3.1 Introduction

We have already analyzed the accuracy and truncation error for finite difference quotients. Now we analyze the truncation error and accuracy of a finite-difference scheme, which is defined as a finite-difference equation that approximates, term-by-term, a differential equation. Using the methods outlined in Chapter 2, we can find approximations to each term of a differential equation, and we have already seen that the error of such an approximation can be made as small as desired, almost effortlessly. This is not our goal, however. Our goal is to find an approximation to the solution of the differential equation. You might think that if we have a finite-difference equation, $F$, that is constructed by writing down a good approximation to each term of a differential equation, $D$, then the solution of $F$ will be a useful approximation to the solution of $D$. Wouldn't that be nice? Unfortunately, it isn't necessarily true.

As an example, we will analyze the one-dimensional advection equation, given by

$$\frac{\partial A}{\partial t} + c \frac{\partial A}{\partial x} = 0,$$

(1)

where $A = A(x, t)$. The physical meaning of (1) is that $A$ remains constant at the position of a particle that moves in the $x$-direction with speed $c$. We can interpret $A$ as a property of the particle. We will assume for now that $c$ is a constant. Eq. (1) is a first-order linear partial differential equation with a constant coefficient, namely $c$. It looks harmless, but it causes no end of trouble, as we will see.

Suppose that

$$A(x,0) = F(x) \text{ for } -\infty < x < \infty.$$

(2)
This is an “initial condition.” Our goal is to determine \( u(x,t) \). This is a simple example of an initial value problem. We first work out the analytic solution, for later comparison with our numerical solution. Define

\[
\xi \equiv x - ct , \text{ so that } \left( \frac{\partial x}{\partial t} \right)_\xi = c .
\]

(3)

Suppose that we are at position \( x \) and time \( t \). We can interpret \( \xi \) as our previous position (i.e., our previous value of \( x \)) at time \( t = 0 \). Using the chain rule, we write

\[
\left( \frac{\partial A}{\partial x} \right)_\xi = \left( \frac{\partial A}{\partial x} \right)_x + \left( \frac{\partial A}{\partial t} \right)_x \left( \frac{\partial t}{\partial x} \right)_\xi \\
= \left( \frac{\partial A}{\partial x} \right)_x + \left( \frac{\partial A}{\partial t} \right)_x \cdot 1 \\
= \left( \frac{\partial A}{\partial x} \right)_x + \left( \frac{\partial A}{\partial t} \right)_x \cdot \frac{1}{c} \\
= 0 .
\]

(4)

The first line of (4) can be understood by reference to Fig. 3.1. The last line comes by use of (1). Similarly,

\[
\left( \frac{\partial A}{\partial t} \right)_\xi = \left( \frac{\partial A}{\partial t} \right)_x + \left( \frac{\partial A}{\partial x} \right)_x \left( \frac{\partial x}{\partial t} \right)_\xi \\
= \left( \frac{\partial A}{\partial t} \right)_x + \left( \frac{\partial A}{\partial x} \right)_x \cdot c \\
= 0 .
\]

(5)
The last equality follows from (1). We conclude that

\[ A = f(\xi) \]  

(6)
is the general solution to (1). This means that \( A \) depends only on \( \xi \).

The initial condition is

\[ \xi = x \text{ and } A(x) = f(x) \text{ at } t = 0, \]

(7)
i.e., the shape of \( f(\xi) \) is determined by the initial condition. Eq. (6) means that \( A \) is constant along the line \( \xi = \text{constant} \). In order to satisfy the initial condition, we chose \( f \equiv F \) [see Eq. (2)]. Referring to (6), we see that \( A(\xi) = F(\xi) = F(x - ct) \) is the solution to (1) that satisfies the initial condition (7). An initial value simply “moves along” the lines of constant \( \xi \), which are called characteristics. The initial shape of \( A(x) \), namely \( F(x) \), is just carried along by the wind. From a physical point of view this is obvious. Partial differential equations whose solutions are constant along characteristics are called “hyperbolic” equations. The advection equation is hyperbolic. Further discussion is given later.

### 3.2 The upstream scheme

Keeping in mind the exact solution, we now investigate one possible numerical solution of (1). We construct a grid, as in Fig. 3.2. One of the infinitely many possible finite difference approximations to (1) is

\[ \frac{A_j^{n+1} - A_j^n}{\Delta t} + c \left( \frac{A_j^n - A_{j-1}^n}{\Delta x} \right) = 0. \]

(8)

Here we have used the forward difference quotient in time and the backward difference quotient in space. If we know \( A_j^n \) at some time level \( n \) for all \( j \), then we can use (8) to compute \( A_j^{n+1} \) at the next time level, \( n + 1 \). If \( c > 0 \), (8) is called the “upstream” scheme. The upstream scheme is one-sided or asymmetric in both space and time. It seems naturally suited to modeling advection, in which air comes from one side and goes to the other, as time passes by. The upstream scheme has some serious weaknesses, but it also has some very useful properties. It is a scheme worth remembering. That’s why I put it in a box in Eq. (8).
Because

$$\frac{A_j^{n+1} - A_j^n}{\Delta t} \rightarrow \frac{\partial A}{\partial t} \text{ as } \Delta t \rightarrow 0,$$

(9)

and

$$\frac{A_j^n - A_{j-1}^n}{\Delta x} \rightarrow \frac{\partial A}{\partial x} \text{ as } \Delta x \rightarrow 0,$$

(10)

we conclude that (8) does approach (1) as $\Delta t$ and $\Delta x$ both approach zero. In view of (9) and (10), it may seem obvious that the solution of (8) approaches the solution of (1) as $\Delta x \rightarrow 0$ and $\Delta t \rightarrow 0$. This “obvious” conclusion is not necessarily true.

3.3 The truncation error of a finite-difference scheme.

Letting $A(x,t)$ denote the (exact) solution of the differential equation, we see that $A(j\Delta x,n\Delta t)$ is the value of this exact solution at the discrete point $(j\Delta x,n\Delta t)$ on the grid shown in Fig. 3.2, while $A_j^n$ is the “exact” solution of a finite-difference equation, at the same point.

Figure 3.2: A grid for the solution of the one-dimensional advection equation.
general, \( A_j^n \neq A(\Delta x, n\Delta t) \). We wish that they were equal!

A measure of the accuracy of the finite-difference scheme can be obtained by substituting the solution of the differential equation into the finite-difference equation. For the upstream scheme given by (78), we get

\[
\frac{A[j\Delta x, (n+1)\Delta t] - A(j\Delta x, n\Delta t)}{\Delta t} + c \left( \frac{A(j\Delta x, n\Delta t) - A((j-1)\Delta x, n\Delta t)}{\Delta x} \right) = \varepsilon
\]

where \( \varepsilon \) is called the “truncation error” of the scheme. The truncation error of the scheme is a measure of how accurately the solution \( A(x, t) \) of the original differential equation (1), satisfies the finite-difference equation, (8). It is far from a perfect measure of accuracy, however, as will become evident.

Because \( A_j^n \) is defined only at discrete points, it is not differentiable, and so we cannot substitute \( A_j^n \) into the differential equation. Because of this, we cannot measure how accurately \( A_j^n \) satisfies the differential equation.

If we obtain the terms of (11) from Taylor Series expansion of \( A(x, t) \) about the point, \( (j\Delta x, n\Delta t) \) and use the fact that \( A(x, t) \) satisfies (1), we find that

\[
\varepsilon = \left( \frac{1}{2!} \frac{\partial^2 A}{\partial t^2} + \cdots \right) + c \left( \frac{-1}{2!} \Delta x \frac{\partial^2 A}{\partial x^2} + \cdots \right).
\]

We say this is a “first-order scheme" because the lowest powers of \( \Delta t \) and \( \Delta x \) in (12) are 1. The notations \( O(\Delta t, \Delta x) \) or \( O(\Delta t) + O(\Delta x) \) can be used to express this. The upstream scheme is first-order accurate in both space and time. A scheme is said to be “consistent" with the differential equation if the truncation error of the scheme approaches zero as \( \Delta t \) and \( \Delta x \) approach zero. The upstream scheme is consistent. Consistency is necessary for a good scheme, but it is nowhere near sufficient.

### 3.4 Discretization error and convergence

There are two sources of error in a numerical solution. One is the round-off error, which is the difference of a numerical solution from the “exact” solution of the finite difference equation, \( A_j^n \), and is a property of the machine being used (and to some extent the details of the program). The other source of error is the discretization error defined by \( A_j^n - A(j\Delta x, n\Delta t) \).
Round-off error sometimes causes problems, but usually it doesn’t, and we will not worry about it in this course.

The truncation error, discussed in the last section, can be made as small as desired by making $\Delta x$ and $\Delta t$ smaller and smaller, so long as the scheme is consistent and $A(x,t)$ is a smooth function. A decrease in the truncation error does not necessarily ensure that the discretization error becomes small, however. This is demonstrated below.

We now analyze how the discretization error changes as the grid is refined (i.e., as $\Delta t$ and $\Delta x \to 0$). If the discretization error approaches zero, then we say that the solution converges. Fig. 3.3 illustrates a situation in which accuracy is increased but the solution nevertheless does not converge. The thin diagonal line in the figure shows the characteristic along which $A$ is “carried,” i.e., $u$ is constant along the line. This is the exact solution. To work out the numerical approximation to this solution, we first choose $\Delta x$ and $\Delta t$ such that the grid points are the dots in the figure. The set of grid points carrying values of $A$ on which $A_j^n$ depends is called the “domain of dependence.” The shaded area in the figure shows the domain of dependence for the upstream scheme, (8).

![Diagram of the domain of dependence](image)

Figure 3.3: The shaded area represents the “domain of dependence” of the solution of the upstream scheme at the point $(j\Delta x, n\Delta t)$.

We could increase the accuracy of the scheme by cutting $\Delta x$ and $\Delta t$ in half, but the
domain of dependence does not change, no matter how dense the grid is, as long as the ratio \( \frac{c\Delta t}{\Delta x} \) remains the same. This is a clue that \( \frac{c\Delta t}{\Delta x} \) is an important quantity.

Suppose that the line through the point \((j\Delta x,n\Delta t)\), i.e., \(x - ct = x_0\), where \(x_0\) is a constant, does not lie in the domain of dependence. This is the situation shown in Fig. 3.3. In general, there is no hope of obtaining smaller discretization error, no matter how small \(\Delta x\) and \(\Delta t\) become, as long as \(\frac{c\Delta t}{\Delta x}\) is unchanged, because the true solution depends only on the initial value of \(A\) at the single point \((x_0,0)\) which cannot influence \(A^n_j\). You could change \(A(x_0,0)\) [and hence \(A(j\Delta x,n\Delta t)\)], but the computed solution would remain the same. In such a case, the error of the solution usually will not be decreased by refining the grid. This illustrates that if the value of \(c\) is such that \(x_0\) lies outside of the domain of dependence, it is not possible for the solution of the finite-difference equation to approach the solution of the differential equation, no matter how fine the mesh becomes. The finite difference equation converges to the differential equation, but the solution of the finite-difference equation does not converge to the solution of the differential equation. The truncation error goes to zero, but the discretization error does not.

A finite-difference scheme for which the discretization error can be made small for any initial condition is called convergent. The discussion above shows that

\[
0 \leq \frac{c\Delta t}{\Delta x} \leq 1
\]

is a necessary condition for convergence of the upstream scheme.

Notice that if \(c\) is negative (giving what we might call a “downstream” scheme), it is impossible to satisfy (13). Of course, for \(c < 0\) we can use

\[
\frac{A_j^{n+1} - A_j^n}{\Delta t} + c \left( \frac{A_j^{n+1} - A_j^n}{\Delta x} \right) = 0,
\]

in place of (8). Our computer program can have an “if-test” that checks the sign of \(c\), and uses (8) if \(c \geq 0\), and [(14) if \(c < 0\)]. This is bad, though, because if-tests can cause slow execution on certain types of computers, and besides, if-tests are ugly. If we define
\[ c_+ \equiv \frac{c + |c|}{2} \geq 0, \quad \text{and} \quad c_- \equiv \frac{c - |c|}{2} \leq 0, \]

then the upstream scheme can be written as

\[
\frac{A_j^{n+1} - A_j^n}{\Delta t} + c_+ \left( \frac{A_j^n - A_{j+1}^n}{\Delta x} \right) + c_- \left( \frac{A_{j+1}^n - A_j^n}{\Delta x} \right) = 0. 
\]

This form avoids the use of if-tests and is also convenient for use in pencil-and-paper analysis, as discussed later.

In summary: Truncation error measures the accuracy of an approximation to a differential operator or operators. It is a measure of the accuracy with which the terms of a differential equation have been approximated. Discretization error measures the accuracy with which the solution of the differential equation has been approximated. Minimizing the truncation error is usually easy. Minimizing the discretization error can be much harder.

### 3.5 Interpolation and extrapolation

Referring back to (8), we can rewrite the upstream scheme as

\[
A_j^{n+1} = A_j^n (1 - \mu) + A_{j+1}^n \mu, 
\]

where

\[
\mu \equiv \frac{c \Delta t}{\Delta x}. 
\]
This scheme has the form of either an interpolation or an extrapolation, depending on the value of $\mu$. To see this, refer to Figure 3.4. Along the line plotted in the figure

\[ A = A_{j-1}^n - \left( \frac{x - x_{j-1}}{x_j - x_{j-1}} \right) \left( A_j^n - A_{j-1}^n \right) \]

\[ = A_j^n \left[ 1 - \left( \frac{x - x_{j-1}}{x_j - x_{j-1}} \right) \right] + A_{j-1}^n \left( \frac{x - x_{j-1}}{x_j - x_{j-1}} \right), \]

(19)

which has the same form as our scheme if we identify

\[ A \equiv A_{j+1}^{n+1} \text{ and } \mu \equiv \frac{x - x_{j-1}}{x_j - x_{j-1}}. \]

(20)

For $0 \leq \mu \leq 1$ we have interpolation. For $\mu < 0$ or $\mu > 1$ we have extrapolation. For the case of interpolation, $A_{j+1}^{n+1}$ will be intermediate in value between $A_{j-1}^n$ and $A_j^n$. One implication of this is that if $A_{j-1}^n$ and $A_j^n$ are both positive, then $A_{j+1}^{n+1}$ will also be positive. For the case of extrapolation, $A_{j+1}^{n+1}$ will lie outside the range of $A_{j-1}^n$ and $A_j^n$.

Both interpolation and extrapolation are used extensively in atmospheric modeling. Much more discussion of both is given later.
3.6 Stability

We now investigate the behavior of the discretization error $|A^n_j - A(jΔx, nΔt)|$ as $n$ increases, for fixed $Δx$ and $Δt$. Does the error remain bounded for any initial condition? If so, the scheme is said to be stable; otherwise it is unstable.

In most physical problems, the true solution is bounded, at least for finite $t$, so that the solution of the scheme is bounded if the scheme is stable.

There are at least three ways in which the stability of a scheme can be tested. These are: 1) the direct method, 2) the energy method, and 3) von Neumann's method.

The direct method establishes stability by demonstrating that the largest absolute value of $A^{n+1}_j$ on the grid does not increase with time. As an illustration of the direct method, consider the upstream scheme, as given by (8). Note that $A^{n+1}_j$ is a weighted mean of $A^n_j$ and $A^n_{j-1}$. If $0 ≤ μ ≤ 1$ (the necessary condition for convergence according to (13)), we may write

$$|A^{n+1}_j| ≤ |A^n_j|(1 - μ) + |A^n_{j-1}|μ.$$  \hspace{1cm} (21)

Therefore,

$$\max_{(j)}|A^{n+1}_j| ≤ \max_{(j)}|A^n_j|(1 - μ) + \max_{(j)}|A^n_{j-1}|μ$$

$$= \max_{(j)}|A^n_j|.$$ \hspace{1cm} (22)

where $\max_{(j)}$ denotes the largest value at any point on the grid. The second line of (22) follows because $\max_{(j)}|A^n_j| = \max_{(j)}|A^n_{j-1}|$. Eq. (22) demonstrates that the scheme is stable provided that our assumption

$$0 ≤ μ ≤ 1$$ \hspace{1cm} (23)

is satisfied. This means that the solution remains bounded for all time for $0 ≤ μ ≤ 1$, and so it is a sufficient condition for stability. For the upstream scheme, a sufficient condition for stability has turned out to be the same as the necessary condition for convergence. In other words, if the scheme is convergent it is stable, and vice versa.
In the solution of the exact advection equation, the maxima and minima of \( A(x,t) \) never change. They are just carried along to different spatial locations. So, for the exact solution, the equality in (22) would hold.

Eq. (22), the sufficient condition for stability, is actually obvious from (17), because when \( 0 \leq \mu \leq 1 \), \( A_{j+1}^n \) is obtained by linear interpolation in space, from the available \( A_j^n \) to the point \( x = j\Delta x - c\Delta t \). This is reasonable, because the nature of advection is such that the time rate of change at a point is closely related to the spatial variations in the neighborhood of that point.

The direct method cannot be used to check the stability of complex schemes. The second method, which is called the energy method, is more widely applicable, even for some nonlinear equations, and is quite important in practice. We illustrate it here by means of application to the upstream scheme. With the energy method we ask: “Is \( \sum_j (A_j^n)^2 \) bounded after an arbitrary number of time steps?” Here the summation is over the entire domain. If the sum is bounded, then each \( A_j^n \) must also be bounded. Whereas in the direct method we checked, \( \max_j |A_{j+1}^n| \), with the energy method we check \( \sum_j (A_j^n)^2 \). The two approaches are somewhat similar.

Returning to (17), squaring both sides, and summing over the domain, we obtain

\[
\sum_j (A_{j+1}^n)^2 = \sum_j \left[ (A_j^n)^2 (1 - \mu)^2 + 2\mu(1 - \mu)A_j^nA_{j-1}^n + \mu^2(A_{j-1}^n)^2 \right] = (1 - \mu)^2 \sum_j (A_j^n)^2 + 2\mu(1 - \mu)\sum_j A_j^nA_{j-1}^n + \mu^2\sum_j (A_{j-1}^n)^2 .
\]

(24)

If \( A \) is periodic in \( x \), then

\[
\sum_j (A_{j-1}^n)^2 = \sum_j (A_j^n)^2 .
\]

(25)

By starting from \( \sum_j (A_j^n - A_{j-1}^n)^2 \geq 0 \), and using (25), we can show that

\[
\sum_j A_j^nA_{j-1}^n \leq \sum_j (A_j^n)^2 .
\]

(26)

Another way to derive (26) is to use (25) and Schwartz’s inequality (e.g., Arfken, p. 527), i.e.,
\[(\sum_j a_j b_j)^2 \leq (\sum_j a_j^2)(\sum_j b_j^2),\]  
(27)

which holds for any sets of \(a\)'s and \(b\)'s. (An interpretation of Schwartz's inequality is that the dot product of two vectors is less than or equal to the product of the magnitudes of the two vectors.)

Use of (25) and (26) in (24) gives

\[\sum_j (A_j^{n+1})^2 \leq \sum_j (A_j^n)^2,\]
(28)

provided that \(\mu(1 - \mu) \geq 0\), which is equivalent to \(0 \leq \mu \leq 1\). It follows that

\[\sum_j (A_j^{n+1})^2 \leq \sum_j (A_j^n)^2,\] provided that \(0 \leq \mu \leq 1.\)
(29)

The conclusion is the same as that obtained using the direct method, i.e., \(0 \leq \mu \leq 1\) is a sufficient condition for stability.

A very powerful tool for testing the stability of linear partial difference equations with constant coefficients is von Neumann's method. It is the method that will be used most often in this course. Solutions to linear partial differential equations can be expressed as superposition of waves, by means of Fourier series. Von Neumann's method simply tests the stability of each Fourier component.

To illustrate von Neumann's method, we return first to the exact advection equation, (1). We assume for simplicity that the domain is infinite. First, we look for a solution with the wave form

\[A(x,t) = \text{Re}[\hat{A}(t)e^{ikx}],\]
(30)

where \(\hat{A}(t)\) is the amplitude of the wave. Here we consider a single wave number, for simplicity, but in general we could replace the right-hand side of (30) by a sum over all relevant wave numbers. Substituting (30) into (1), we find that

\[\frac{d\hat{A}}{dt} + ikc\hat{A} = 0.\]
(31)
By Fourier expansion we have converted the partial differential equation (1) into an ordinary differential equation, (31), whose solution is

$$\hat{A}(t) = \hat{A}(0)e^{-ikt},$$

(32)

where $\hat{A}(0)$ is the initial value of $\hat{A}$. The sign convention used here implies that for $c > 0$ the signal will move towards larger $x$. Substituting (32) back into (30), we find that the full solution to (31) is

$$A(x,t) = \text{Re}\left[\hat{A}(0)e^{ik(x-ct)}\right],$$

(33)

provided that $c =$ constant.

For a finite-difference equation, the assumed form of the solution, given by Eq. (30), is replaced by

$$A^n_j = \text{Re}\left[\hat{A}^n e^{ik\Delta x}\right].$$

(34)

Then $|\hat{A}^n|$ is the amplitude of the wave at time-level $n$. Note that the shortest resolvable wave, with $L = 2\Delta x$, has $k\Delta x = \pi$, while longer waves have $k\Delta x < \pi$, so there is never any need to consider $k\Delta x > \pi$.

Now define $\lambda$, which may be complex, by

$$\hat{A}^{n+1} = \lambda \hat{A}^n,$$

(35)

so that $|\hat{A}^{n+1}| = |\lambda||\hat{A}^n|$. We call $\lambda$ the “amplification factor.” In general, $\lambda$ depends on $k$, so we could write $\lambda_k$, but usually we suppress that urge for the sake of keeping the notation simple. The value of $\lambda$ also depends on the size of the time step. As shown below, we can work out the form of $\lambda$ for a particular finite-difference scheme.

Before doing that, we use the definition (35) to compute the effective value of $\lambda$ for the exact solution to the differential equation. From (32), we find that
for the exact advection equation \( \hat{A}(t + \Delta t) \equiv e^{i k c \Delta t} \hat{A}(t) \),

\( (36) \)

from which it follows “by inspection” that

for the exact advection equation \( \lambda = e^{i k c \Delta t} \).

\( (37) \)

Eq. (37) implies that

for the exact advection equation \( |\lambda| = 1 \),

\( (38) \)

regardless of the value of \( \Delta t \). It will be shown later that for processes other than advection the exact value of \( |\lambda| \) can differ from 1, and can depend on \( \Delta t \).

From (35) we see that after \( n \) time steps, starting from \( n = 0 \), the solution will be

\[ \hat{A}^n = \hat{A}^0 \lambda^n. \]

\( (39) \)

From (39), the requirement for stability, i.e., that the solution remains bounded after arbitrarily many time steps, implies that

\[ |\lambda| \leq 1. \]

\( (40) \)

This is the condition for stability. Therefore, to evaluate the stability of a finite-difference scheme, using von Neumann's method, we need to work out the value of \( |\lambda| \) for that scheme.

We now work out the form of \( |\lambda| \) for the particular case of the upstream scheme, as given by (8). Substituting (34) into (8) leads to

\[ \frac{\hat{A}^{n+1} - \hat{A}^n}{\Delta t} + c \left( 1 - e^{-i k c \Delta x} \frac{\Delta x}{\Delta x} \right) \hat{A}^n = 0. \]

\( (41) \)
Notice that the true advection speed, \(c\), is multiplied, in (41), by the factor \(\left(\frac{1-e^{-ik\Delta x}}{\Delta x}\right)\).

Comparing with (31), we see that \(\left(\frac{1-e^{-ik\Delta x}}{\Delta x}\right)\) is “taking the place” of \(ik\) in the exact solution. In fact, you should be able to show that,

\[
\lim_{\Delta x \to 0} \left(\frac{1-e^{-ik\Delta x}}{\Delta x}\right) = ik.
\]

This is a clue that the upstream scheme gives an error in the advection speed. We return to this point later.

For now, we use the definition of \(\lambda\), i.e., (35), together with (41), to infer that

\[
\lambda = 1 - \mu (1 - \cos k\Delta x + i \sin k\Delta x).
\]

Note that \(\lambda\) is complex. Computing the square of the modulus of both sides of (43), we obtain

\[
|\lambda|^2 = 1 + 2\mu(\mu - 1)(1 - \cos k\Delta x).
\]

As an example, for \(\mu = \frac{1}{2}\), (44) reduces to

\[
|\lambda|^2 = \frac{1}{2}(1 - \cos k\Delta x).
\]

According to (45), the amplification factor \(|\lambda|\) depends on the wave number, \(k\). Fig. 3.5 shows that the upstream scheme damps for \(0 \leq \mu \leq 1\) and is unstable for \(\mu < 0\) and \(\mu > 1\). For \(\mu\) close to zero, the scheme is close to neutral, but many time steps are needed to complete a given simulation. For \(\mu\) close to one, the scheme is again close to neutral, but it is also close to instability. We are therefore motivated to choose intermediate values of \(\mu\), but they strongly damp the shortest modes. No matter what we do, there are problems with the scheme.
Although \( \lambda \) depends on \( k \), it does not depend on \( x \) (i.e., on \( j \)) or on \( t \) (i.e., on \( n \)). Why not? The reason is that our “coefficient,” \( c \), has been assumed to be independent of \( x \) and \( t \). Of course, in realistic problems the advecting current varies in both space and time. We normally apply von Neumann’s method to idealized versions of our equations, in which the various coefficients, such as \( c \), are treated as constants. As a result, von Neumann’s method can “miss” instabilities that arise from variations of the coefficients. It can also miss instabilities that arise from nonlinearity. The energy method does not suffer from these limitations.

It is very important to keep in mind that von Neumann’s method can only analyze the stability of a linearized version of the equation. In fact, the equation has to be linear and with constant coefficients. This is an important weakness of the method, because the equations used in numerical models are typically nonlinear and/or have spatially variable coefficients – if this were not true, we would solve them analytically! The point is that von Neumann’s method can sometimes tell us that a scheme is stable, when in fact it is unstable. In such cases, the instability arises from nonlinearity and/or through the effects of spatially variable coefficients. This kind of instability will be discussed in a later chapter. If von Neumann’s method says that a scheme is unstable, you can be confident that it is really is unstable.

As mentioned above, the full solution for \( A_j^n \) can be expressed as a Fourier series. For simplicity, we assume that the solution is periodic in \( x \), with period \( L_0 \). You might want to think of \( L_0 \) as the distance around a latitude circle. Then \( A_j^n \) can be written as

\[
A_j^n = \frac{1}{L_0} \sum_{m=-\infty}^{\infty} \hat{A}_m e^{i2\pi jm/L_0}
\]
\[ A^n_j = \text{Re} \left[ \sum_{m=-\infty}^{\infty} \hat{A}_m e^{imk_0 j \Delta x} \right] \]
\[ = \text{Re} \left[ \sum_{m=-\infty}^{\infty} \hat{A}_m e^{imk_0 j \Delta x} (\lambda_m)^n \right], \]  
(46)

where

\[ k \equiv mk_0, \]  
(47)

\[ k_0 \equiv \frac{2\pi}{L_0}, \]  
(48)

and \( m \) is an integer, which is analogous to what we call the “zonal wave number” in large-scale dynamics. We can interpret \( k_0 \) as the lowest wave number in the solution, so if \( L_0 \) is the distance around a latitude circle then the lowest wave number \( k_0 \) corresponds to one “high” and one “low” around a latitude circle. In (46), the summation has been formally taken over all integers, although of course only a finite number of \( m \)’s can be used in a real application. We can interpret \( |\lambda_m| \) as the amplification factor for mode \( m \). We can write

\[ |A^n_j| \leq \sum_{m=-\infty}^{\infty} |\hat{A}_m e^{imk_0 j \Delta x} (\lambda_m)^n| \]
\[ \leq \sum_{m=-\infty}^{\infty} |\hat{A}_m e^{imk_0 j \Delta x} (\lambda_m)^n| \]
\[ = \sum_{m=-\infty}^{\infty} |\hat{A}_m| |\lambda_m|^n. \]  
(49)

If \( |\lambda_m| \leq 1 \) is satisfied for all \( m \), then

\[ |A^n_j| \leq \sum_{m=-\infty}^{\infty} |\hat{A}_m| \]  
(50)

Therefore, \( |A^n_j| \) will be bounded provided that \( \sum_{m=-\infty}^{\infty} \hat{A}_m e^{imk_0 j \Delta x} \), which gives the initial condition, is an absolutely convergent Fourier series. The point is that \( |\lambda_m| \leq 1 \) for all \( m \) is sufficient for
stability. It is also necessary, because if $|\lambda_m| > 1$ for a particular $m$, say $m = m_1$, then the solution for the initial condition $u_{m_1} = 1$ and $u_m = 0$ for all $m \neq m_1$ is unbounded.

From (107), $\lambda_m$ for the upstream scheme is given by

$$\lambda_m = 1 - \mu \left(1 - \cos mk_0 \Delta x + i \sin mk_0 \Delta x\right).$$

(51)

The amplification factor is

$$|\lambda_m| = \sqrt{1 + 2\mu(\mu - 1)(1 - \cos mk_0 \Delta x)}.$$

(52)

From (52) we can show that $|\lambda_m| \leq 1$ holds for all $m$, if and only if $\mu(\mu - 1) \leq 0$, which is equivalent to $0 \leq \mu \leq 1$. This is the necessary and sufficient condition for the stability of the scheme.

To explicitly allow for a finite periodic domain, the upstream scheme can be written in matrix form as

$$
\begin{bmatrix}
A_{i+1}^n \\
A_{j+1}^n \\
\vdots \\
A_{j-1}^n \\
A_j^{n+1}
\end{bmatrix}
= 
\begin{bmatrix}
1 - \mu & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & \mu \\
\mu & 1 - \mu & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 \\
0 & \mu & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 \\
0 & 0 & \cdots & 1 - \mu & 0 & 0 & \cdots & 0 & 0 \\
0 & 0 & \cdots & \mu & 1 - \mu & 0 & \cdots & 0 & 0 \\
0 & 0 & \cdots & 0 & \mu & 1 - \mu & \cdots & 0 & 0 \\
0 & 0 & \cdots & 0 & 0 & \mu & \cdots & 0 & 0 \\
0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 1 - \mu & 0 \\
0 & 0 & \cdots & 0 & 0 & 0 & \cdots & \mu & 1 - \mu
\end{bmatrix}
\begin{bmatrix}
A_1^n \\
A_2^n \\
\cdots \\
A_{j-1}^n \\
A_j^n
\end{bmatrix},

(53)

or

$$
\begin{bmatrix}
A_j^{n+1}
\end{bmatrix} = [M] \begin{bmatrix}
A_j^n
\end{bmatrix},

(54)

where $[M]$ is the matrix written out on the right-hand side of (53). In writing (53), the cyclic boundary condition.
\[ A_j^{n+1} = (1 - \mu) A_j^n + \mu A_i^n \]  

(55)

has been assumed, and this leads to the \( \mu \) in the top-right corner of the matrix. From the definition of \( \lambda \), (35), we can write

\[
\begin{bmatrix}
A_1^{n+1} \\
A_2^{n+1} \\
\vdots \\
A_{j-1}^{n+1} \\
A_j^{n+1} \\
A_{j+1}^{n+1} \\
\vdots \\
A_{J-1}^{n+1} \\
A_J^{n+1}
\end{bmatrix} = 
\lambda \begin{bmatrix}
0 & 0 & 0 & \cdots & 0 \\
0 & \lambda & 0 & \cdots & 0 \\
0 & 0 & \lambda & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \cdots & \lambda
\end{bmatrix} 
\begin{bmatrix}
A_1^n \\
A_2^n \\
\vdots \\
A_{j-1}^n \\
A_j^n \\
A_{j+1}^n \\
\vdots \\
A_{J-1}^n \\
A_J^n
\end{bmatrix}
\]

(56)

or

\[
\begin{bmatrix}
A_j^{n+1}
\end{bmatrix} = \lambda I A_j^n,
\]

(57)

where \( I \) is the identity matrix. Comparing (54) and (57), we see that

\[
([M] - \lambda I) A_j^n = 0.
\]

(58)

This equation must hold regardless of the values of the \( A_j^n \). It follows that the amplification factors, \( \lambda \), are the eigenvalues of \([A]\), obtained by solving

\[
\begin{bmatrix}
[M] - \lambda I
\end{bmatrix} = 0,
\]

(59)

where the absolute value signs denote the determinant. For the current example, we can show that (59) leads to
\[
\lambda = 1 - \mu \left(1 - e^{\frac{2\pi i}{J}}\right), \quad m = 0, 1, 2, \ldots, J - 1.
\]  
(60)

This has essentially the same form as (43), and so it turns out that \(0 \leq \mu \leq 1\) is the stability condition again.

3.7 What happens if we increase the number of grid points and cut the time step?

Consider what happens when we increase the number of grid points, while fixing the domain size, \(D\), the wind speed, \(c\), and the wave number \(k\) of the advected signal. We consider grid spacing \(\Delta x\), such that

\[
D = J \Delta x.
\]  
(61)

As we decrease \(\Delta x\), we increase \(J\) correspondingly, so that \(D\) does not change, and

\[
k \Delta x = \frac{kD}{J}.
\]  
(62)

Substituting this into (44), we obtain

\[
|\lambda|^2 = 1 + 2\mu(\mu - 1) \left[1 - \cos \left(\frac{kD}{J}\right)\right].
\]  
(63)

In order to maintain computational stability, we keep \(\mu\) fixed as \(\Delta x\) decreases, so that

\[
\Delta t = \frac{\mu \Delta x}{c} = \frac{\mu D}{cJ}.
\]  
(64)

The time required for the air to flow through the domain is

\[
T = \frac{D}{c}.
\]  
(65)

Let \(N\) be the number of time steps needed for the air to flow through the domain, so that
\[ N = \frac{T}{\Delta t} \]
\[ = \frac{D}{c\Delta t} \]
\[ = \frac{D}{\mu \Delta x} \]
\[ = \frac{J}{\mu}. \]

(66)

The total amount of damping that “accumulates” as the air moves across the domain is given by

\[ |\lambda|^N = \left(|\lambda|^2\right)^{N/2} = \left\{1 - 2\mu(1 - \mu)\left[1 - \cos\left(\frac{kD}{J}\right)\right]\right\}^{\frac{J}{\mu}}. \]

(67)

Here we have used (63) and (66).

Figure 3.6: "Total" damping experienced by a disturbance crossing the domain, as a function of \( J \), the number of grid points across the domain, for two different fixed values of \( \mu \). In these examples we have assumed \( D/L = 2 \), i.e., the wavelength is half the width of the domain.

In Fig 3.6, we show the dependence of \( |\lambda|^N \) on \( J \), for two different fixed values of \( \mu \). The wavelength is assumed to be half the domain width, so that \( kD = 4\pi \). As we increase the resolution with a fixed domain size, \( J \) increases, for both values of \( \mu \). This causes the cosine factor in (68) to approach 1, which weakens the damping associated with \( |\lambda| < 1 \); but on the other
hand it also causes the exponent in (68) to increase, which strengthens the damping. Which effect dominates? The answer can be seen in Fig. 3.6. Increasing $J$ leads to less total damping for a given value of $\mu$, even though the number of time steps needed to cross the domain increases. This is good news.

On the other hand, if we fix $J$ and decrease $\mu$ (by decreasing the time step), the damping increases, so the solution becomes less accurate. This means that, for the upstream scheme, we can minimize the amplitude error by using the largest stable value of $\mu$. We minimize the error by living dangerously.

3.8 Summary

This chapter gives a quick introduction to the solution of a finite-difference equation, using the upstream scheme for the advection equation as an example. We have encountered the concepts of convergence and stability, and three different ways to test the stability of a scheme.

Suppose that we are given a non-linear partial differential equation and wish to solve it by means of a finite difference approximation. The usual procedure would be as follows:

- **Check the truncation error.** Normally this is done by means of a Taylor series expansion. We are concerned with the lowest powers of the space and time grid-interval in the expansion of the independent variables.

- **Check linear stability** for a simplified (linearized, constant coefficients) version of the equation. Von Neumann’s method is often used here.

- **Check nonlinear stability**, if possible. This can be accomplished, in some cases, by using the energy method. Otherwise, empirical tests are needed.

Increased accuracy as measured by truncation error does not always imply a better scheme. For example, consider two schemes A and B, such that scheme A is first-order accurate but stable, while scheme B is second-order accurate but unstable. Given such a choice, the “less accurate” scheme is definitely better.

Almost always, the design of a finite-difference scheme is an exercise in trade-offs. For example, a more accurate scheme is usually more complicated and expensive than a less accurate scheme. We have to ask whether the additional complexity and computational expense are justified by the increased accuracy. The answer depends on the particular application.

In general, “good” schemes have the following properties, among others:

- High accuracy.
- Stability.
- Simplicity.
- Computational economy.
Later, we will extend this list.

**Problems**

1. Program the upstream scheme on a periodic domain with 100 grid points. Give a sinusoidal initial condition with a single mode such that exactly four wavelengths fit in the domain. Integrate for $\mu = -0.1, 0.1, 0.5, 0.9, 1$ and $1.1$. In each case, take enough time steps so that in the exact solution the signal will just cross the domain. Plot and discuss your results.

2. *Using the energy method*, determine the stability of the forward time-differencing scheme as applied to the following pair of equations, which describe inertial oscillations:

   \[
   \frac{du}{dt} = f v, \tag{68}
   \]

   \[
   \frac{dv}{dt} = -fu. \tag{69}
   \]