u>negative root</u> λ_- :

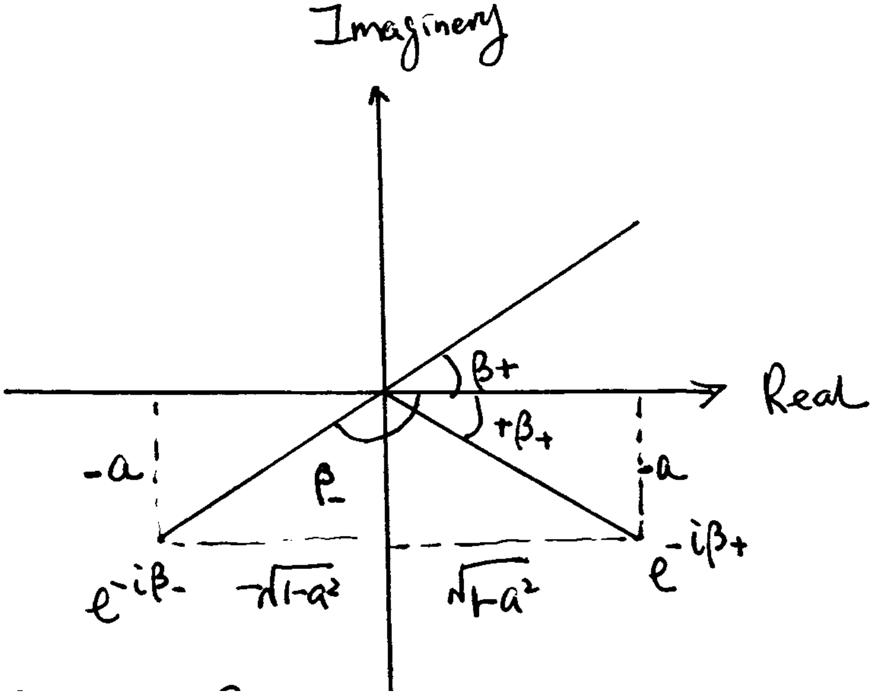
$$\lambda_- = |\lambda_-| \exp(-i\beta_-)$$

If $\mu \le 1$, $|\lambda_-| = 1$.

$$l_\_ = \cos(b_\_) - i\sin(b_\_) = -ia - [1 - a^2]^{1/2}$$

with the aid of the following schematics, we can see that

$$-b_\_ = p + b_+ .$$



We see that the phase of the negative root is the same as that of the positive root shifted by π then multiplied by –1.

What does all this mean then?

For a single wave k, we can write the solution as a linear combination of these two modes (since both modes are present):

$$
\begin{aligned}
u_i^n &= (A\,l_+^n + B\,l_-^n)e^{ikx_i} \\
&= [Ae^{-ib_+n} + Be^{i(p+b_+)n}]e^{ikx_i} \\
&= [Ae^{-ib_+n} + B(-1)^n e^{ib_+n}]e^{ikx_i}
\end{aligned}
\tag{23}
$$

where A and B are the amplitude of these two modes at time 0.

Which root corresponds to the computational mode then? The negative one, the one that give rises to the second term in (23), because of the following observations:

(1) The computational mode changes sign every time step. The period of oscillation is 2Δt.

(2) It has a phase opposite to the physical mode, therefore it propagates in the opposite direction from the physical mode.

(3) Because of the 2Δt period, the computational mode can be damped effectively using a time filter, which will be discussed in next section.

(4) The presence of the computational mode is due to the use of three time levels, which requires two initial conditions instead of one – the first and second time step integrations start from time level –1 and 0 respectively, which are two different initial conditions.

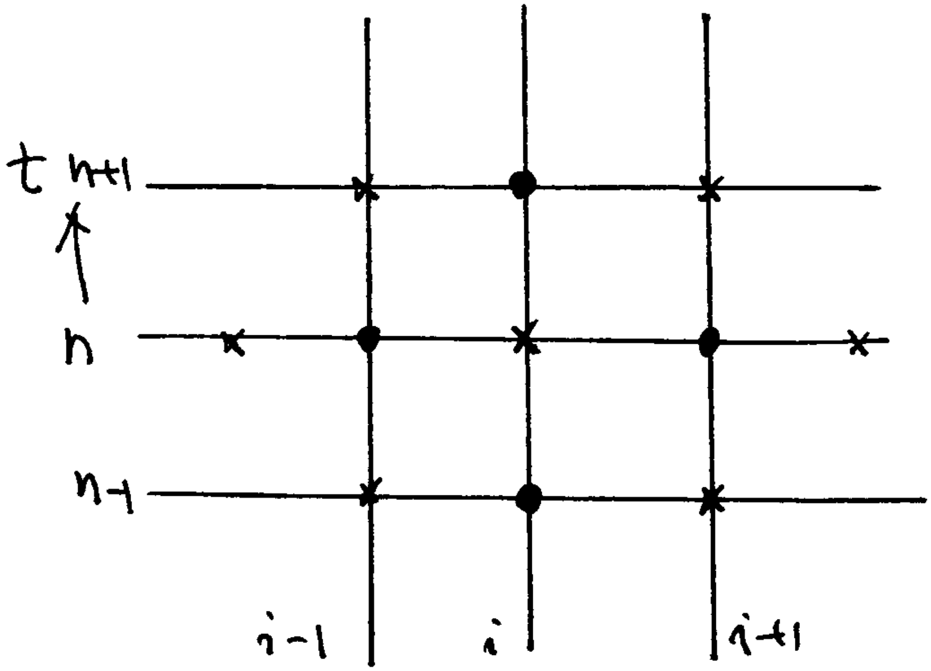
In practice, we usually have only one initial condition – we often start the time integration by using forward-in-time scheme for the first step, i.e., for the first step, we do

$$\frac{u_i^1 - u_i^0}{\Delta t} = -c\frac{u_{i+1}^0 - u_{i-1}^0}{2\Delta x}$$

and for the second, we do

$$\frac{u_i^2 - u_i^0}{2\Delta t} + c\frac{u_{i+1}^1 - u_{i-1}^1}{2\Delta x} = 0\,.$$

An additional note: when $u_i^{n+1} = u_i^{n-1} - \frac{c\Delta t}{\Delta x}(u_{i+1}^n - u_{i-1}^n)$ is used to integrate the advection equation, we can experience the grid separation problem, as show schematically below:



Due to the layout of the computational stencil, the solution at cross points never know what's going on at the dot points. As the solution march forward in time, the solutions at neighboring points can split away from each other. This problem is also related to the use of three time levels, and can be alleviated by the use of Asselin time filter. An artificial spatial smoothing term of the form of $K\partial^2u/\partial x^2$ will also help. In practice, other forcing terms in the equation can also couple the solutions together.

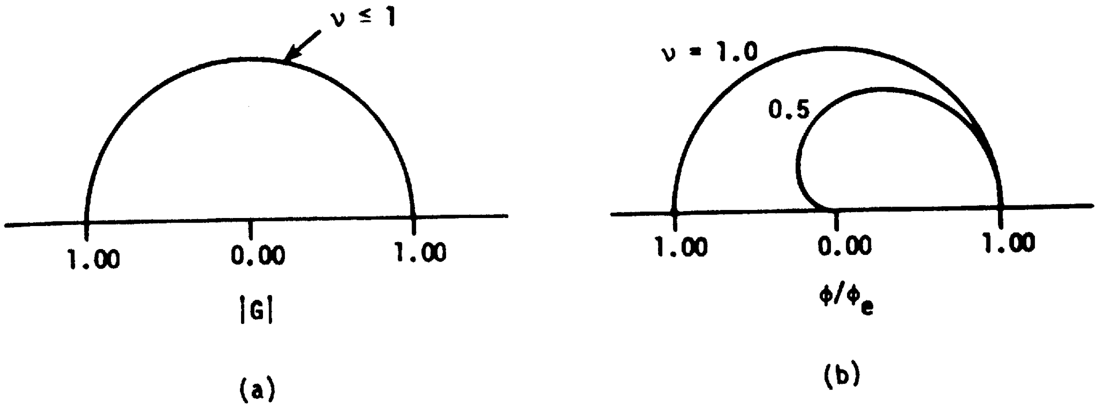


Figure 4.7 Leap frog method. (a) Amplification factor modulus. (b) Relative phase error.

3.3.3. Asselin Time Filter

The Asselin (also called Robert-Asselin) time filter (Robert 1966; Asselin 1972) is designed to re-couple of the splitting solutions in time and damp the computational mode found with the leapfrog scheme and others.

It is a two-step process:

(1) u is integrated to time level n+1 using the regular leapfrog scheme,

$$u_i^{*n+1} = u_i^{n-1} - m\,(u_{i+1}^{*n} - u_{i-1}^{*n}) \qquad (24)$$

where * indicates values that have not been 'smoothed'.

(2) a filter is then applied to three time levels of data

$$u_i^n = u_i^{*n} + e\,(u_i^{*n+1} - 2u_i^{*n} + u_i^{n-1})\,. \qquad (25)$$

Note that the term in second term in (15) is a finite difference version of $\partial^2 u/\partial t^2$ - the diffusion in time which tends to damp high-frequency oscillations.

If we use (25) in (24), we can do a stability analysis and examine the impact of the time filter on solution accuracy:

$$l_\pm = -ia + e \ \pm [b - a^2]^{1/2} \qquad (26)$$

where a= $m\sin(k\Delta x)$ and b = $(1-\varepsilon)^2$ [compare (22) with (16)].

If b – $a^2 \geq 0$ (note that this stability condition has also changed), we have

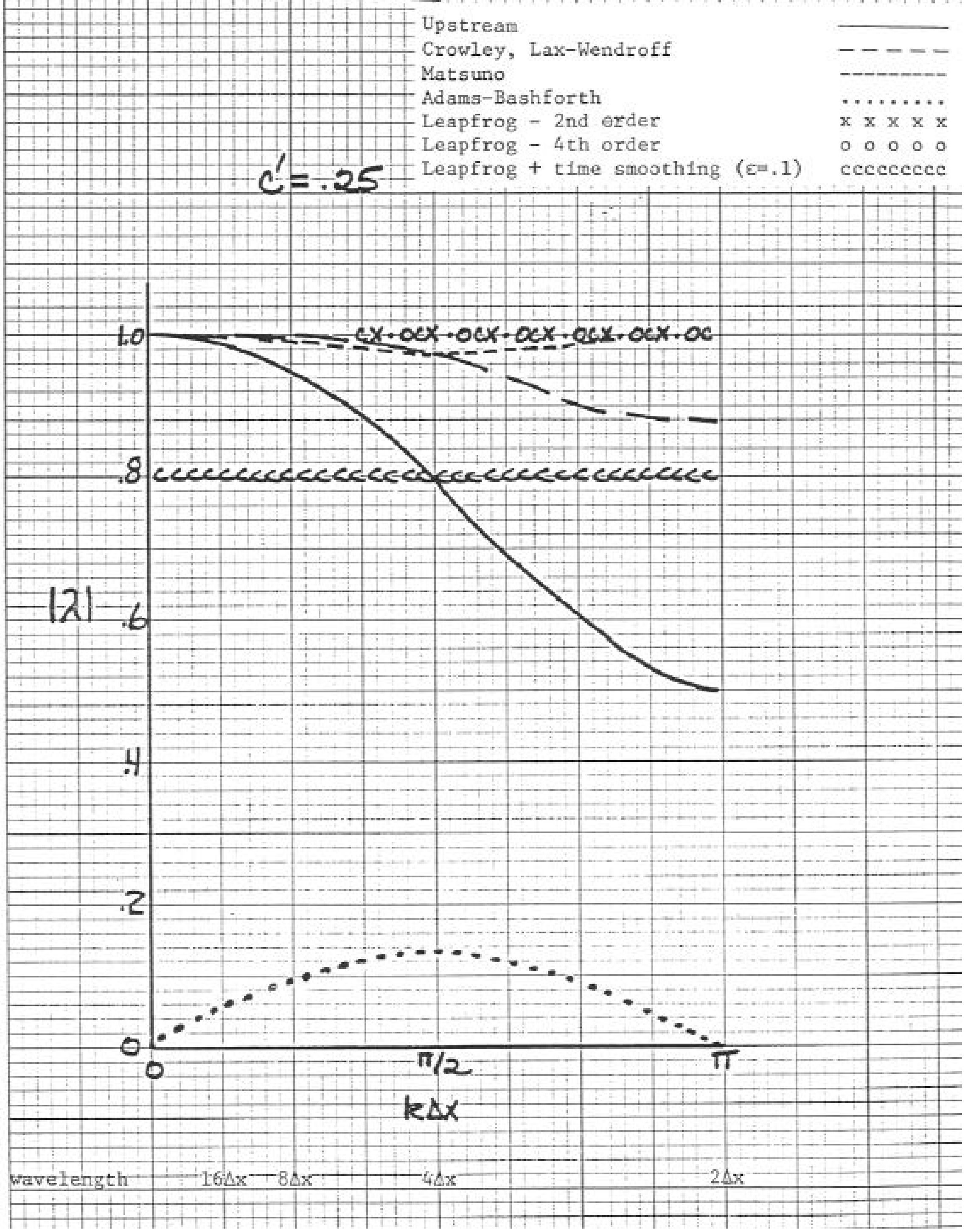
$$|\lambda|^2 = (\varepsilon^2 + b) \pm 2\varepsilon\,(b - a^2)^{1/2}.$$

We can plot this to determine its effect on the solution.

We will find that:

(1) amplitude error is introduced by the time filter;
(2) the time filter reduces the time integration scheme from second-order accurate to first-order accuracy only
(3) the filter makes the stability condition more restrictive (can use smaller Δt now).

FIG 1


Upstream ————
Crowley, Lax-Wendroff — — — — —
Matsuno ---------
Adams-Bashforth ....
Leapfrog - 2nd order x x x x x
Leapfrog - 4th order o o o o o
Leapfrog + time smoothing (ε=.1) ccccccccc
c' = .25
1.0
.8
.6
.4
.2
0
|λ|
cx·ocx·ocx·ocx·ocx·ocx·oc
0
π/2
π
kΔx
wavelength
16Δx
8Δx
4Δx
2Δx

We want to use as small a ε as possible. Typically $\varepsilon = 0.05$ to 0.1.

The leapfrog (2nd or 4th-order) centered difference scheme combined with the Asselin filter is used in the ARPS for the advective process (more complex monotonic advection schemes are also available for scalar advection).

Reference:

Robert, A. J., 1966: The integration of a low order spectral form of the primitive meteorological equations. *J. Meteor. Soc. Japan*, **44**, 237-245.

Asselin, R., 1972: Frequency filter for time integration. *Mon. Wea Rev.*, **100**, 487-490.

Reading: Durran Section 2.3.5.

3.2.4. Adam-Bashforth schemes

Second-order Adam-Bashforth Scheme

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = -c\left[\frac{3}{2}\frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x} - \frac{1}{2}\frac{u_{i+1}^{n-1} - u_{i-1}^{n-1}}{2\Delta x}\right]$$

- The RHS is a linear extrapolation of $\delta_{2x}u$ from n-1 and n to n+1/2, so that the scheme is "centered" in time at n+1/2.
- Second order in time and space
- Stability analysis shows that

$$l_\pm = \frac{1}{2}\left[1 - \frac{3}{2}is \pm \left(1 - \frac{9}{4}s^2 - i\,s\right)^{1/2}\right]$$

where $s = m\sin(k\Delta x)$.

Note that this scheme is also a 3 time level scheme and 3 time level schemes always have two modes – one physical and one computational. We can see there for the physical mode, λ –> 1 as s –> 0, and for the computational mode, λ –> 0 as s –> 0.

If s << 1, we can show that

$$|\lambda_+| \approx (1 + s^4/4)^{1/2}$$

$$|\lambda_-| \approx 0.5\, s(1 + s^2)^{1/2}$$

(You can show it by performing binomial expansion).

Clearly $|\lambda_+| > 1$ for $s \neq 0$ therefore the scheme is absolutely unstable.

However, for small enough values of s (i.e., Courant number), because s is raised to the 4th power, $|\lambda_+|$ can be close enough to 1 so that the growth rate is small enough for the scheme to be still usable.

One can estimate the growth rate in terms of e-folding time – i.e., the time taken for a wave to growth by a factor of e.

However, it is higher-order Adam-Bashforth (AB) schemes that we are more interested in. The higher-order AB scheme can be obtained by extrapolating the right hand side of the equation (i.e., F in $u_t = F$) to time level n+1/2, as we do for the 2nd-order AB scheme, but using high-order (e.g., 2nd-order) polynomials, which will also involve more time levels.

Third-order Adam-Bashforth Scheme

The 3rd-order AB scheme thus obtained has the form of

$$\frac{u_i^{n+1}-u_i^n}{\Delta t}=-\frac{c}{12}[23d_{2x}u^n-16d_{2x}u^{n-1}+5d_{2x}u^{n-2}]$$

- It involves data at four time levels – require more storage space.
- And it has two computational modes and one physical mode.
- The computational modes are strongly damped, however, unlike the leapfrog scheme, so there is no need for time filtering.

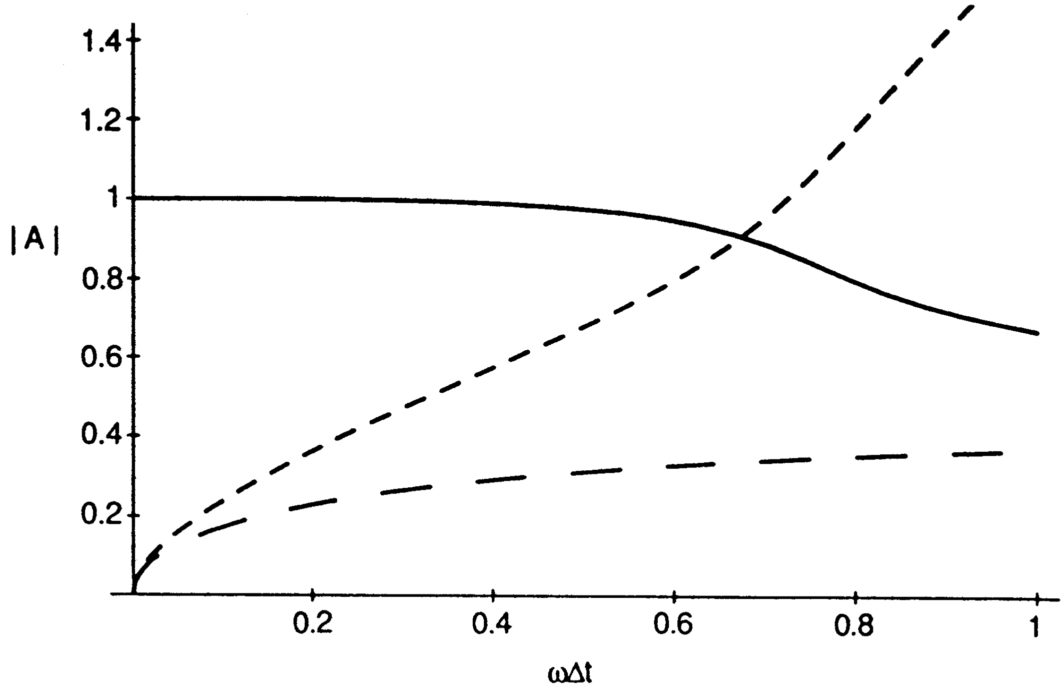


FIG. 1. Magnitude of the amplification factor for the third-order Adams–Bashforth scheme plotted as a function of $\omega\Delta t$. Solid line is the physical mode; dashed curves are the two computational modes.

- Most accurate results are obtained for μ near stability limit. This is not true for the leapfrog 4th-order centered in space scheme. That solution is more accurate for μ < 0.5 where certain cancellation between time and space truncation errors occur.

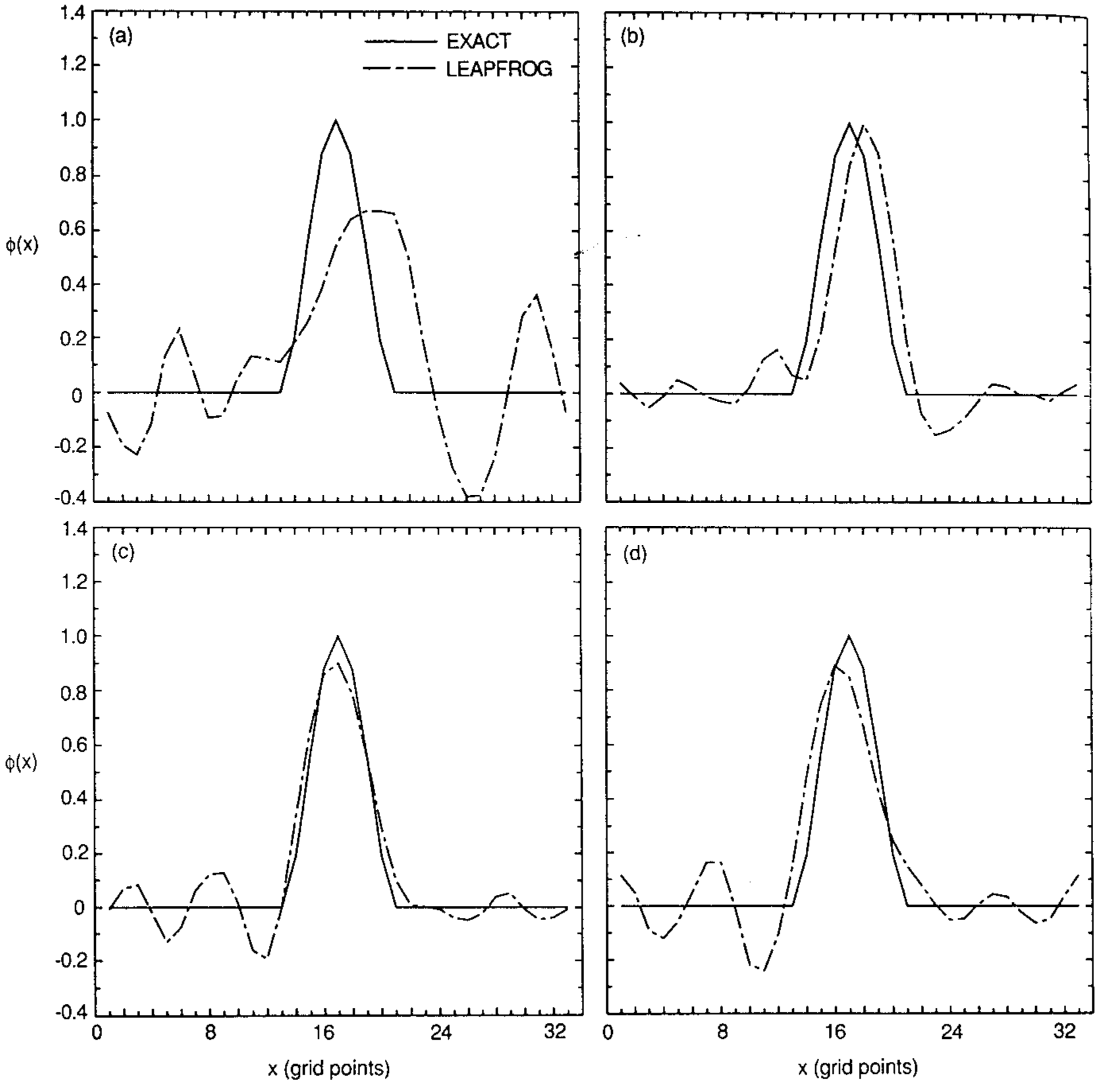


FIG. 3. Effect of leapfrog stepsize on the accuracy of fourth-order centered-difference solution to the advection equation. Shown are the exact and numerical solutions computed using Courant numbers of (a) 0.7272, (b) 0.5, (c) 0.3, and (d) 0.1. All results are for a nondimensional time of 3.

- Durran (1991 MWR) shows that 3rd-order AB time difference combined with 4th-order spatial difference is a good choice – it is in generally more accurate than the commonly used leapfrog 4th-order centered-in-space scheme.

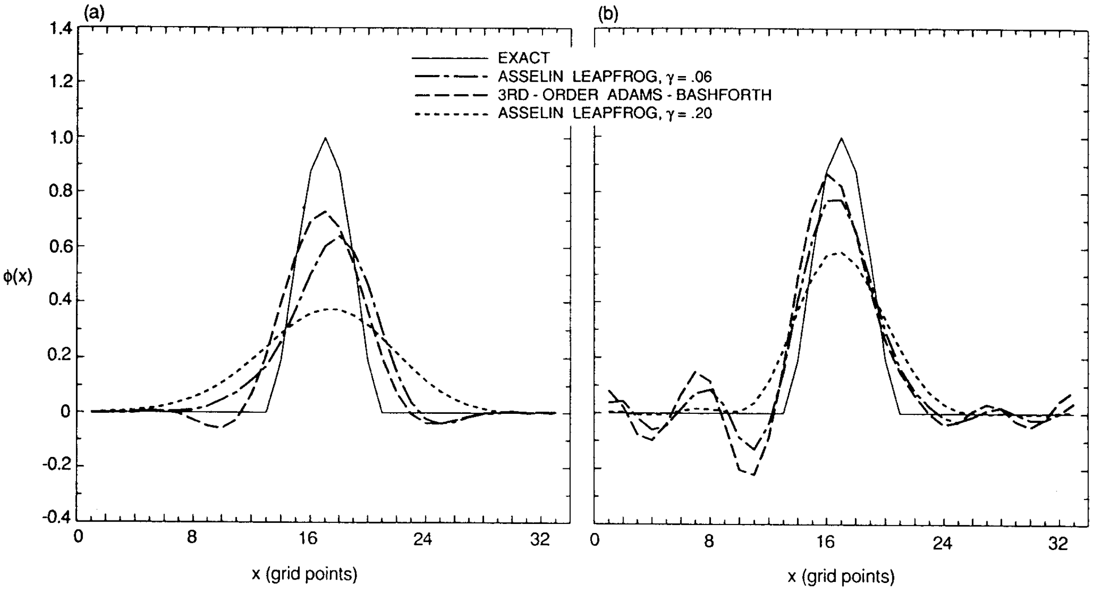


FIG. 5. Comparison of an exact solution to the advection equation with results obtained using Adams–Bashforth and Asselin-filtered leapfrog time differencing in a fourth-order finite-difference model at a nondimensional time of 3, for (a) μ = 0.5 and (b) μ = 0.2.

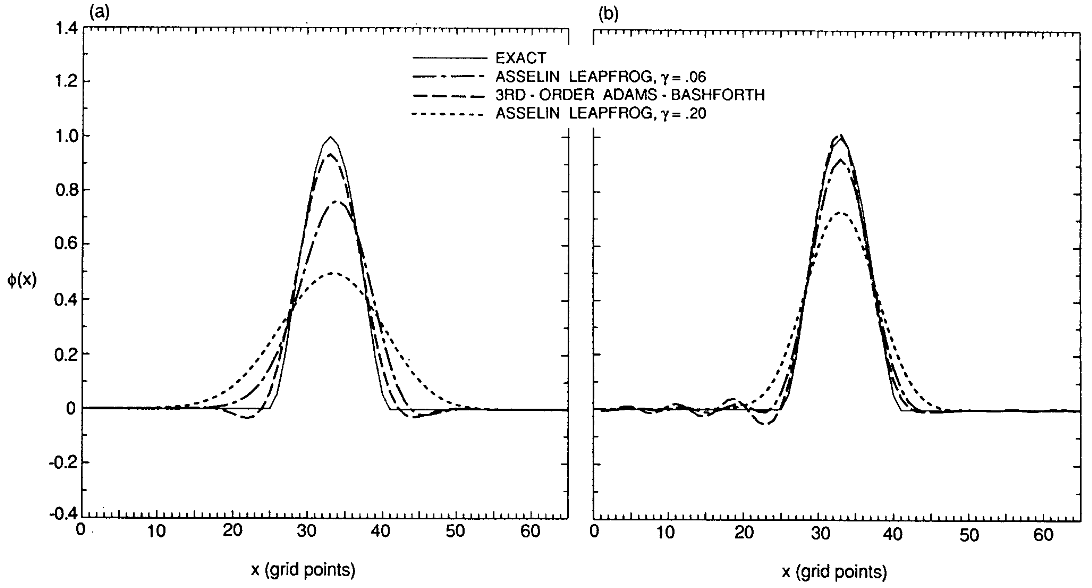


FIG. 6. As in Fig. 5, except that the spatial resolution has been doubled.

3.2.5. Other schemes

There are many other schemes for solving the advection equation. In the following are a some of them, given together with brief discussions on their important properties.

Euler explicit (Euler refers to forward in time)

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + c\frac{u_{i+1}^n - u_i^n}{\Delta x} = 0, \quad c > 0 \quad \text{- forward-in-time, downstream in space}$$

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + c\frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x} = 0 \quad \text{- forward-in-time, centered in space}$$

- Both schemes are absolutely unstable. You can show it for yourself.
- They are of no use.

Lax Method

$$\frac{u_i^{n+1} - (u_{i+1}^n + u_{i-1}^n)/2}{\Delta t} + c\frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x} = 0$$

- 1st-order in time, 2nd-order in space.
- Stable when $|\mu| \le 1$.
- Large dissipation error.
- Significant eading phase error - waves propagate faster.
  $2\Delta x$ waves twice as fast when μ=0.2.

Lax-Wendroff

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = -c\frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x} + \frac{c^2 \Delta t}{2}\frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{(\Delta x)^2}$$

- Effectively an Euler explicit (FTCS) scheme plus a diffusion term.

  Its derivation is interesting – it's based the Taylor series expansion in time first:

$$u_i^{n+1} = u_i^n + \Delta t u_t + \frac{1}{2}(\Delta t)^2 u_{tt} + O(\Delta t^3)$$

and use $u_t = -c\, u_x$ and $u_{tt} = c^2\, u_{xx}$ to rewrite it as

$$u_i^{n+1} = u_i^n - c\Delta t u_x + \frac{1}{2}c^2 (\Delta t)^2 u_{xx} + O(\Delta t^3)\,.$$

It is then discretized in space.

- Stable when $|\mu| \leq 1$
- Amplitude (dissipation) error for short waves
- Mostly lagging phase error, for short waves. Leading phase error for shortest waves when μ is near 0.75.

We have actually obtained this scheme before based on characteristics and second order interpolation. See Section 2.3.

MacCormack (an example of two-step predictor-corrector method)

Predictor: $$(u_i^{n+1})^* = u_i^n - c\Delta t \frac{u_{i+1}^n - u_i^n}{\Delta x}$$

Corrector: $$u_i^{n+1} = \frac{1}{2}\left[u_i^n + (u_i^{n+1})^* - c\Delta t \frac{(u_i^{n+1})^* - (u_{i-1}^{n+1})^*}{\Delta x} \right]$$

- Combination of upwind and downwind steps
- Intermediate prediction is used in the second corrector step
- In the corrector step, the time difference is 'backward in time'
- For linear advection equation, this scheme is <u>equivalent</u> to (you can show this by substituting the 1st eq. into the 2nd), therefore its properties are the same as, the Lax-Wendroff scheme.

Euler Implicit (Euler refers to forward in time)

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + c\frac{u_{i+1}^{n+1} - u_{i-1}^{n+1}}{2\Delta x} = 0$$

- 1st-order in time and 2nd-order in space.
- Unconditionally stable.
- Relatively small dissipation error, only for intermediate wave lengths.
  No dissipation error for longest and shortest waves.
- Significant lagging phase error for short waves.
- Need to solve a coupled system of equations.
  Tridiagonal in 1-D. Block trigiagonal in 2-D.

Time-centered Implicit (Trapezoidal)

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + \frac{c}{2}\left[\frac{u_{i+1}^{n+1} - u_{i-1}^{n+1}}{2\Delta x} + \frac{u_{i+1}^{n} - u_{i-1}^{n}}{2\Delta x} \right] = 0$$

- 2nd-order in both time and space.
- Absolutely stable.
- No dissipation error for all waves (similar to leapfrog scheme which is also 2nd-order accurate in time)
- Significant lagging phase error for short waves, similar to Euler implicit.

Matsuno (forward-backward two-step) Scheme

$$\frac{(u_i^{n+1})^* - u_i^n}{\Delta t} + c\frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x} = 0$$

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + c\frac{(u_{i+1}^{n+1})^* - (u_{i-1}^{n+1})^*}{2\Delta x} = 0$$

- 1st-order in time, second order in space
- Stable when $\mu \le 1$.
- Relatively large dissipation and phase error

Leapfrog Fourth-order Centered-in-Space Scheme

$$\frac{u_i^{n+1} - u_i^{n-1}}{2\Delta t} + c\left[\frac{4}{3}\frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x} - \frac{1}{3}\frac{u_{i+2}^n - u_{i-2}^n}{4\Delta x}\right] = 0$$

- 2nd-order in time and 4th-order in space
- Stable for $\mu \le 0.728$ (more restrictive than 2nd-order)
- No dissipation error without time filter
- Also contains computational mode, as all three-time level schemes do
- Smaller phase error than 2nd-order centered-in-space counterpart
- Leapfrog scheme can be combined with centered spatial difference schemes of even higher order

FIG 1

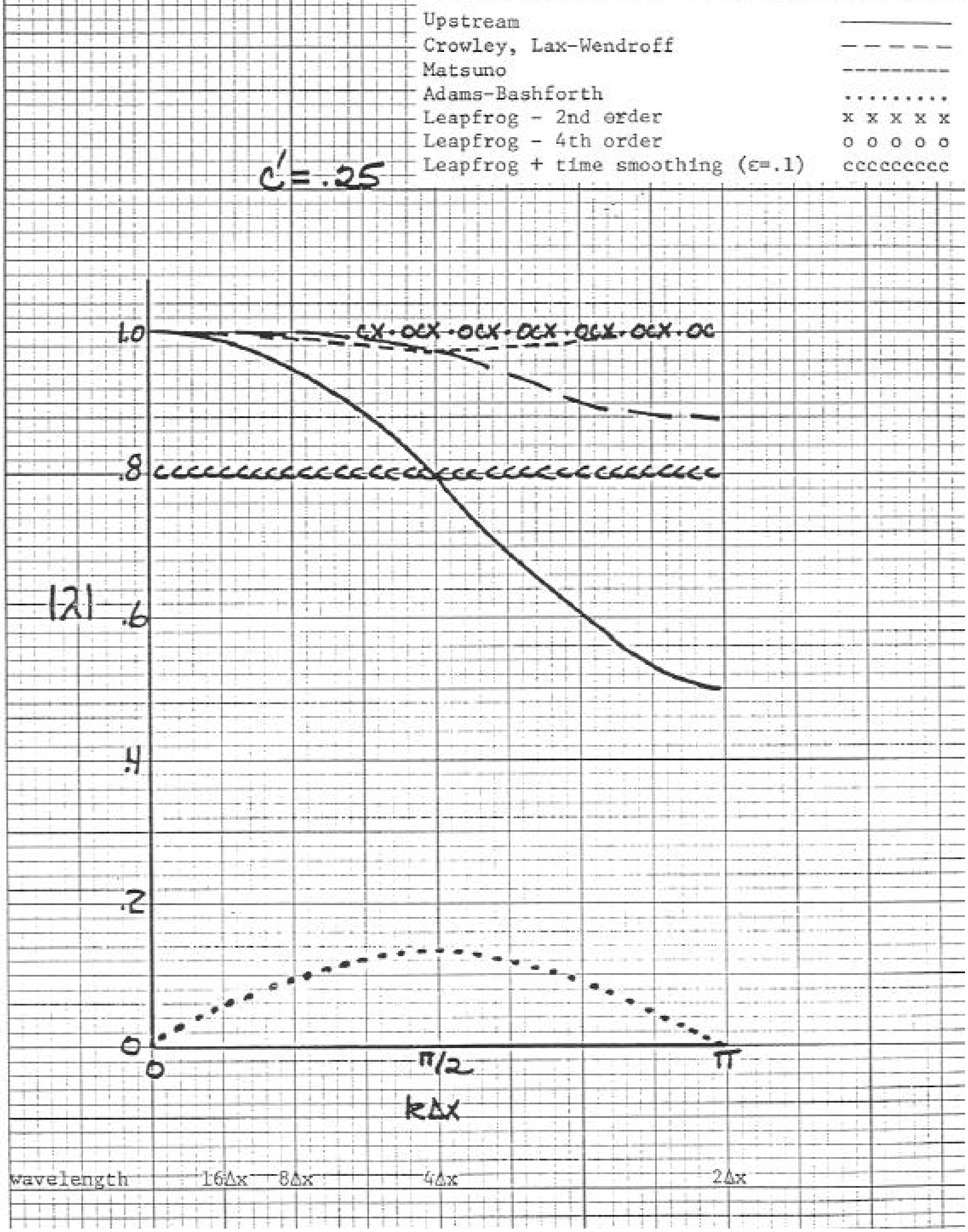

Upstream ———————
Crowley, Lax-Wendroff — — — — —
Matsuno ---------
Adams-Bashforth ....
Leapfrog - 2nd order x x x x x
Leapfrog - 4th order o o o o o
Leapfrog + time smoothing (ε=.1) ccccccccc
c' = .25
cx·ocx·ocx·ocx·ocx·ocx·oc
1.0
.8
.6
.4
.2
0
|λ|
0
π/2
π
kΔx
wavelength
16Δx
8Δx
4Δx
2Δx

FIG 2

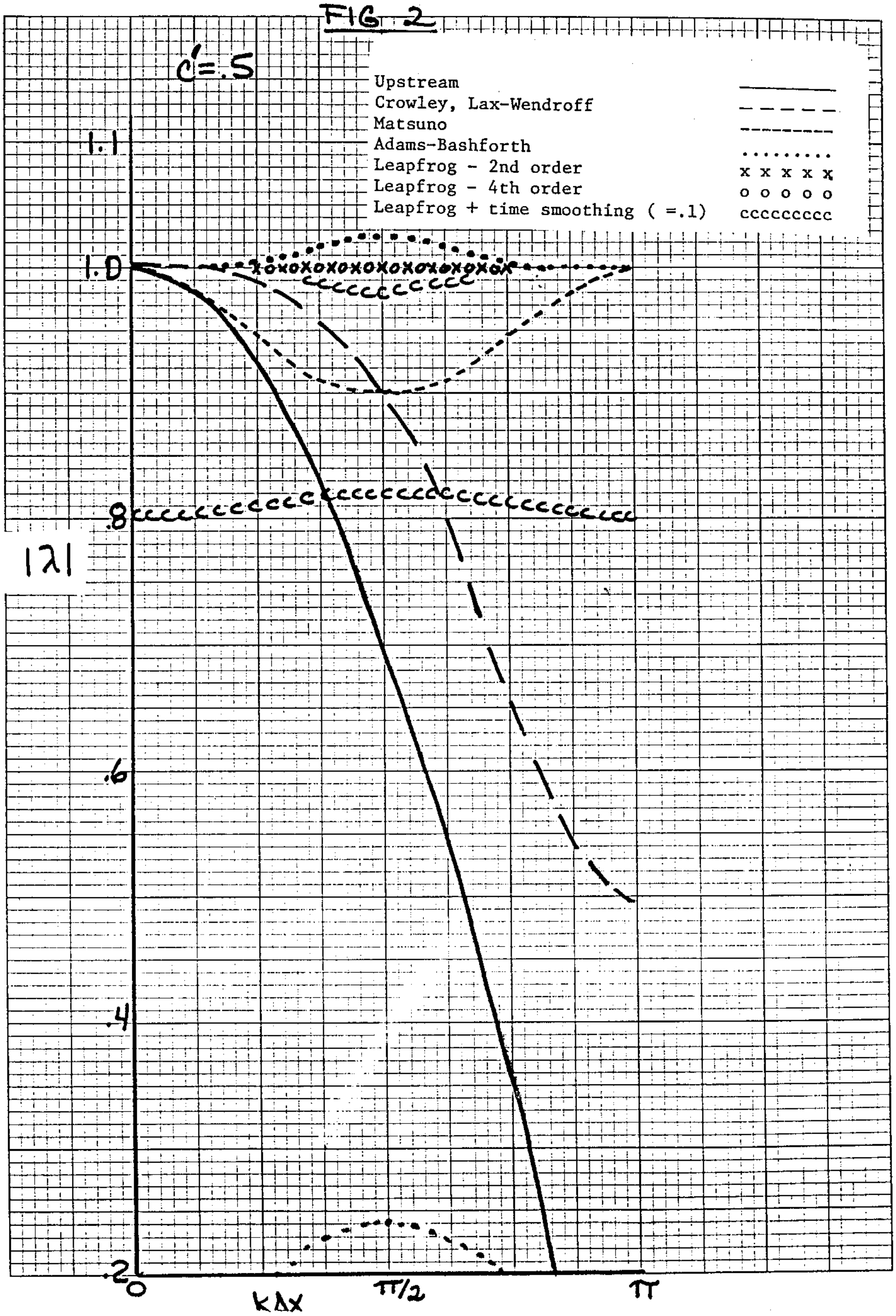

c' = .5
Upstream
Crowley, Lax-Wendroff
Matsuno
Adams-Bashforth
Leapfrog - 2nd order
Leapfrog - 4th order
Leapfrog + time smoothing (=.1)
x x x x x
o o o o o
ccccccccc
1.1
1.0
.8
|λ|
.6
.4
.2
0
kΔx
π/2
π

FIG 3

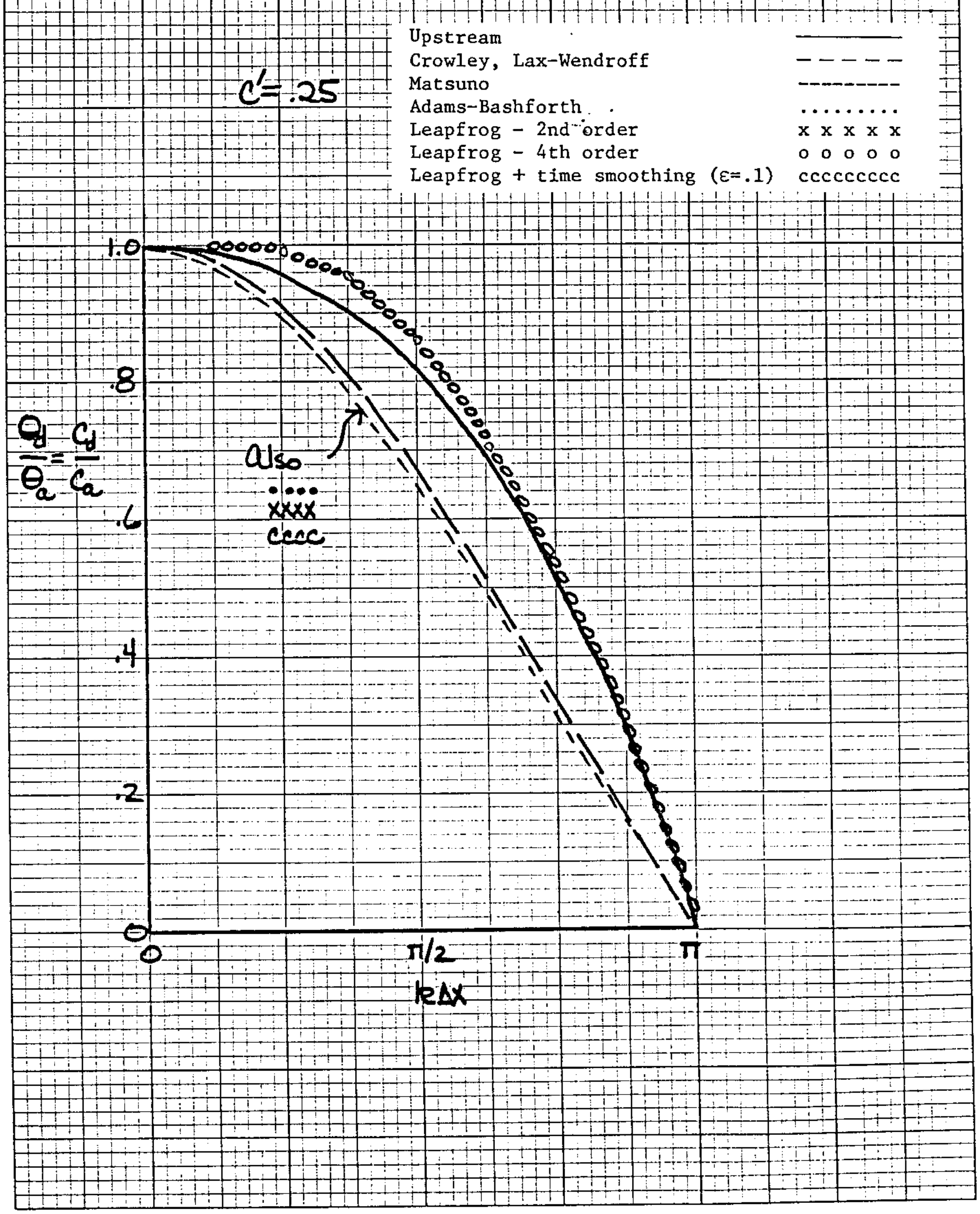
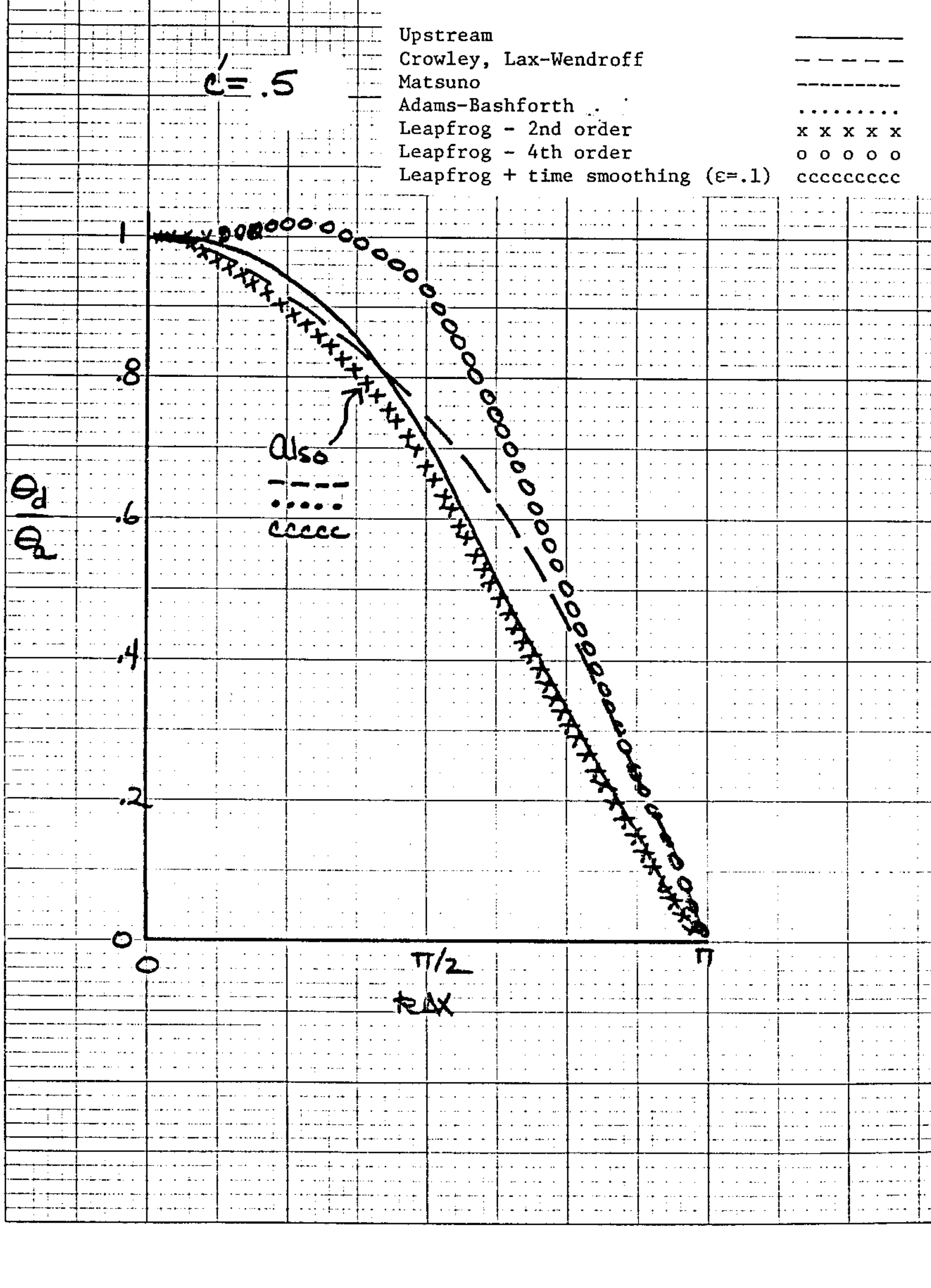
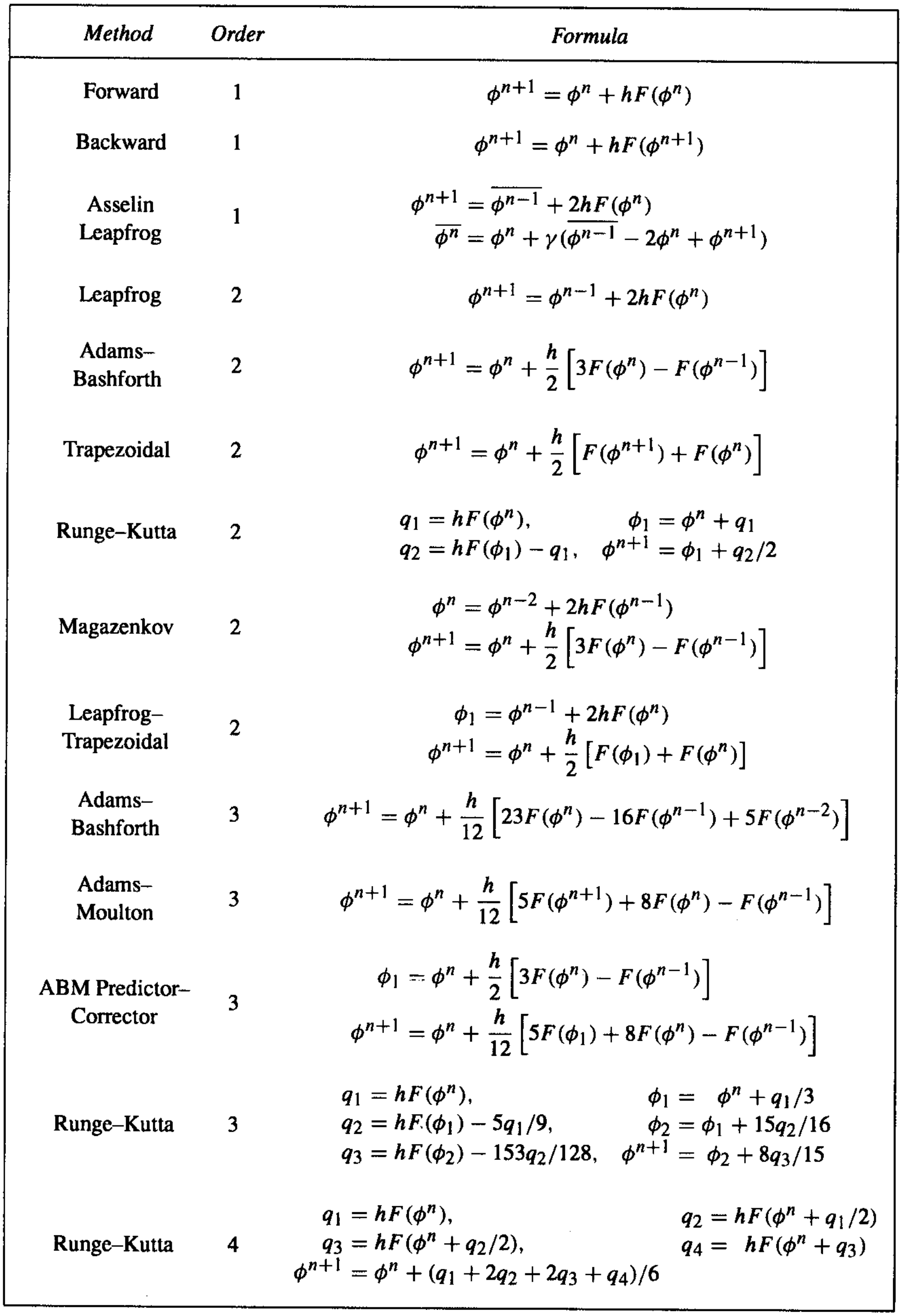

Upstream ————
Crowley, Lax-Wendroff — — — — —
Matsuno ---------
Adams-Bashforth ....
Leapfrog - 2nd order x x x x x
Leapfrog - 4th order o o o o o
Leapfrog + time smoothing (ε=.1) ccccccccc
C′=.25
1.0
.8
.6
.4
.2
0
$\frac{\Theta_d}{\Theta_a}=\frac{C_d}{C_a}$
also
....
xxxx
cccc
0
π/2
π
kΔx

FIG 4


Upstream
Crowley, Lax-Wendroff
Matsuno
Adams-Bashforth
Leapfrog - 2nd order x x x x x
Leapfrog - 4th order o o o o o
Leapfrog + time smoothing (ε=.1) ccccccccc
c = .5
also
ccccc
$\frac{\Theta_d}{\Theta_a}$
1
.8
.6
.4
.2
0
0
π/2
π
kΔx

List of commonly used time difference schemes and their basic properties (from Durran):



Method	*Order*	*Formula*
Forward	1	$\phi^{n+1} = \phi^n + hF(\phi^n)$
Backward	1	$\phi^{n+1} = \phi^n + hF(\phi^{n+1})$
Asselin Leapfrog	1	$\phi^{n+1} = \overline{\phi^{n-1}} + 2hF(\phi^n)$ $\overline{\phi^n} = \phi^n + \gamma(\overline{\phi^{n-1}} - 2\phi^n + \phi^{n+1})$
Leapfrog	2	$\phi^{n+1} = \phi^{n-1} + 2hF(\phi^n)$
Adams–Bashforth	2	$\phi^{n+1} = \phi^n + \frac{h}{2}\left[3F(\phi^n) - F(\phi^{n-1})\right]$
Trapezoidal	2	$\phi^{n+1} = \phi^n + \frac{h}{2}\left[F(\phi^{n+1}) + F(\phi^n)\right]$
Runge–Kutta	2	$q_1 = hF(\phi^n), \quad \phi_1 = \phi^n + q_1$ $q_2 = hF(\phi_1) - q_1, \quad \phi^{n+1} = \phi_1 + q_2/2$
Magazenkov	2	$\phi^n = \phi^{n-2} + 2hF(\phi^{n-1})$ $\phi^{n+1} = \phi^n + \frac{h}{2}\left[3F(\phi^n) - F(\phi^{n-1})\right]$
Leapfrog–Trapezoidal	2	$\phi_1 = \phi^{n-1} + 2hF(\phi^n)$ $\phi^{n+1} = \phi^n + \frac{h}{2}\left[F(\phi_1) + F(\phi^n)\right]$
Adams–Bashforth	3	$\phi^{n+1} = \phi^n + \frac{h}{12}\left[23F(\phi^n) - 16F(\phi^{n-1}) + 5F(\phi^{n-2})\right]$
Adams–Moulton	3	$\phi^{n+1} = \phi^n + \frac{h}{12}\left[5F(\phi^{n+1}) + 8F(\phi^n) - F(\phi^{n-1})\right]$
ABM Predictor–Corrector	3	$\phi_1 = \phi^n + \frac{h}{2}\left[3F(\phi^n) - F(\phi^{n-1})\right]$ $\phi^{n+1} = \phi^n + \frac{h}{12}\left[5F(\phi_1) + 8F(\phi^n) - F(\phi^{n-1})\right]$
Runge–Kutta	3	$q_1 = hF(\phi^n), \quad \phi_1 = \phi^n + q_1/3$ $q_2 = hF(\phi_1) - 5q_1/9, \quad \phi_2 = \phi_1 + 15q_2/16$ $q_3 = hF(\phi_2) - 153q_2/128, \quad \phi^{n+1} = \phi_2 + 8q_3/15$
Runge–Kutta	4	$q_1 = hF(\phi^n), \quad q_2 = hF(\phi^n + q_1/2)$ $q_3 = hF(\phi^n + q_2/2), \quad q_4 = hF(\phi^n + q_3)$ $\phi^{n+1} = \phi^n + (q_1 + 2q_2 + 2q_3 + q_4)/6$

TABLE 2.1. Summary of methods for the solution of ordinary differential equations. The second- and third-order Runge–Kutta methods are low storage variants; $h = \Delta t$

Method	*Storage Factor*	*Efficiency Factor*	*Amplification Factor*	*Phase Error*	*Max s*
Forward	2	0	$1 + \frac{s^2}{2}$	$1 - \frac{s^2}{3}$	0
Backward	*	∞	$1 - \frac{s^2}{2}$	$1 - \frac{s^2}{3}$	∞
Asselin Leapfrog	3	< 1	$1 - \frac{\gamma s^2}{2(1-\gamma)}$	$1 + \frac{(1+2\gamma)s^2}{6(1-\gamma)}$	< 1
Leapfrog	2	1	1	$1 + \frac{s^2}{6}$	1
Adams–Bashforth–2	3	0	$1 + \frac{s^4}{4}$	$1 + \frac{5}{12}s^2$	0
Trapezoidal	*	∞	1	$1 - \frac{s^2}{12}$	∞
Runge–Kutta–2	2	0	$1 + \frac{s^4}{8}$	$1 + \frac{s^2}{6}$	0
Magazenkov	3	0.67	$1 - \frac{s^4}{4}$	$1 + \frac{s^2}{6}$	0.67
Leapfrog–Trapezoidal	3	0.71	$1 - \frac{s^4}{4}$	$1 - \frac{s^2}{12}$	1.41
Adams–Bashforth–3	4	0.72	$1 - \frac{3}{8}s^4$	$1 + \frac{289}{720}s^4$	0.72
Adams–Moulton–3	*	0	$1 + \frac{s^4}{24}$	$1 - \frac{11}{720}s^4$	0
ABM Predictor–Corrector–3	4	0.60	$1 - \frac{19}{144}s^4$	$1 + \frac{1243}{8640}s^4$	1.20
Runge–Kutta–3	2	0.58	$1 - \frac{s^4}{24}$	$1 + \frac{s^4}{30}$	1.73
Runge–Kutta–4	$4^{\dagger}$	0.70	$1 - \frac{s^6}{144}$	$1 - \frac{s^4}{120}$	2.82

[†] A storage factor of 3 may be achieved following the algorithm of Blum (1962).

.BLE 2.2. Characteristics of the schemes listed in Table 2.1. The amplification fac
d relative phase change are for well-resolved solutions to the oscillation equation, a
= $\kappa\Delta t$. "Max s" is the maximum value of $\kappa\Delta t$ for which the solution is nonamplifyi
e storage and efficiency factors are defined in the text. No storage factor is given
plicit schemes.

3.3. Practical Measures of Dissipation and Dispersion Errors

Takacs (1985 MWR) proposed a practical measure for estimating dissipation and dispersion errors based on numerical solutions. The methods divide the total mean square error into two parts, one indicative of dissipation error and one the dispersion error.

The total mean square error is given as

$$t = \frac{1}{N}\sum_{i}^{N}(u_a - u_d)^2 \; . \qquad (27)$$

u_a is the analytical solution and u_d the numerical (discrete) solution.

It can be rewritten as (show it yourself):

$$t = s^2(u_a) + s^2(u_d) - 2rs(u_a)s(u_d) + (\overline{u}_a - \overline{u}_d)^2 \qquad (28)$$

where $s^2(u_a) = \frac{1}{N}\sum_{i}^{N}(u_a - \overline{u}_a)^2$, $s^2(u_d) = \frac{1}{N}\sum_{i}^{N}(u_d - \overline{u}_d)^2$ are the <u>variance</u> of the u_a and u_d, respectively. $\mathrm{cov}(u_a, u_d) = \frac{1}{N}\sum_{i}^{N}(u_a - \overline{u}_a)(u_d - \overline{u}_d)$ is the <u>co-variance</u> between u_a and u_d and $r = \frac{\mathrm{cov}(u_a, u_d)}{s(u_a)s(u_d)}$ is the <u>correlation coefficient</u>.

(28) can be rewritten as

$$t = [s(u_a) - s(u_d)]^2 + (\overline{u}_a - \overline{u}_d)^2 + 2(1 - r)s(u_a)s(u_d) \qquad (29)$$

Takacs definite the first two terms of the RHS of (29) as the dissipation error and the third term as the dispersion error, i.e.,

$$t_{DISS} = [s(u_a) - s(u_d)]^2 + (\overline{u}_a - \overline{u}_d)^2 \qquad (30a)$$

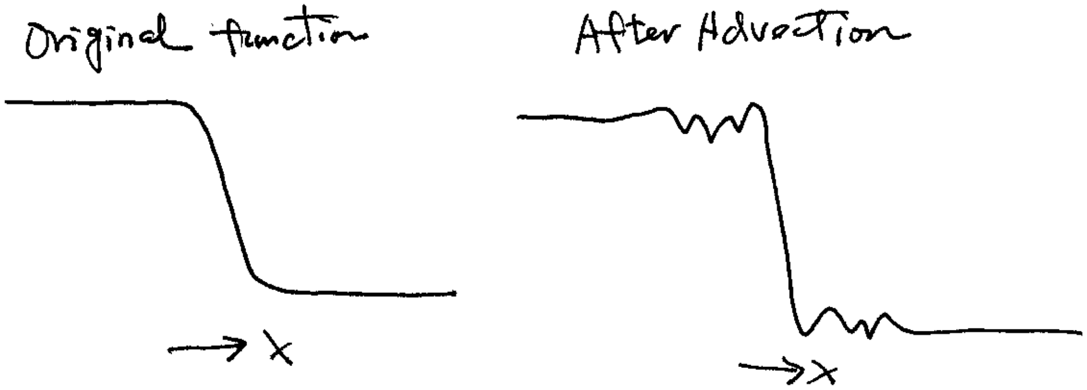
$$t_{DISP} = 2(1 - r)s(u_a)s(u_d) \qquad (30b)$$

We can see that when two wave patterns differ only in amplitude but not in phase, the their correlation coefficient ρ should be 1. According to (30a), τ_{DISP} = 0. That's a reasonable result.

4. Monotonicity of Advection Schemes

4.1. Concept of Monotonicity

When numerical schemes are used to advect a monotonic function, e.g., a monotonically decreasing function of x, the numerical solutions do not necessarily preserve the mononotic property – in fact, most of the time they do not, and the errors tend to be large near sharp gradient. This is illustrated in the following:



A few example solutions are given below:

Sample Solutions to the Inviscid Burgers' Equation

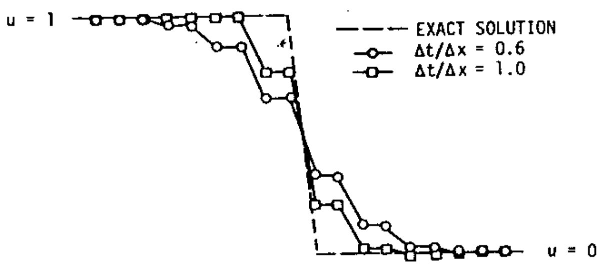


Figure 4-27 Numerical solution of Burgers' equation using Lax method.

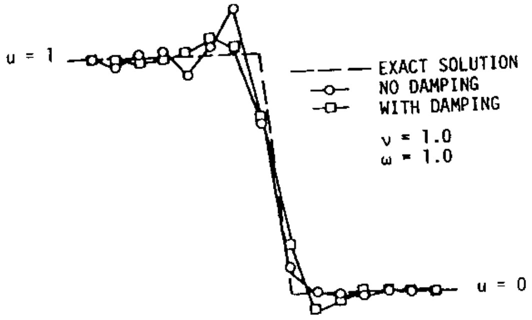


Figure 4-36 Solution for right-moving discontinuity time-centered implicit method, delta form.

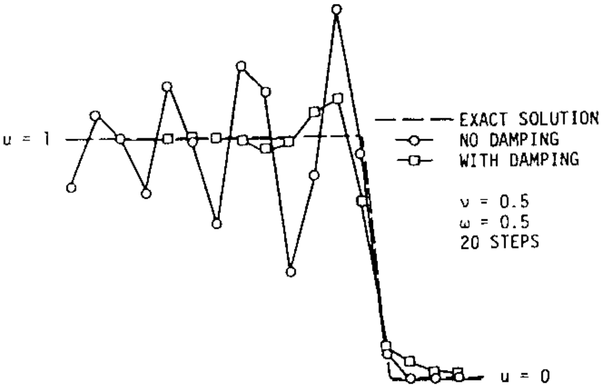


Figure 4-35 Solution of Burgers' equation using Beam-Warming (trapezoidal) method.

Monotonic numerical schemes are ones which, given an initial distribution which is monotonic before advection, produce a monotonic distribution after advection.

A consequence of this property is that monotonic schemes neither create new extrema in the solution nor amplify existing extrema.

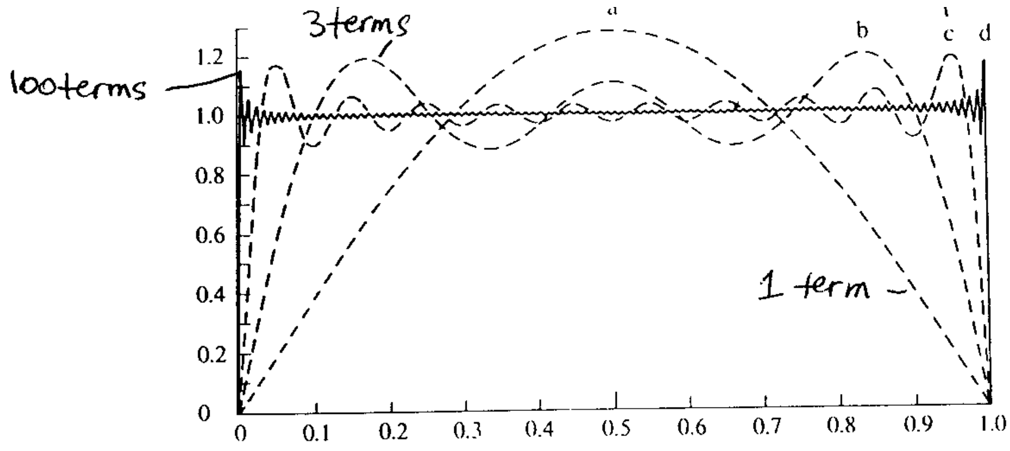
S.K. Godunov (1959) showed that no schemes having greater than first-order accuracy in space can be monotonic by construction (i.e., without using some artificial modification to ensure monotonicity). The highly dissipative upstream scheme is the classic example of a monotonic scheme.

Monotonic schemes are widely used in computational fluid dynamics because they do not allow the Gibbs Phenomenon to occur. This phenomenon results from attempting to represent a sharp gradient or discontinuity by a truncated number of waves, and always produces "undershoots and overshoots" relative to the amplitude of the initial distribution.

- These oscillations typically appear in the "wake" of a traveling wave which exhibits a sharp gradient, but do not necessarily grow in time.
- They are short waves that become noises in the solution – the damping of them results in smoothing of numerical solution.
- The oscillations can cause positive-definite fields, such as mass and water, to turn negative.

The Gibbs phenomenon is illustrated in the figure below, which shows how a square wave is represented by various numbers of waves in a Fourier expansion. Even if 100 terms are retained in the expansion, small over- and under-shoots remain. Monotonic schemes do not allow such oscillations to occur, i.e., one can think of the oscillations being removed by very selective damping.

Spectral methods use truncated spectral series to represent variable fields – they are particularly suspect to the Gibbs errors.



Monotonic schemes are often constructed by examining local features of the advected field, and adjust the advective fluxes of certain high-order schemes explicitly so that no new extrema is created in the solution.

4.2. Two basic classes of monotonic schemes

One is called the Flux-corrected transport (FCT) scheme, original proposed by Boris and Book (1973) and extended to multiple dimensions by Zalesak (1979).

With this scheme, the advective fluxes are essentially a weighted average of a lower-order monotonic scheme (usually 1st-order upwind) and a higher-order non-monotonic scheme. The idea is to use the high-order scheme as much as one can without violating the monotonicity condition. Details can also be found in Section 5.4 of Durran's book. In the ARPS, the FCT scheme is available as an option for scalar advection – it is three to four times as expensive as a regular 1st or second advection, however.

The other class is the so-called flux limiter method. With this method, the advective fluxes of a high-order scheme is directly modified (limited by a limiter) and the goal is that the total variation of the solution does not increase in time and this property is usually referred to as total variation diminishing (TVD).

The total variation of a function ϕ is defined as

$$TV(f) = \sum_{j-1}^{N-1} | f_{j+1} - f_j |$$

A TVD scheme ensures that $TV(f^{n+1}) \le TV(f^n)$.

Sweby (1984) presented a systematic derivation of the flux limiter for this class (see also Durran Section 5.5.1).

With both methods, the flux correction or limiting is done grid point by grid point – in effect, the coefficients of the finite difference schemes are solution dependent therefore they are often called non-linear schemes.

Recommended Reading: Sections 5.2.1, 5.2.2., 5.3-5.5 of Durran.

Summarizing comments:

By now, you should have realized that no scheme is perfect, although some is better than the others. When we design or choose a scheme, we need to look at a number of properties, including accuracy (in terms of amplitude and phase), stability (implicit schemes tends to be more stable), computational complexity (implicit schemes cost more to solve per step), monotonicity (can we tolerate negative water generation?), and conservation properties etc. You need also consider the problem at hand – e.g., does it contain sharp gradient that is important to your solution? What is your target computer? The computational and storage requirement are other factors to consider.

5. Multi-Dimensional Advection

Reading: Durran section 3.2.1. Smolarkievicz (1982 MWR).

Similar to the diffusion or heat transfer equations, there are three general approaches for solving multi-dimensional advection equations, namely:

1) Fully multi-dimensional methods
2) Direct extensions of 1-D schemes
3) Directional splitting methods

We will look at each in the following.

5.1. Direct Extension

Many 1-D advection schemes can be directly extended to multiple dimensions.

Multi-dimensional extension of 1-D explicit schemes often have a more restrictive stability condition.

We will look at the 2-D leapfrog centered scheme first.

For equation

$$\frac{\partial u}{\partial t} + c_x \frac{\partial u}{\partial x} + c_y \frac{\partial u}{\partial y} = 0, \qquad (31)$$

the leapfrog centered discretization is

$$u_{m,j}^{n+1} - u_{m,j}^{n-1} = -\frac{c_x \Delta t}{\Delta x}(u_{m+1,j}^{n} - u_{m-1,j}^{n}) - \frac{c_y \Delta t}{\Delta y}(u_{m,j+1}^{n} - u_{m,j-1}^{n}) \qquad (32)$$

$$t = O(\Delta x^2, \Delta y^2, \Delta t^2)$$

Let the individual wave component be

$$u_{m,j}^{n} = l^{\,n} \exp[i(km\Delta x + lj\Delta y] \qquad (33)$$

where k and l are wave number in x and y directions, respectively.

Substituting (33) into (32) and solve for λ, you can obtain (do it yourself):

$$l_{\pm} = -i\left[\frac{c_x\Delta t}{\Delta x}\sin(k\Delta x) + \frac{c_y\Delta t}{\Delta y}\sin(l\Delta y)\right] \pm \left\{1 - \left[\frac{c_x\Delta t}{\Delta x}\sin(k\Delta x) + \frac{c_y\Delta t}{\Delta y}\sin(l\Delta y)\right]\right\}^{1/2} \qquad (34)$$

Similar to the 1-D case, if

$$1 - \left[\frac{c_x\Delta t}{\Delta x}\sin(k\Delta x) + \frac{c_y\Delta t}{\Delta y}\sin(l\Delta y)\right] \geq 0, \qquad (35)$$

then $| l_{\pm} | \equiv 1$, the scheme is stable (and has not amplitude error).

Inequality (35) is satisfied when

$$\left|\frac{c_x\Delta t}{\Delta x}\sin(k\Delta x) + \frac{c_y\Delta t}{\Delta y}\sin(l\Delta y)\right| \leq 1. \qquad (36)$$

Let's consider the simpler case of $\Delta x = \Delta y = d$, and rewrite

$c_x = u_s \cos(\theta)$, $c_y = u_s \sin(\theta)$,

where u_s is the flow speed, (36) then becomes

$$\frac{u_s\Delta t}{d}\left|\cos(q)\sin(k\Delta x) + \sin(q)\sin(l\Delta y)\right| \leq 1. \qquad (37)$$

Since we want (37) to be satisfied for <u>all possible </u>waves, we choose the most stringent case of $\sin(kDx) = 1$ and $\sin(lDy) = 1$, (37) the becomes

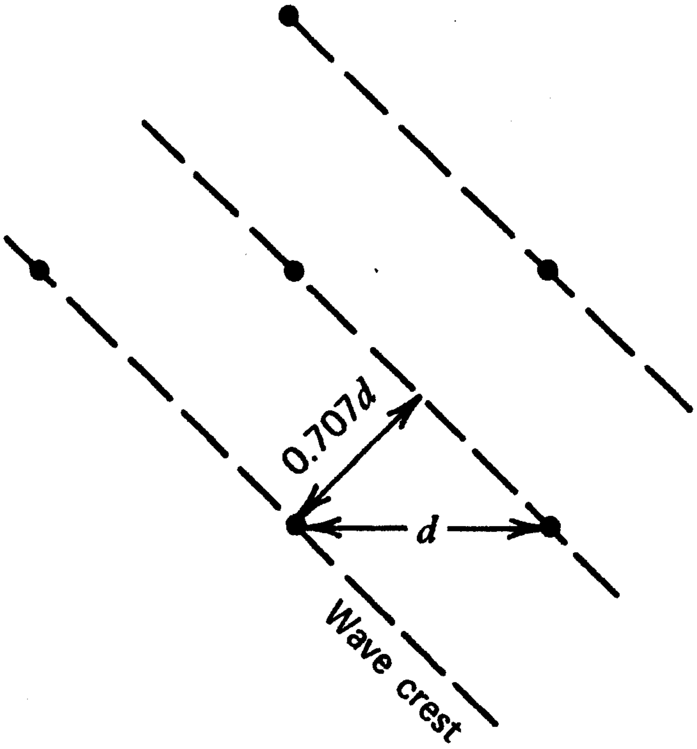
$$\frac{u_s\Delta t}{d}\left|\cos(q) + \sin(q)\right| \leq 1.$$

The maximum value of $\left|\cos(q) + \sin(q)\right|$ is $\sqrt{2}$ which occurs when $\theta = \pi/4$, the result is the stability condition for 2-D advection equation in the case of $Dx = Dy$:

$$\frac{u_s\Delta t}{d}\sqrt{2} \leq 1 \quad \text{or} \qquad \frac{u_s\Delta t}{d} \leq 0.707 \qquad (38)$$

i.e., the Courant number has to be less than 0.707, instead of 1 as we get for 1-D case.

The reason that Δt has to be about 30% smaller is explained by the following diagram:



As seen from the figure, for a wave propagating from SW to NE, the effective distance between two grid points is d/$\sqrt{2}$ instead of *d*. A wave signal cannot propagate more than one (effective) grid interval with this explicit second-order leapfrog-centered scheme for stability.

Similar reduction of time step size occurs for most other explicit schemes, including the upwind scheme.

5.2. Fully Multi-Dimensional Method

Not all direction extensions of 1-D schemes are stable, unfortunately.

Consider the Lax-Wendroff (also called Crowley) scheme we derived earlier using both second-order interpolation method (section 2.3 of Chapter 2) and the Taylor series expansion method (section 3.2.5 of this chapter):

1-D Lax-Wendroff or Crowley scheme:

$$\frac{u_i^{n+1}-u_i^n}{\Delta t} = -c\frac{u_{i+1}^n-u_{i-1}^n}{2\Delta x} + \frac{c^2\Delta t}{2}\frac{u_{i+1}^n-2u_i^n+u_{i-1}^n}{(\Delta x)^2} \qquad (39)$$

The scheme is table when |μ| ≤ 1.

Using the notion of finite-difference operators, (39) becomes

$$u_i^{n+1} = u_i^n - c\Delta t d_{2x}u^n + \frac{(c\Delta t)^2}{2}d_{xx}u^n \qquad (40).$$

Direct extension of (40) into 2-D is:

$$u_i^{n+1} = u_i^n - \Delta t(c_x d_{2x}u^n + c_y d_{2y}u^n) + \frac{(c_x\Delta t)^2}{2}d_{xx}u^n + \frac{(c_y\Delta t)^2}{2}d_{yy}u^n \qquad (41).$$

It turns out that (41) is <u>absolutely unstable</u>. This is because the cross-derivative terms are neglected!

To see it, we need to go back to original derivation of the Lax-Wendroff scheme:

$$u^{n+1} = u^n + \Delta t u_t + \frac{1}{2}(\Delta t)^2 u_{tt} + O(\Delta t^3) \qquad (42)$$

Use $\quad u_t = -c_x\, u_x - c_y\, u_y$

and $\quad \underline{u_{tt} = -c_x\, u_{tx} - c_y\, u_{ty} = c_x^2\, u_{xx} + c_y^2\, u_{yy} + 2c_xc_y\, u_{xy}}$,

and replace the spatial derivatives with the corresponding finite differences, (42) becomes

$$u_i^{n+1} = u_i^n - \Delta t(c_x d_{2x}u^n + c_y d_{2y}u^n) + \frac{(c_x\Delta t)^2}{2}d_{xx}u^n + \frac{(c_y\Delta t)^2}{2}d_{yy}u^n + (c_xc_y\Delta t^2)d_{xy}u^n \qquad (43)$$

Clearly, the last term on the RHS is additional, compared to (41).

Note that we can also obtain (43) using the characteristics method plus quardratic interpolation, as long as all terms in the second-order 2-D polynomial are retained.

Equation (43) is an example of <u>fully multidimensional</u> scheme, which is different from the direct extension of 1-D counterpart.

Smolarkiewicz (1982 MWR) discuss the MD Crowley scheme in details (handout).

5.3. Directional Splitting

It turned out that by using directionally splitting method (i.e., applying 1-D scheme in one direction at a time), the effect of cross-derivative terms can also be retained and a stable scheme result.

The algorithms is

$$(u_{i,j}^{n+1})^* = u_{i,j}^n - c_x \Delta t d_{2x} u_{i,j}^n + \frac{(c_x \Delta t)^2}{2} d_{xx} u_{i,j}^n \qquad (44a)$$

$$u_{i,j}^{n+1} = (u_{i,j}^{n+1})^* - c_y \Delta t d_{2y} (u_{i,j}^{n+1})^* + \frac{(c_y \Delta t)^2}{2} d_{yy} (u_{i,j}^{n+1})^* \qquad (44b)$$

In this case, we preserve the stability of each step and $\lambda = \lambda_x \lambda_y$.

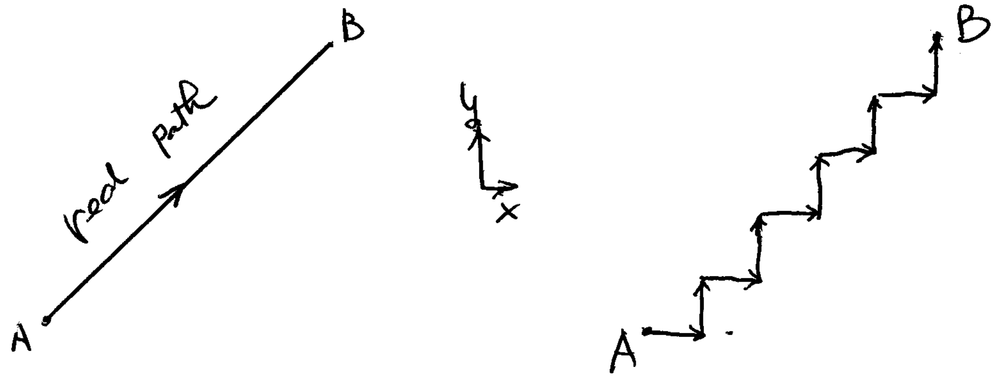
With the above scheme, we have

Advantages:

1. 1-D advection is straightforward – properties of schemes are well understood.
2. The time step constraint is not as severe as for true multi-dimensional problems.

Disadvantages:

1. We implicitly assume that features move obliquely to the grid may be represented as a series of orthogonal steps in the coordinate directions:



In an implicit scheme, where the time step can be large, these errors can be substantial.

2. The biggest disadvantage is that splitting introduces an O(Δt^2) error in the form of a spurious source term. To see this, consider the 2-D advection being solved using directional splitting upstream advection:

$$u_t + Uu_x + Vu_y = 0 \tag{45}$$

where U=U(x,y) >0 and V(x,y) >0 .

Writing this as a direct extension of the upwind scheme in 1-D, we have

$$u^{n+1} = u^n - \Delta t U d_{-x} u - \Delta t V d_{-y} u \, . \tag{46}$$

The directional splitting version is

$$u^* = u^n - \Delta t U d_{-x} u \tag{47a}$$

$$u^{n+1} = u^* - \Delta t V d_{-y} u^* \tag{47b}$$

Substitute (47a) into (47b), we obtain a single step scheme

$$u^* = u^n - \Delta t U d_{-x} u - \Delta t V d_{-y} u + UV \Delta t^2 d_{-x} u d_{-y} u + V \Delta t^2 d_{-x} u d_{-y} U \, . \tag{48}$$

We can see that the last term on RHS of (48) is actually spurious and is not zero when U is not constant.