

CHAPTER 3***A Survey of Time-Differencing Schemes
for the Oscillation and Decay Equations***

Copyright 2003 David A. Randall

3.1 Introduction

In atmospheric dynamics, the governing equations are usually non-linear partial differential equations. Some knowledge of finite-difference approximations to ordinary differential equations (especially first order) is needed, however. In fact, if we linearize a governing partial differential equation and assume a wave form for the solution, the equation simply reduces to an ordinary differential equation. An example of this was given in Chapter 2. The stability of a finite difference approximation to such an ordinary differential equation can be examined using von Neumann's method, as explained in Chapter 2. In this Chapter, we deliberately side-step the complexities of space differencing and consider the problem of time differencing in isolation.

Consider an arbitrary first-order ordinary differential equation of the form:

$$\frac{dq}{dt} = f[q(t), t]. \quad (3.1)$$

Both q and $f(q, t)$ may be complex variables. In the following two subsections, we do not specify $f(q, t)$. Later we will consider particular cases. Keep in mind that $f(q, t)$ could be very complicated.

3.2 Non-iterative schemes.

Suppose that we integrate (3.1) with respect to time, from $(n - m)\Delta t$ to $(n + 1)\Delta t$. Here we assume that m is either zero or a positive integer. We also assume that $n \geq m$, which may not be true close to the initial condition; this point is considered later. Then we have

$$q[(n + 1)\Delta t] - q[(n - m)\Delta t] = \int_{(n - m)\Delta t}^{(n + 1)\Delta t} f(q, t) dt. \quad (3.2)$$

Equation (3.2) is still “exact;” no finite-difference approximations have been introduced. With a finite difference-scheme, q and, therefore, f are defined only at discrete time levels. Suppose that we approximate the integral on the right-hand side of (3.2) using the values of f at the discrete time levels. We use symbol q^{n+1} in place of $q[(n + 1)\Delta t]$, f^{n+1} in place of

$f\{q[(n+1)\Delta t], (n+1)\Delta t\}$, etc. Equation (3.2), divided by $(1+m)\Delta t$, can be approximated by

$$\frac{q^{n+1} - q^{n-m}}{(1+m)\Delta t} \cong \beta f^{n+1} + \alpha_n f^n + \alpha_{n-1} f^{n-1} + \alpha_{n-2} f^{n-2} + \dots + \alpha_{n-l} f^{n-l}, \quad (3.3)$$

where l can be minus one (for $\beta \neq 0$ only), zero, or a positive integer. Take a minute to look carefully at the form of (3.3), which is a slightly modified version of an equation discussed by Baer and Simons (1970). The left-hand side is a “time step” across a time interval of $(m+1)\Delta t$, as illustrated in Fig. 3.1. The right-hand side consists of a weighted sum of instances of the function f , evaluated at various time levels. The first time level, $n+1$, is in “the future.” The second, n , is in “the present.” The remaining time levels are in “the past.” Time level $n-l$ is furthest back in the past; this is essentially the definition of l . We get to choose l and m when we design a scheme. A *family of schemes* is defined by (3.3). It is possible to have $l > m$ or $l < m$ or $l = m$. Viable schemes can be constructed with all three possibilities, and examples will be given below.

If $\beta \neq 0$ the scheme is called “implicit,” and if $\beta = 0$ it is called “explicit.” Implicit schemes have nice properties, as discussed later, but they can be complicated because the “unknown” or “future” value of q , namely q^{n+1} , appears on the right-hand-side of the equation, as the argument of f^{n+1} . Examples will be shown later.

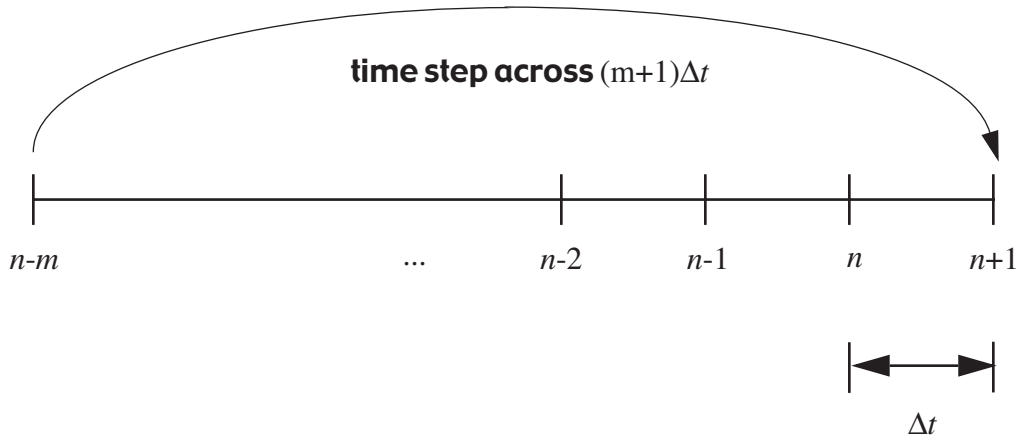


Figure 3.1: In Eq. (3.3), we use a weighted combination of $f^{n+1}, f^n, f^{n-1}, \dots, f^{n-l}$ to compute an “average” value of $f \equiv \frac{dq}{dt}$ over the time interval $(m+1)\Delta t$.

The smallest possible value of l is -1, in which case only time level $n+1$ appears on

the right-hand side of (3.3); this is the case with the backward-implicit scheme, for which $\beta \neq 0$. The Euler forward scheme uses time level n , so that $l = 0$. Note, however, that the Euler forward scheme omits time level $n + 1$; it is an explicit scheme with $\beta = 0$. There are many (in principle, infinitely many) other possibilities, as will be discussed later in this Chapter.

Now substitute the *true solution* $q(t)$ and corresponding $f[q(t), t]$ into (3.3), and expand into a Taylor series around $t = n\Delta t$. We get

$$\begin{aligned}
 & \frac{1}{(1+m)\Delta t} \left\{ \left(q + \Delta t q' + \frac{\Delta t^2}{2!} q'' + \frac{\Delta t^3}{3!} q''' + \frac{\Delta t^4}{4!} q'''' + \dots \right) \right. \\
 & \quad \left. - \left[q - (m\Delta t)q' + \frac{(m\Delta t)^2}{2!} q'' - \frac{(m\Delta t)^3}{3!} q''' + \frac{(m\Delta t)^4}{4!} q'''' - \dots \right] \right\} \\
 & = \beta \left[f + \Delta t f' + \frac{\Delta t^2}{2!} f'' + \frac{\Delta t^3}{3!} f''' + \dots \right] \\
 & \quad + \alpha_n f \\
 & \quad + \alpha_{n-1} \left[f - \Delta t f' + \frac{\Delta t^2}{2!} f'' - \frac{\Delta t^3}{3!} f''' + \dots \right] \\
 & \quad + \alpha_{n-2} \left[f - 2\Delta t f' + \frac{(2\Delta t)^2}{2!} f'' - \frac{(2\Delta t)^3}{3!} f''' + \dots \right] \\
 & \quad + \alpha_{n-3} \left[f - 3\Delta t f' + \frac{(3\Delta t)^2}{2!} f'' - \frac{(3\Delta t)^3}{3!} f''' + \dots \right] \\
 & \quad + \dots \\
 & \quad + \alpha_{n-l} \left[f - l\Delta t f' + \frac{(l\Delta t)^2}{2!} f'' - \frac{(l\Delta t)^3}{3!} f''' + \dots \right] \\
 & \quad + \varepsilon, \tag{3.4}
 \end{aligned}$$

where ε is the truncation error and a prime denotes a time derivative. Collecting powers of Δt , and using $q' = f$, $q'' = f'$, etc., we obtain

$$\begin{aligned}
& q'[1-(\beta + \alpha_n + \alpha_{n-1} + \alpha_{n-2} + \alpha_{n-3} + \dots + \alpha_{n-l})] \\
& + \Delta t q'' \left\{ \frac{1}{2} \frac{1-m^2}{1+m} - \beta + \alpha_{n-1} + 2\alpha_{n-2} + 3\alpha_{n-3} + \dots + l\alpha_{n-l} \right\} \\
& + \frac{(\Delta t)^2}{2!} q''' \left\{ \frac{1}{3} \frac{1+m^3}{1+m} - \beta - \alpha_{n-1} - 4\alpha_{n-2} - 9\alpha_{n-3} - \dots - l^2\alpha_{n-l} \right\} \\
& + \frac{(\Delta t)^3}{3!} q'''' \left\{ \frac{1}{4} \frac{1-m^4}{1+m} - \beta + \alpha_{n-1} + 8\alpha_{n-2} + 27\alpha_{n-3} + \dots + l^3\alpha_{n-l} \right\} \\
& + \dots \\
& = \varepsilon .
\end{aligned} \tag{3.5}$$

Each line on the left-hand side of (3.5) goes to zero “automatically” as $\Delta t \rightarrow 0$, except for the first line, which does not involve Δt at all. We have to force the first line to be zero, because otherwise the error, ε , will not go to zero as $\Delta t \rightarrow 0$. In order to force the first line to be zero, we have to choose β and the various α ’s in such a way that

$$1 = \beta + \alpha_n + \alpha_{n-1} + \alpha_{n-2} + \alpha_{n-3} + \dots + \alpha_{n-l} . \tag{3.6}$$

Note that (3.6) simply means that the sum of the coefficients on the right-hand side of (3.3) is equal to one, so that the right-hand side is a kind of “average f .” We refer to (3.6) as the “*consistency condition*.” When (3.6) is satisfied, the expression for the truncation error reduces to

$$\varepsilon = \Delta t q'' \left\{ \frac{1}{2} \left(\frac{1-m^2}{1+m} \right) - \beta + \alpha_{n-1} + 2\alpha_{n-2} + 3\alpha_{n-3} + \dots + l\alpha_{n-l} \right\} + O[(\Delta t)^2] . \tag{3.7}$$

This shows that when (3.6) is satisfied the scheme has at least first order accuracy, i.e. the scheme is consistent, and the error, ε , goes to zero at least as fast as Δt . *Note, however, that we are still free to choose $l+1$ coefficients.* Moreover, the value of l itself is under our control; we can choose it when we design the scheme. If $l \geq 0$, then we can choose the coefficients in such a way that the coefficient of Δt in (3.5) is also zero. This will give us a second-order scheme, i.e. one in which the error, ε , goes to zero like $(\Delta t)^2$. Obviously this process can be continued, giving higher and higher accuracy, so long as the value of l is large enough. Examples are given below.

In summary, the order of accuracy of our time-differencing scheme can be made at least as high as $l+2$ by appropriate choices of the coefficients. One of these coefficients is β . Recall that $\beta = 0$ for explicit schemes. Generally, then, the accuracy of an explicit

scheme can be made at least as high as $l + 1$. Later we refer back to these rules of thumb.

With the approach outlined above, schemes of higher order accuracy are made possible by bringing in more time levels. It is also possible to obtain schemes of higher accuracy in other ways. This will be explained later.

We now survey a number of time-differencing schemes, without specifying f . In this analysis, we can determine the order of accuracy of each scheme. We cannot decide whether a scheme is stable or unstable, however, unless f is specified. Later in this Chapter we investigate what happens with two particular choices of f , and later in the course we will consider additional choices for f .

3.2.1 Explicit schemes ($\beta = 0$)

$m = 0, l = 0$ (Forward scheme or Euler scheme)

For this case we have $\alpha_n \neq 0$, and all of the other α 's are zero. The consistency condition, (3.6), immediately forces us to choose $\alpha_n = 1$. The scheme (3.3) then reduces to

$$\frac{q^{n+1} - q^n}{\Delta t} = f^n \quad (3.8)$$

The truncation error is $\frac{\Delta t}{2} q'' + O(\Delta t^2) = O(\Delta t)$. Therefore, the scheme has first-order accuracy.

$m = 0, l > 0$ (Adams-Bashforth schemes)

Better accuracy can be obtained by proper choice of the α 's, if we use $l > 0$. For example, consider the case $l = 1$. The scheme reduces to

$$\frac{q^{n+1} - q^n}{\Delta t} = \alpha_n f^n + \alpha_{n-1} f^{n-1}, \quad (3.9)$$

the consistency condition, (3.6), reduces to

$$\alpha_n + \alpha_{n-1} = 1, \quad (3.10)$$

and the truncation error is

$$\varepsilon = \Delta t q'' \left(\alpha_{n-1} + \frac{1}{2} \right) + O(\Delta t^2). \quad (3.11)$$

If we choose $\alpha_{n-1} = -\frac{1}{2}$, the scheme has second-order accuracy. Of course, this means that $\alpha_n = \frac{3}{2}$. This is the second-order Adams-Bashforth scheme.

Although the right-hand side of (3.9) involves two different values of f , we only have to evaluate f once per time step, if we simply save one “old” time level of f for later use on the next time step. We have to allocate additional memory in the computer to save the “old” time level of f , but often this is not a problem. Note, however, that something special will have to be done on the first time step only, since when $n = 0$ time level $n - 1$ is “before the beginning” of the computation.

In a similar way, we can obtain Adams - Bashforth schemes with higher accuracy by using larger l , and choosing the α 's accordingly. The table below shows the results for $l = 1, 2$, and 3 . See the paper by Durran (1991) for an interesting discussion of the third-order Adams-Bashforth scheme. We can think of the forward scheme as the “first-order Adams-Bashforth scheme”, with $l = 0$.

Table 3.1: Adams-Bashforth Schemes ($\beta = m = 0, l > 0$)

l	α_n	α_{n-1}	α_{n-2}	α_{n-3}	truncation error
1	3/2	-1/2			$O(\Delta t^2)$
2	23/12	-4/3	5/12		$O(\Delta t^3)$
3	55/24	-59/24	37/24	-9/24	$O(\Delta t^4)$

$m = 1, l = 0$ (The leapfrog scheme)

The leap-frog scheme is

$$\frac{1}{2\Delta t}(q^{n+1} - q^{n-1}) = f^n \quad (3.12)$$

From (3.5) we can immediately see that the truncation error is $\frac{\Delta t^2}{6}q''' + O(\Delta t^4)$.

Note that for the leap-frog scheme the order of accuracy is higher than $l + 1 = 1$, i.e., it is better than would be expected from the general rule, stated earlier, for explicit schemes. The leapfrog scheme has been very widely used, but it has some serious disadvantages, as will be discussed later.

$$m = 1, l = 1$$

Here there is no gain in accuracy. The highest accuracy (second order) is obtained for $\alpha_{n-1} = 0$ (the leapfrog scheme).

$$m = 1, l > 1 \text{ (Nystrom schemes)}$$

We can increase the order of accuracy by choosing appropriate α 's if $l > 1$.

Schemes with $m > 1$ are not of much interest and will not be discussed here.

3.2.2 Implicit schemes ($\beta \neq 0$)

Here we should be able to achieve accuracy at least as high as $l + 2$. Note that with implicit schemes it is possible to have $l = -1$, whereas with the explicit schemes the smallest allowed value of l is 0.

$$m = 0, l = 0$$

Eq. (3.3) reduces to

$$\frac{q^{n+1} - q^n}{\Delta t} = \beta f^{n+1} + \alpha_n f^n. \quad (3.13)$$

The consistency condition reduces to $\alpha_n + \beta = 1$. The truncation error is $\Delta t q'' \left(\frac{1}{2} - \beta \right) + O(\Delta t^2)$. When $\beta = 1, \alpha_n = 0$, the scheme is called the backward (implicit) scheme. It has first-order accuracy. It can be said to correspond to $l = -1$. Higher accuracy is obtained for $\beta = \alpha = \frac{1}{2}$, which gives the “trapezoidal” (implicit) scheme. It has second-order accuracy, as we expect from the general rule for implicit schemes.

$$m = 0, l > 0 \text{ (Adams-Moulton schemes)}$$

These are analogous to the Adams-Bashforth schemes, except that $\beta \neq 0$. Table 3.2

summarizes the properties of these schemes, for $l = 1, 2$, and 3 .

Table 3.2: Adams–Moulton schemes.

l	β	α_n	α_{n-1}	α_{n-2}	α_{n-3}	truncation error
1	5/12	8/12	-1/12			$O(\Delta t^3)$
2	9/24	19/24	-5/24	1/24		$O(\Delta t^4)$
3	251/720	646/720	-264/720	106/720	-19/720	$O(\Delta t^5)$

$$m = 1, l = 0$$

Highest accuracy (2nd order) is obtained for $\beta = 0$, which gives the leapfrog scheme.

$$m = 1, l = 1 \text{ (Milne corrector}^1\text{)}$$

Eq. (3.3) reduces to

$$\frac{q^{n+1} - q^{n-1}}{2\Delta t} = \beta f^{n+1} + \alpha_n f^n + \alpha_{n-1} f^{n-1}, \quad (3.14)$$

where

$$\beta + \alpha_n + \alpha_{n-1} = 1. \quad (3.15)$$

The truncation error is

$$\varepsilon = \Delta t q''(-\beta + \alpha_{n-1}) + \frac{\Delta t^2}{2!} q''' \left(\frac{1}{3} - \beta - \alpha_{n-1} \right) + \frac{\Delta t^3}{3!} q''''(-\beta + \alpha_{n-1}) + O(\Delta t^4).$$

From this we can see that $\beta = \frac{1}{6}$, $\alpha_n = \frac{4}{6}$, $\alpha_{n-1} = \frac{1}{6}$ gives fourth-order accuracy. This is again more than would be expected from the general rule.

¹. If there is a “Milne corrector,” then there must be a “Milne predictor.” (See subsection 3.3 for an explanation of this terminology.) In fact, the Milne predictor is an explicit scheme with $m = 3$, $l = 3$, and $\alpha_n = \frac{2}{3}$, $\alpha_{n-1} = -\frac{1}{3}$, $\alpha_{n-2} = \frac{2}{3}$, $\alpha_{n-3} = 0$.

$$m = 1, l = 2$$

Here there is no gain in accuracy. The highest accuracy is obtained for $\alpha_{n-2} = 0$, so that the scheme reduces to the Milne corrector.

3.3 Iterative schemes

Iterative schemes are sometimes called “predictor-corrector” schemes. The idea is that we obtain q^{n+1} through an iterative, multi-step procedure, which involves multiple evaluations of the function f . In a two-step iterative scheme, the first step is called the “predictor,” and the second step is called the “corrector.”

The advantage of iterative schemes is that we can gain higher accuracy. The disadvantage is computational expense, because each evaluation of f involves doing a certain amount of arithmetic. In contrast, non-iterative schemes such as those discussed in the preceding subsection involve only a single evaluation of f for each time step. For this reason, iterative schemes tend to be computationally more expensive than non-iterative schemes, for a given order of accuracy.

Consider (3.13) as an example. Replace $f^{n+1} \equiv f[q^{n+1}, (n+1)\Delta t]$ by $f^{n+1*} \equiv f[q^{n+1*}, (n+1)\Delta t]$, where q^{n+1*} is obtained by the Euler scheme,

$$\frac{q^{n+1*} - q^n}{\Delta t} = f^n. \quad (3.16)$$

Then

$$\frac{q^{n+1} - q^n}{\Delta t} = \beta^* f^{n+1*} + \alpha f^n. \quad (3.17)$$

When $\beta^* = 1$, $\alpha = 0$, Eq. (3.17) is an imitation of the backward (implicit) difference scheme, and is called the Euler-backward scheme or the Matsuno scheme (Matsuno, 1966).

When $\beta^* = \frac{1}{2}$, $\alpha = \frac{1}{2}$, Eq. (3.17) is an imitation of the trapezoidal (implicit) scheme and is called the Heun scheme or the second-order Runge-Kutta scheme. The Matsuno scheme has first-order accuracy, and the Heun scheme has second-order accuracy.

Note that (3.17) cannot be “fit” into the framework of (3.3), because f^{n+1*} does not appear on the right-hand side of (3.3), and in general f^{n+1*} cannot be written as a combination of f^{n-l} ’s.

Also note that the Heun scheme is explicit, and does not require the past history (does

not require $l > 0$). Still, it has second order accuracy, because of the iteration. This illustrates that *iteration can increase the order of accuracy*.

A famous example of an iterative scheme is the fourth-order Runge-Kutta scheme. This is an excellent scheme when f has a simple form, but it is not economically practical when f is complicated. The scheme is given by:

$$\begin{aligned}
 q^{n+1} &= q^n + \Delta t \frac{1}{6} (k_1 + 2k_2 + 2k_3 + k_4), \\
 k_1 &= f(q^n, n\Delta t), & k_2 &= f\left[q^n + \frac{k_1\Delta t}{2}, \left(n + \frac{1}{2}\right)\Delta t\right], \\
 k_3 &= f\left[q^n + \frac{k_2\Delta t}{2}, \left(n + \frac{1}{2}\right)\Delta t\right], & k_4 &= f(q^n + k_3\Delta t, (n+1)\Delta t).
 \end{aligned} \tag{3.18}$$

Each of the k 's can be interpreted as an approximation to f . The k 's have to be evaluated successively, which means that the function f has to be evaluated four times to take one time step. None of these f 's can be "re-used" on the next time step. For this reason, the scheme is not very practical unless a long time step can be used.

Fig. 3.2 provides a simple fortran example to illustrate more clearly how the fourth-order Runge-Kutta scheme actually works. The appendix to this chapter provides a proof that the scheme really has fourth-order accuracy.

3.4 Finite-difference schemes applied to the oscillation equation

In order to test the stability of the finite-difference schemes discussed in the previous section, we must specify the form of the function $f(q, t)$. As a first example, consider the oscillation equation:

$$\frac{dq}{dt} = i\omega q, \quad q \text{ complex, } \omega \text{ real.} \tag{3.19}$$

The exact solution is $q = \hat{q}(t)e^{i\omega t}$, where \hat{q} is the "initial" value of q , at $t = 0$. The amplitude \hat{q} is an invariant of this system, i.e.,

$$\frac{d\hat{q}}{dt} = 0. \tag{3.20}$$

The following are examples of more familiar equations which are reducible to (3.19).

- **Advection:** The governing equation is $\frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} = 0$. If $u = \hat{u}(t)e^{ikx}$, then

```

c      Initial conditions for time-stepped variables X, Y, and Z.
c      The time step is dt, and dt2 is half of the time step.

      X=2.5
      Y=1.
      Z=0.

      do n=1,nsteps
c      Subroutine dot evaluates time derivatives of X, Y, and Z.

      call dot(X, Y, Z,Xdot1,Ydot1,Zdot1)

c      First provisional values of X, Y, and Z.

      X1 = X + dt2 * Xdot1
      Y1 = Y + dt2 * Ydot1
      Z1 = Z + dt2 * Zdot1

      call dot(X1,Y1,Z1,Xdot2,Ydot2,Zdot2)

c      Second provisional values of X, Y, and Z.

      X2 = X + dt2 * Xdot2
      Y2 = Y + dt2 * Ydot2
      Z2 = Z + dt2 * Zdot2

      call dot(X2,Y2,Z2,Xdot3,Ydot3,Zdot3)

c      Third provisional values of X, Y, and Z.

      X3 = X + dt * Xdot3
      Y3 = Y + dt * Ydot3
      Z3 = Z + dt * Zdot3

      call dot(X3,Y3,Z3,Xdot4,Ydot4,Zdot4)

c      "Final" values of X, Y, and Z for this time step.

      X = X + dt * (Xdot1 + 2.*Xdot2 + 2.*Xdot3 + Xdot4)/6.
      Y = Y + dt * (Ydot1 + 2.*Ydot2 + 2.*Ydot3 + Ydot4)/6.
      Z = Z + dt * (Zdot1 + 2.*Zdot2 + 2.*Zdot3 + Zdot4)/6.

      end do

```

Figure 3.2: A simple fortran example to illustrate how the fourth-order Runge-Kutta scheme works. Note the four calls to subroutine "dot." This makes the scheme expensive.

$\frac{d\hat{u}}{dt} = -ikc\hat{u} = i\omega\hat{u}$, with $\omega \equiv -kc$. An observer at a point watching a *single Fourier mode* advect by will see an oscillation.

- **Rotation:** Pure inertial motion is described by

$$\frac{du}{dt} - fv = 0, \quad (3.21)$$

$$\frac{dv}{dt} + fu = 0. \quad (3.22)$$

Multiplying (3.22) by i and adding it to (3.21), we obtain

$$\frac{d}{dt}(u + iv) + f(-v + iu) = 0. \quad (3.23)$$

With $q = u + iv$, we get

$$\frac{dq}{dt} + ifq = 0, \quad (3.24)$$

which is identical to (3.21) with $\omega = -f$. Note that, although u and v are real, q is complex, and $|q|^2 = u^2 + v^2$ is twice the kinetic energy per unit mass. We can obtain $\frac{d}{dt}|q|^2 = 0$, i.e., kinetic energy conservation, directly from (3.21) and (3.22). By differentiating (3.21) with respect to time, and substituting from (3.22) for $\frac{dv}{dt}$, we obtain $\frac{d^2u}{dt^2} = -f^2u$, perhaps a more familiar form of the oscillation equation, which in this case describes a pure inertial oscillation.

In principle, any of the schemes described earlier in this chapter can be applied to (3.19). Each scheme has its own properties, as discussed below.

3.4.1 Non-iterative two-level schemes for the oscillation equation

Write a finite difference analog of (3.19) as follows

$$q^{n+1} - q^n = i\omega\Delta t(\alpha q^n + \beta q^{n+1}). \quad (3.25)$$

We require $\alpha + \beta = 1$ in order to guarantee consistency. We obtain the Euler scheme with $\alpha = 1$, $\beta = 0$; the backward scheme with $\alpha = 0$, $\beta = 1$; and the trapezoidal-implicit scheme with $\alpha = \beta = \frac{1}{2}$. Eq. (3.25) can easily be solved for q^{n+1} :

$$(1 - i\Omega\beta)q^{n+1} = (1 + i\Omega\alpha)q^n, \quad (3.26)$$

or

$$q^{n+1} = \left(\frac{1 + i\Omega\alpha}{1 - i\Omega\beta} \right) q^n \equiv \lambda q^n, \quad (3.27)$$

where we introduce the shorthand notation $\Omega \equiv \omega\Delta t$. In (3.27), λ is the amplification factor.

We want to know how the amplitude $|q|$ behaves in time. Recall that $|q|$ is invariant for the differential equation. This means that, for the true solution, $|\lambda| = 1$. If the computed $|\lambda|$ is not equal to one, we have “amplitude errors.”

Since λ is complex, we write

$$\lambda = \lambda_r + i\lambda_i = |\lambda|e^{i\theta}, \quad \text{where} \quad \tan\theta = \frac{\lambda_i}{\lambda_r}, \lambda_r = |\lambda|\cos\theta, \lambda_i = |\lambda|\sin\theta. \quad (3.28)$$

We can interpret θ as the phase change of the oscillation per time step. Positive θ denotes counterclockwise rotation in the complex plane. For example, if $\theta = \frac{\pi}{2}$, it takes four time steps to complete one oscillation. This is the case in which λ is pure imaginary. The discrete numerical solution may look as shown schematically in Fig. 3.3 for the case of $\theta = \frac{\pi}{2}$; the

ordinate represents the imaginary part of q^n . Note that for the exact solution the phase change over the time interval Δt is $\Omega \equiv \omega\Delta t$. Generally θ (the computed phase change) and Ω (the true or exact phase change) will differ because of discretization errors. If the computed θ is not equal to Ω , we have “phase errors.”

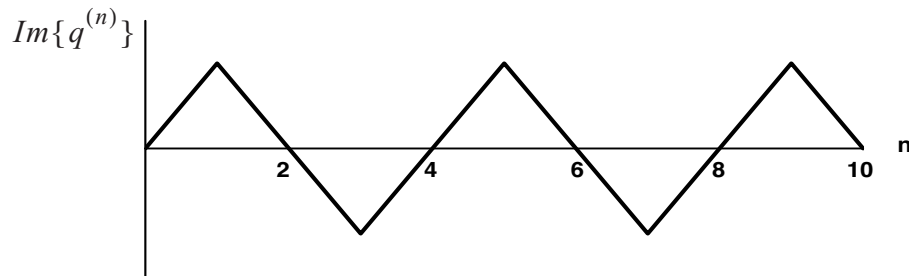


Figure 3.3: Schematic illustration of the solution of the oscillation equation for the case in which λ is pure imaginary and the phase changes by $\pi/2$ on each time step.

The true solution to (3.19) is $q \propto e^{i\omega t}$, so that

$$q[(n+1)\Delta t] = e^{i\omega\Delta t} q(n\Delta t) = e^{i\Omega} q(n\Delta t). \quad (3.29)$$

Now $|e^{i\Omega}| = 1$, and so $\lambda = |\lambda|e^{i\theta}$ corresponds to (“should be”) $e^{i\Omega}$; in other words, $|\lambda|$ should be equal to 1 and θ should be equal to Ω . We want to compare the computed amplification factor $|\lambda|$ with the true one, which is $|e^{i\Omega}| = 1$, and the computed phase change per time step, θ , with the true one, which is Ω . We will examine both the phase error and the amplitude error.

For the forward (Euler) scheme, $\alpha = 1$, $\beta = 0$, and so from (3.27) we find that

$$\lambda = 1 + i\Omega. \quad (3.30)$$

Therefore,

$$|\lambda| = \sqrt{1 + \Omega^2} > 1. \quad (3.31)$$

We conclude that, for the oscillation equation, the forward scheme is *unconditionally unstable*. We have reason to *suspect*, therefore, that forward time differencing is not a good choice for the advection or coriolis terms of a dynamical model. In reality, whether or not the forward scheme is a good choice for the advection terms depends on the space-differencing scheme used, as will be discussed in Chapter 4. From (3.30) we see that the phase change per time step, θ , satisfies $\tan\theta = \Omega$, so that $\theta \cong \Omega$ for small Δt , as expected.

For the backward scheme, $\alpha = 0$, $\beta = 1$, and

$$\lambda = \frac{1}{1 - i\Omega} = \frac{1 + i\Omega}{1 + \Omega^2}, \quad (3.32)$$

so that

$$|\lambda| = \frac{\sqrt{1 + \Omega^2}}{1 + \Omega^2} = \frac{1}{\sqrt{1 + \Omega^2}} < 1. \quad (3.33)$$

This scheme is, therefore, unconditionally stable, and in fact the amplitude of the oscillation decreases with time. The real part of λ is always positive, which means that $-\frac{\pi}{2} < \theta < \frac{\pi}{2}$. This scheme can be used for the coriolis terms in a model, but because of the damping it is not a very good choice. The phase change per time step again satisfies $\tan\theta = \Omega$, so again the phase error is small for small Δt .

For the trapezoidal implicit scheme, $\alpha = \frac{1}{2}$, $\beta = \frac{1}{2}$, we find that

$$|\lambda| = \frac{\left|1 + \frac{i\Omega}{2}\right|}{\left|1 - \frac{i\Omega}{2}\right|} = 1. \quad (3.34)$$

This scheme is *unconditionally stable*; in fact, it has no amplitude error at all. Its phase error per time step is small. It is a nice scheme.

3.4.2 Iterative two-level schemes for the oscillation equation

Now consider a finite-difference analogue of (3.19) given by

$$q^{n+1*} - q^n = i\Omega q^n, \quad (3.35)$$

$$q^{n+1} - q^n = i\Omega(\alpha q^n + \beta^* q^{n+1*}). \quad (3.36)$$

Recall that $\alpha = 0$, $\beta^* = 1$ gives the Matsuno scheme, and $\alpha = \beta^* = \frac{1}{2}$ gives the Heun scheme. Eliminating q^{n+1*} between (3.35) and (3.36) for the Matsuno scheme, we find that

$$\lambda = (1 - \Omega^2) + i\Omega,$$

$$|\lambda| = \sqrt{1 - \Omega^2 + \Omega^4} \begin{cases} > 1 \text{ for } \Omega > 1 \\ = 1 \text{ for } \Omega = 1 \\ < 1 \text{ for } \Omega < 1 \end{cases} \quad (3.37)$$

This is, therefore, a *conditionally stable* scheme (the condition is $\Omega \leq 1$).

For the Heun scheme, we obtain

$$\lambda = \left(1 - \frac{\Omega^2}{2}\right) + i\Omega$$

$$|\lambda| = \sqrt{\left(1 - \frac{\Omega^2}{2}\right)^2 + \Omega^2} = \sqrt{1 + \frac{\Omega^4}{2}} > 1. \quad (3.38)$$

This scheme is *unconditionally unstable*, but notice that for small Ω it is not as unstable as the forward scheme. In fact, it can be used with some success if very long-term integrations are not required.

The results discussed above are summarized in Fig. 3.4 and Fig. 3.5.

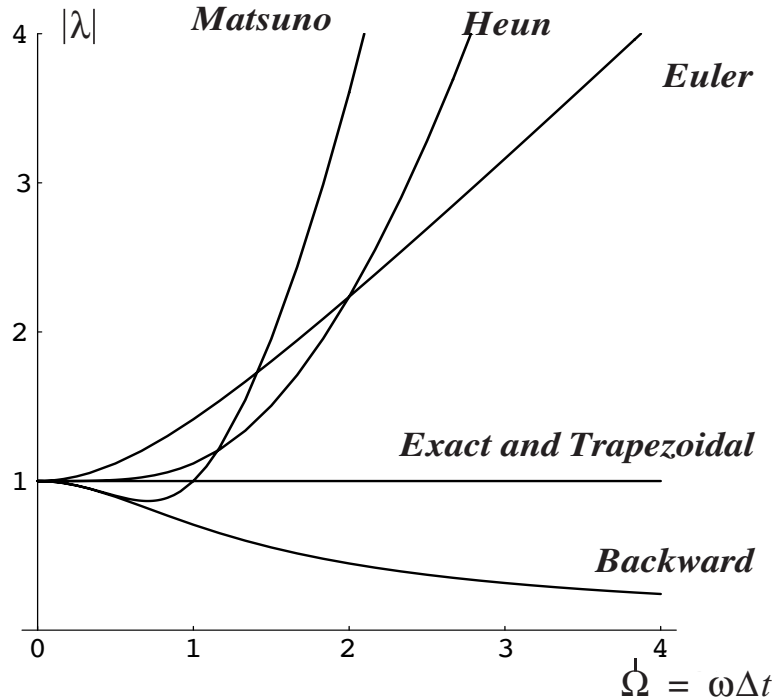


Figure 3.4: This figure shows the magnitude of the amplification factor as a function of $\Omega \equiv \omega\Delta t$ for various difference schemes. The Euler, backward, trapezoidal, Euler-backward, and Heun schemes are shown by curves I, II, III, IV, and V respectively. The magnitude of the amplification factor for the trapezoidal scheme coincides with that of the true solution for all values of $\omega\Delta t$. Caution: This does not mean that the trapezoidal scheme gives the exact solution!

3.4.3 The leapfrog scheme for the oscillation equation

The leapfrog scheme, which is illustrated in Fig. 3.6, is not “self-starting” at $n = 0$, because we do not know q at $n = 1$. A special procedure must, therefore, be used to start the solution, i.e. we must somehow determine the value of q at $n = 1$. We really need two initial conditions to solve the finite-difference problem, even though only one initial condition is needed to solve the exact equation. A similar problem arises with any scheme that involves more than two time levels. One of the two required initial conditions is the “physical” initial condition that we need for the differential equation. The other arises because of the form of the finite-difference scheme itself, and has nothing to do with the physics. It is usually referred to as the “computational” initial condition.

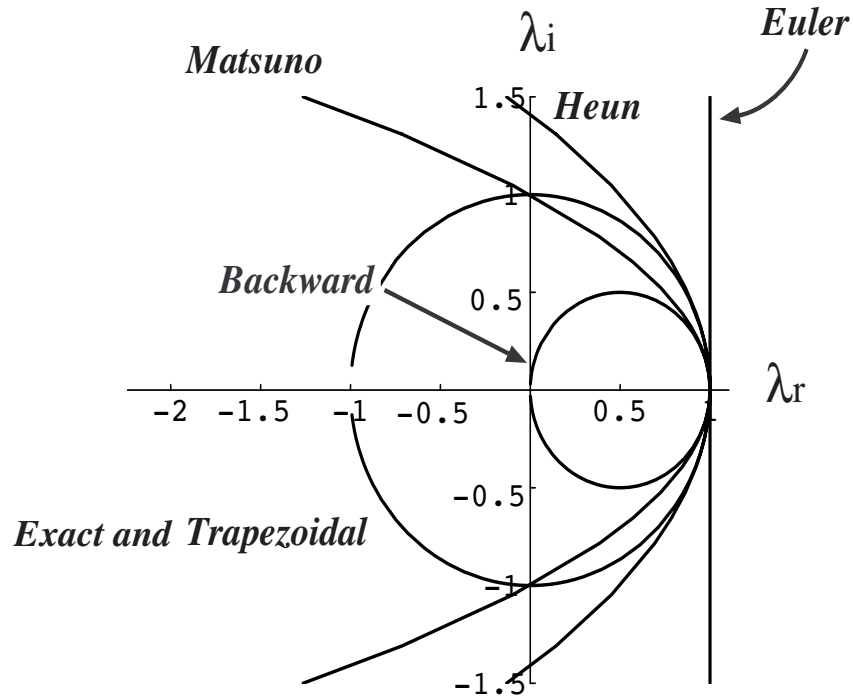


Figure 3.5: This figure shows the behavior of the imaginary (λ_i) and real (λ_r) components of the amplitude, as Ω varies. Recall that $\tan \theta$ is given by $\frac{\lambda_i}{\lambda_r}$ for each scheme.

From this plot we can also see the behavior of $|\lambda|$ as θ varies, for each scheme.

Consider the leapfrog analogue to (3.19):

$$q^{n+1} - q^{n-1} = 2\Delta t i \omega q^n. \quad (3.39)$$

For the simple case $\omega=0$, we obtain

$$q^{n+1} - q^{n-1} = 0. \quad (3.40)$$

Of course, $q = \text{constant}$ is the true solution of $\frac{dq}{dt} = 0$, but according to (3.39) the solution will depend on the initial conditions given at both the levels $n = 0$ and $n = 1$. If these two values are different, an oscillation will occur, as shown in Fig. 3.7. We have $q^2=q^0$, $q^3=q^1$, $q^4=q^2=q^0$, etc. If we assign $q^1 = q^0$, the solution will be constant. This example illustrates that judicious selection of $q^{(1)}$ is essential for schemes with more than two time levels.

Now rewrite (3.39) as

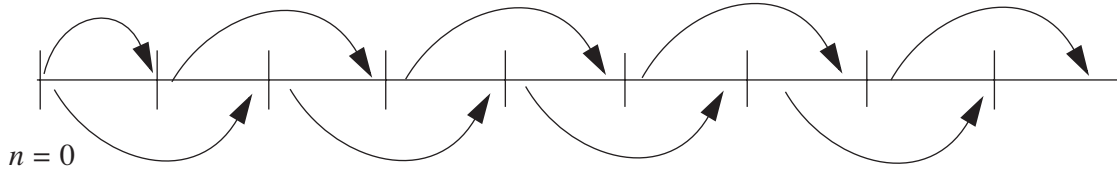


Figure 3.6: The leapfrog scheme.

$$q^{n+1} - 2i\Omega q^n - q^{n-1} = 0, \quad (3.41)$$

where, as before, $\Omega \equiv \omega\Delta t$. We look for a solution of the form $q^{n+1} = \lambda q^n$, for all n . Then (3.41) reduces to

$$\lambda^2 - 2i\Omega\lambda - 1 = 0. \quad (3.42)$$

The solutions of (3.42) are

$$\lambda_1 = i\Omega + \sqrt{1 - \Omega^2}, \quad \lambda_2 = i\Omega - \sqrt{1 - \Omega^2}, \quad (3.43)$$

giving two solutions or “modes”,

$$q_1^{n+1} = \lambda_1 q_1^n, \quad q_2^{n+1} = \lambda_2 q_2^n. \quad (3.44)$$

The differential equation only has one solution, so getting two solutions to the finite-difference equation is bad. Consider the limits of λ_1 and λ_2 as $\Omega \rightarrow 0$ or $\Delta t \rightarrow 0$. Notice that $\lambda_1 \rightarrow 1$, while $\lambda_2 \rightarrow -1$. We know that for the true solution $\lambda = 1$, and so we can identify q_1 as the “physical” mode, and q_2 as the computational mode. Notice that q_2^{n+1} generally does not approach q_2^n as $\Delta t \rightarrow 0$! This illustrates that simply reducing the time step does not reduce problems associated with computational modes.

The computational mode arises from the three-level scheme. Two-level schemes do not have computational modes. Schemes with more than three time levels have multiple computational modes. *The existence of computational modes is a major disadvantage of all schemes that involve more than two time levels.* The current discussion is about computational modes in time. Later we will see that computational modes can also occur in space, in at least two distinct ways.

From (3.44) we have

$$q_1^n = \lambda_1^n q_1^0, \quad q_2^n = \lambda_2^n q_2^0. \quad (3.45)$$

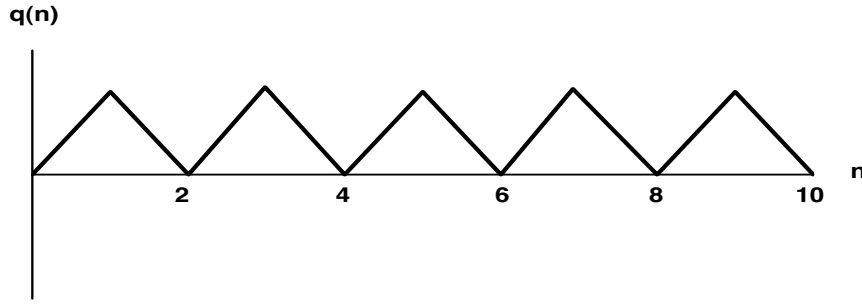


Figure 3.7: An oscillatory solution that arises with the leapfrog scheme for $dq/dt = 0$, for the case in which the two initial values of q are not the same.

The general solution is a linear combination of these two modes

$$q^n = a\lambda_1^n q_1^0 + b\lambda_2^n q_2^0. \quad (3.46)$$

Applying initial conditions, we have

$$(n = 0) \quad aq_1^0 + bq_2^0 = q^0, \quad (3.47)$$

and

$$(n = 1) \quad \lambda_1 a q_1^0 + \lambda_2 b q_2^0 = q^1. \quad (3.48)$$

If we solve (3.47) and (3.48) for aq_1^0 and bq_2^0 in terms of q^0 and q^1 , and substitute the results into (3.46), we obtain

$$q^n = \left(\frac{1}{\lambda_1 - \lambda_2} \right) [(q^1 - \lambda_2 q^0) \lambda_1^n - (q^1 - \lambda_1 q^0) \lambda_2^n]. \quad (3.49)$$

This shows that $\left[\frac{q^1 - \lambda_2 q^0}{\lambda_1 - \lambda_2} \right]$ and $-\left[\frac{q^1 - \lambda_1 q^0}{\lambda_1 - \lambda_2} \right]$ are the initial values of the physical and computational modes, respectively. Which predominates in the numerical solution will, therefore, depend on how we specify q^0 . If we give q^1 such that $q^1 = \lambda_1 q^0$, we will have the physical mode only. In real cases of interest, this is usually impossible to arrange. Notice that (3.49) applies for any choice of $f(q, t)$; it is not specific to the oscillation equation.

A simple way to give q^1 is by the forward (Euler) difference scheme. A more

sophisticated procedure is use of the forward scheme to $n = \frac{1}{2}$, followed by computation of q^1 using the leapfrog scheme. The second method gives a smaller amplitude for the computational mode.

To evaluate the stability of the leapfrog scheme as applied to the oscillation equation, consider three cases.

Case (i). $|\Omega| < 1$

In this case $\sqrt{1 - \Omega^2}$ in (3.43) is real, and we obtain $|\lambda_1| = |\lambda_2| = 1$. This means that both modes -- the physical and the computational -- are stable and neutral. Let the phase changes per time step be denoted by θ_1 and θ_2 for the physical and computational modes, respectively. Then

$$\lambda_1 = e^{i\theta_1}, \quad \lambda_2 = e^{i\theta_2}. \quad (3.50)$$

Comparing (3.50) with (3.43), we find that

$$\begin{aligned} \cos \theta_1 &= \sqrt{1 - \Omega^2}, & \cos \theta_2 &= -\sqrt{1 - \Omega^2}, \\ \sin \theta_1 &= \Omega, & \sin \theta_2 &= \Omega. \end{aligned} \quad (3.51)$$

Note that we can put $\theta_2 = \pi - \theta_1$. For simplicity of notation, let $\theta \equiv \theta_1$, so that $\theta_2 = \pi - \theta$. When Ω is small, $\theta_1 \cong \Omega$, and $\theta_2 \cong \pi$. The apparent frequency of the physical mode is $\frac{\theta_1}{\Delta t}$, which is approximately equal to ω . Then we can write

$$q_1^{n+1} = e^{i\theta} q_1^n \quad (3.52)$$

for the physical mode, and

$$q_2^{n+1} = e^{i(\pi - \theta)} q_2^n \quad (3.53)$$

for the computational mode. Recall that the true solution is given by

$$q[(n+1)\Delta t] = e^{i\Omega} q(n\Delta t). \quad (3.54)$$

In the limit as $\Delta t \rightarrow 0$ (i.e. $\Omega \rightarrow 0$), we have $\theta \rightarrow \Omega$. Panels a and b of Fig. 3.8 respectively show λ_1 and λ_2 in the complex λ -plane. The figures have been drawn for the case of

$\theta = \frac{\pi}{8}$. The absolute value of λ is, of course, always equal to 1. Panel c of Fig. 3.8 shows the

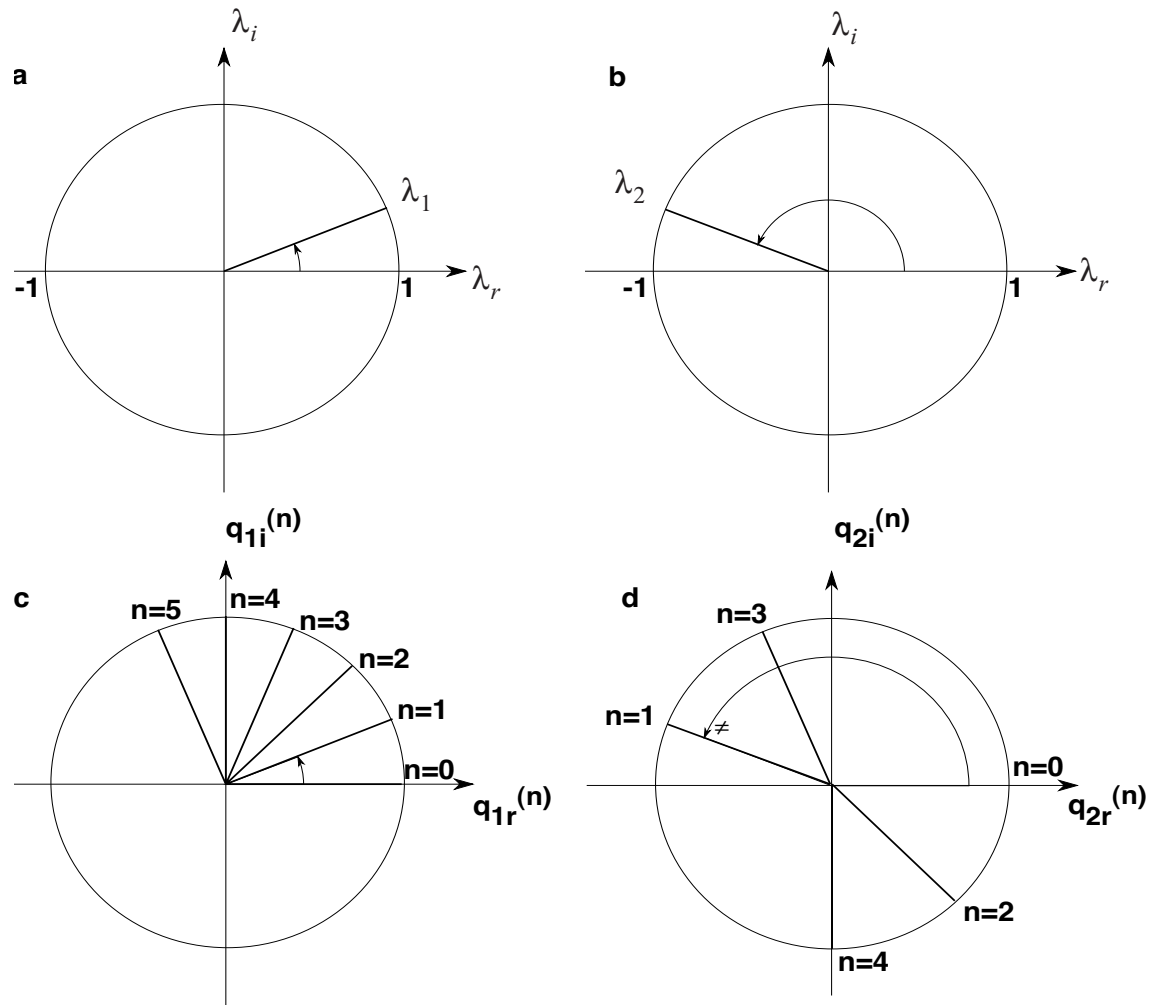


Figure 3.8: Panels a and b: Amplification factors for the leapfrog scheme as applied to the oscillation equation with $|\Omega| < 1$. Panels c and d: Solutions of the oscillation equation as obtained with the leapfrog scheme for $|\Omega| < 1$. In making these figures it has been assumed that $\theta = \frac{\pi}{8}$.

graph of the real part of q_1^n versus its imaginary part. Recall that $q_1^n = \lambda_1^n q_1^0 = e^{in\theta} q_1^0$. The

graph is drawn for the case of $q_{1i}^0 = 0$ and $\theta = \frac{\pi}{8}$. Panel d of Fig. 3.8 gives a similar plot

for q_2^n . Here we see that the real and imaginary parts of the computational mode of q^n both oscillate from one time step to the next. Graphs showing each part versus n are given in Fig. 3.9. The physical mode looks nice. The computational mode is ugly.

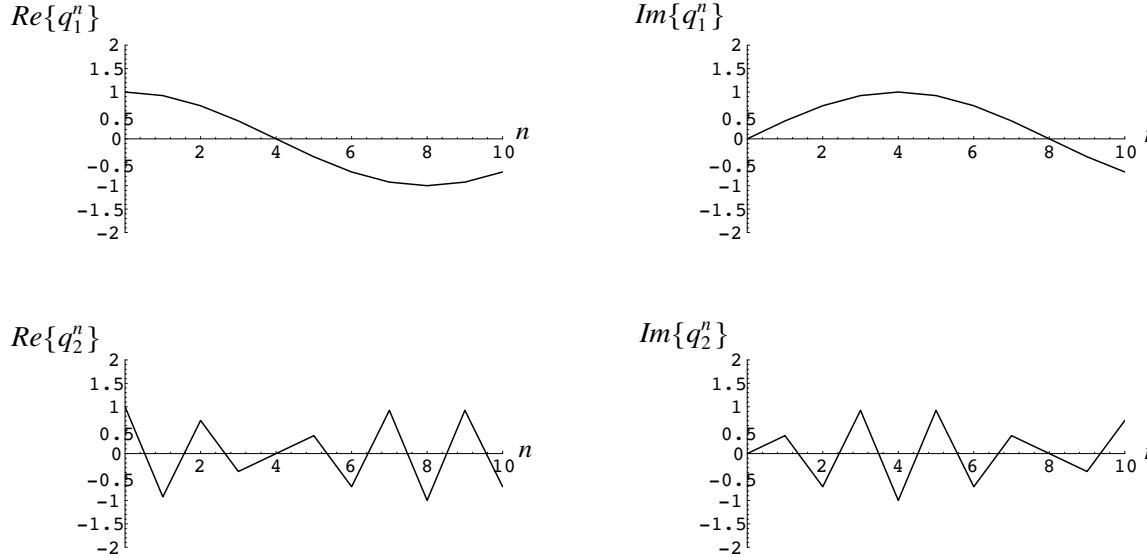


Figure 3.9: Graphs of the real and imaginary parts of the physical and computational modes of the solution of the oscillation equation as obtained with the leapfrog scheme for $|\Omega| < 1$.

Case (ii): $\Omega = \pm 1$

Here $\lambda_1 = \lambda_2 = i\Omega$ [see (3.43)], i.e. both λ 's are pure imaginary, as shown in Fig. 3.10. This means that, as shown in Fig. 3.10, both solutions rotate through $\frac{\pi}{2}$ on each time step, so that the period is $4\Delta t$, regardless of the true value of ω ! Obviously, $|\lambda_1| = |\lambda_2| = 1$, so both modes are neutral. The phase errors are very large, however. This illustrates a simple fact that should be remembered: A scheme that is stable but on the verge of instability is usually subject to large truncation errors and may give a very poor solution; you should not be confident that you have a good solution just because your model does not blow up!

Case (iii): $|\Omega| > 1$

Here again both λ_1 and λ_2 are pure imaginary, so again both solutions rotate by $\frac{\pi}{2}$ on each time step. We find that

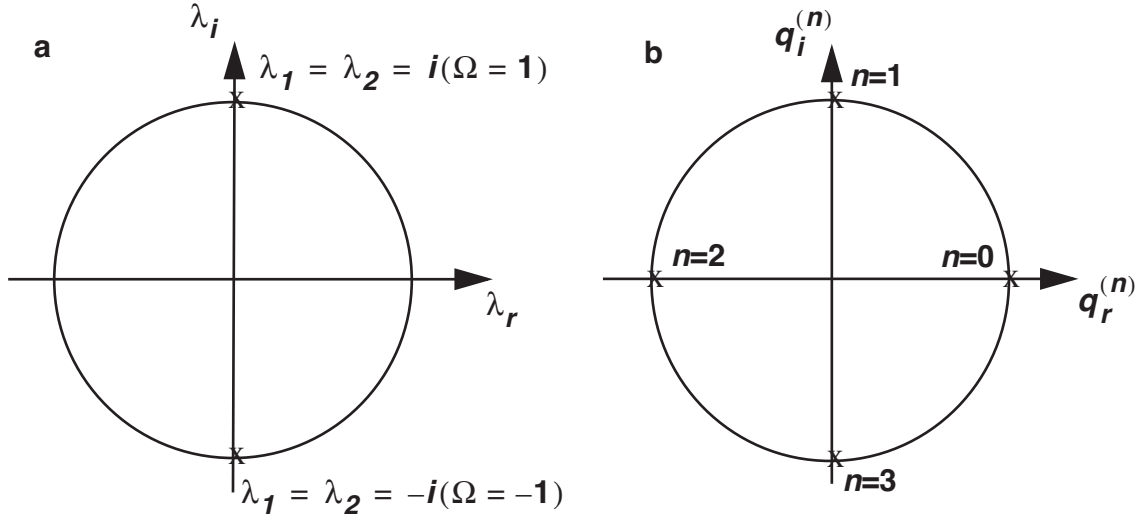


Figure 3.10: Panel a shows the amplification factors for the leapfrog scheme as applied to the oscillation equation with $|\Omega| = 1$. Panel b shows the real and imaginary parts of the corresponding solution, for $n=0, 1, 2$, and 3 .

$$\lambda_1 = i(\Omega + \sqrt{\Omega^2 - 1}) \quad \text{and} \quad \lambda_2 = i(\Omega - \sqrt{\Omega^2 - 1}). \quad (3.55)$$

If $\Omega > 1$, then $|\lambda_1| > 1$ and $|\lambda_2| < 1$, and if $\Omega < -1$, $|\lambda_1| < 1$ and $|\lambda_2| > 1$. In both cases, one of the modes is damped and the other amplifies. Since one of the modes amplifies for $|\Omega| > 1$, the scheme is unstable in this range of Ω .

A graphical representation of λ in the complex plane, for $|\Omega| > 1$, is shown in panels a and b of Fig. 3.11. Note that $\lambda_1 = \left| \Omega + \sqrt{\Omega^2 - 1} \right| e^{i\frac{\pi}{2}}$ and $\lambda_2 = \left| \Omega - \sqrt{\Omega^2 - 1} \right| e^{i\frac{\pi}{2}}$.

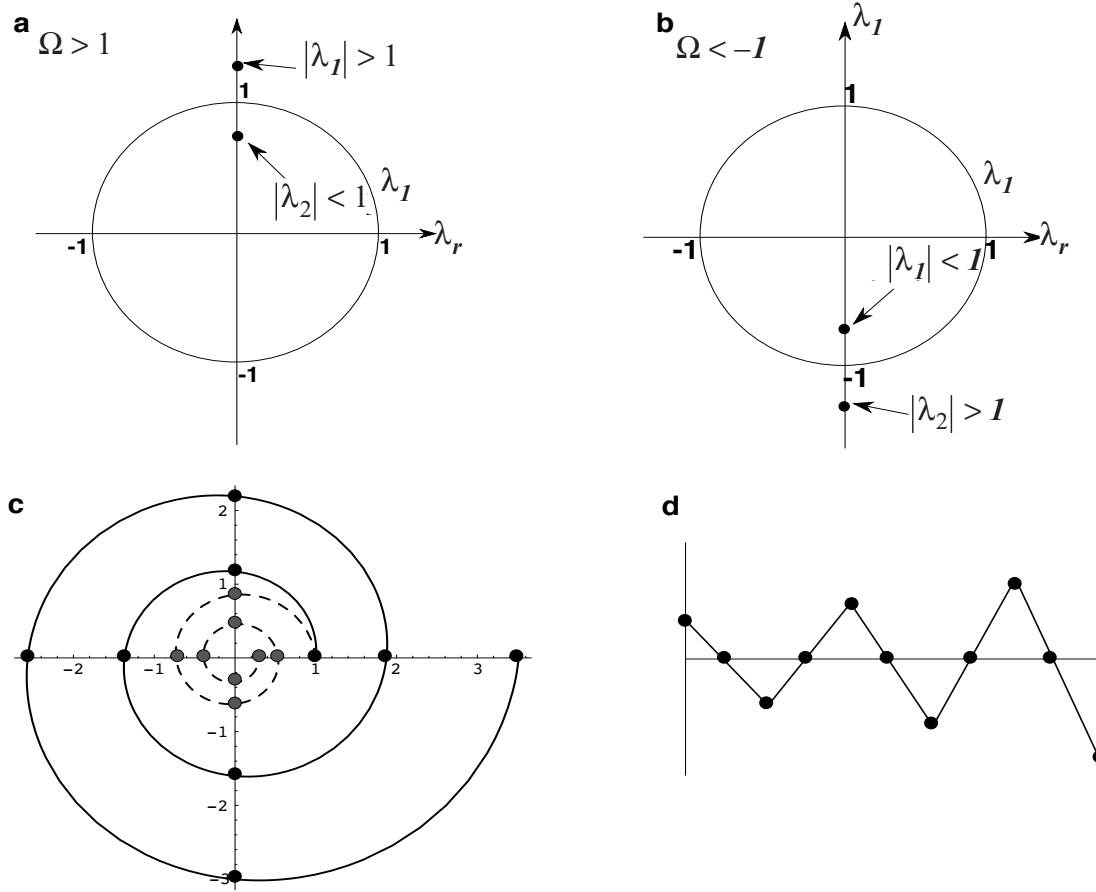


Figure 3.11: Panels a and b show the amplification factors for the oscillation equation with the leapfrog scheme, with $|\Omega| > 1$. Panel c shows the corresponding solution. The solid curve shows the unstable mode, which is actually defined only at the black dots. The dashed curve shows the damped mode, which is actually defined only at the grey dots. Panel d is a schematic illustration of the amplification of the unstable mode. Note the period of $4\Delta t$, which is characteristic of this type of instability.

Panel c of Fig. 3.11 shows a plot of $q^{(n)}$ in the complex plane for the modes corresponding to λ_I and λ_2 for $\Omega > 1$. The phase changes by $\frac{\pi}{2}$ on each step, because λ is pure imaginary,

and so the period is $4\Delta t$. Panel d of Fig. 3.11 schematically shows $q^{(n)}$ as a function of n for the amplifying mode corresponding to λ_I , i.e., q_1 is unstable and q_2 is damped. A growing oscillation of this period is a telltale sign of instability with this type of scheme.

In summary, the centered or leapfrog scheme is a second-order scheme that gives a neutral solution for (3.35) when $|\Omega| \leq 1$. For $|\Omega| > 1$, or large Δt , the scheme is unstable. In other words, the leapfrog scheme is conditionally stable when applied to the oscillation

equation.

We have identified another neutral scheme -- the trapezoidal implicit scheme -- but to use such an implicit scheme in more complicated nonlinear problems is relatively difficult, in comparison with an explicit scheme. The leapfrog scheme is explicit, has a higher accuracy than the general rule, and is neutral if $|\Omega| \leq 1$. For this reason it has been very widely used.

3.4.4 The second-order Adams Bashforth Scheme ($m=0, l=1$) for the oscillation equation

The second-order Adams-Bashforth scheme and its third-order cousin (Durrant, 1991) have some very nice properties. The second-order Adams-Bashforth finite-difference approximation to the oscillation equation is

$$q^{n+1} - q^n = i\Omega \left(\frac{3}{2}q^n - \frac{1}{2}q^{n-1} \right). \quad (3.56)$$

Like the leapfrog scheme, this is a three-level scheme. Since $m = 0$, however, the time interval on the left-hand side is Δt , rather than $2\Delta t$ as in the leapfrog scheme. The right-hand side represents a linear extrapolation (in time) of q from $q^{(n-1)}$ and $q^{(n)}$ to $n + \frac{1}{2}$. It essentially represents a scheme centered around time level $n + \frac{1}{2}$, and it does have second-order accuracy. The amplification factor for this scheme is given by

$$\lambda^2 - \lambda \left(1 + \frac{3}{2}i\Omega \right) + i\frac{1}{2}\Omega = 0. \quad (3.57)$$

Since this is a three-time-level scheme, we have two modes, given by

$$\lambda_1 = \frac{1}{2} \left(1 + i\frac{3}{2}\Omega + \sqrt{1 - \frac{9}{4}\Omega^2 + i\Omega} \right), \quad (3.58)$$

and

$$\lambda_2 = \frac{1}{2} \left(1 + i\frac{3}{2}\Omega - \sqrt{1 - \frac{9}{4}\Omega^2 + i\Omega} \right). \quad (3.59)$$

Since $\lambda_1 \rightarrow 1$ as $\Omega \rightarrow 0$, the first mode is the physical mode and corresponds to the true solution as $\Delta t \rightarrow 0$. Note, however, that $\lambda_2 \rightarrow 0$ as $\Omega \rightarrow 0$. This means that the “computational” mode tends to damp. It is not neutral, as in the leapfrog scheme.

In order to examine $|\lambda_1|$, we just consider $\Omega \ll 1$, since the expression in (3.58) is complicated and in practice Ω is usually small. Using the binomial theorem, we can approximate λ_1 by

$$\lambda_1 \cong 1 + i\Omega - \frac{9}{16}\Omega^2 \cong 1 + i\Omega - \frac{1}{2}\Omega^2, \quad (3.60)$$

so that

$$|\lambda_1| \cong \sqrt{1 + \frac{\Omega^4}{4}} \cong 1 + \frac{\Omega^4}{8}. \quad (3.61)$$

which is always greater than 1. The physical mode is, therefore, unconditionally unstable, but so far as Δt or Ω is sufficiently small, and because the deviation of $|\lambda_1|$ from 1 is of order Ω^4 , the solution is only weakly unstable. If physical damping is included in the problem, the instability may be suppressed.

3.4.5 A survey of time differencing schemes for the oscillation equation

Fig. 3.12 and Fig. 3.13 are taken from the work of Baer and Simons (1970). They summarize the properties of seven explicit and seven implicit schemes, which are listed in Table 3.3. Properties of these schemes are shown in Fig. 3.12 and Fig. 3.13.

Table 3.3: List of time differencing schemes surveyed by Baer and Simons (1970). Schemes whose names begin with “E” are explicit, while those whose names begin with “I” are implicit. The numerical indices in the names are “ m ,” which is the number of “time intervals” over which the scheme steps, as defined in Eq. (3.3) and Fig. 3.1; and l , which controls the number of values of f used, again as defined in (3.3).

<i>Scheme identifier (m,l)</i>	<i>Name</i>	β	α_n	α_{n-1}	α_{n-2}	α_{n-3}	α_{n-4}	<i>Order of accuracy</i>
<i>E01</i>	<i>Adams-Bashforth</i>		$3/2$	$-1/2$				$(\Delta t)^2$
<i>E02</i>	<i>Adams-Bashforth</i>		$23/12$	$-4/3$	$5/12$			$(\Delta t)^3$
<i>E03</i>			$55/24$	$-59/24$	$37/24$	$-9/24$		$(\Delta t)^4$
<i>E04</i>			$1901/720$	$-2774/720$	$2616/720$	$-1274/720$	$251/720$	$(\Delta t)^5$
<i>E11</i>	<i>Leapfrog</i>		1					$(\Delta t)^2$
<i>E12</i>			$7/6$	$-2/6$	$1/6$			$(\Delta t)^3$
<i>E33</i>	<i>Milne Predictor</i>		$2/3$	$-1/3$	$2/3$			$(\Delta t)^4$
<i>I00</i>	<i>Trapezoidal Implicit</i>	$1/2$	$1/2$					$(\Delta t)^2$

Table 3.3: List of time differencing schemes surveyed by Baer and Simons (1970). Schemes whose names begin with “E” are explicit, while those whose names begin with “I” are implicit. The numerical indices in the names are “ m ,” which is the number of “time intervals” over which the scheme steps, as defined in Eq. (3.3) and Fig. 3.1; and l , which controls the number of values of f used, again as defined in (3.3).

Scheme identifier (m, l)	Name	β	α_n	α_{n-1}	α_{n-2}	α_{n-3}	α_{n-4}	Order of accuracy
I01		5/12	8/12	-1/12				$(\Delta t)^3$
I02	Moulton Corrector	9/24	19/24	-5/24	1/24			$(\Delta t)^4$
I03		251/720	646/720	-264/720	106/720	-19/720		$(\Delta t)^5$
I12	Milne Corrector	1/6	4/6	1/6				$(\Delta t)^4$
I13		29/180	124/180	24/180	4/180	-1/180		$(\Delta t)^5$
I34	Milne II Corrector	14/180	64/180	24/180	64/180	14/180		$(\Delta t)^6$

There are many other schemes of higher-order accuracy. Books on numerical analysis discuss such schemes. Since our meteorological interest mainly leads us to partial differential equations, the solutions to which will also suffer from truncation error due to space differencing, we cannot hope to gain much by increasing the accuracy of the time differencing only.

3.5 Finite-difference schemes for the decay equation

We now turn our attention to the decay equation,

$$\frac{dq}{dt} = -\kappa q, \quad \kappa > 0, \quad (3.62)$$

which is relevant to many physical parameterizations, including those of the boundary layer, radiative transfer, cloud physics, and convection. The exact solution is

$$q(t) = q(0)e^{-\kappa t}. \quad (3.63)$$

This describes a simple exponential decay with time. For large time, $q \rightarrow 0$. A good scheme should give $q^{n+1} \rightarrow 0$ as $\kappa \Delta t \rightarrow \infty$.

For the Euler (forward) scheme, the finite-difference analogue of (3.62) is

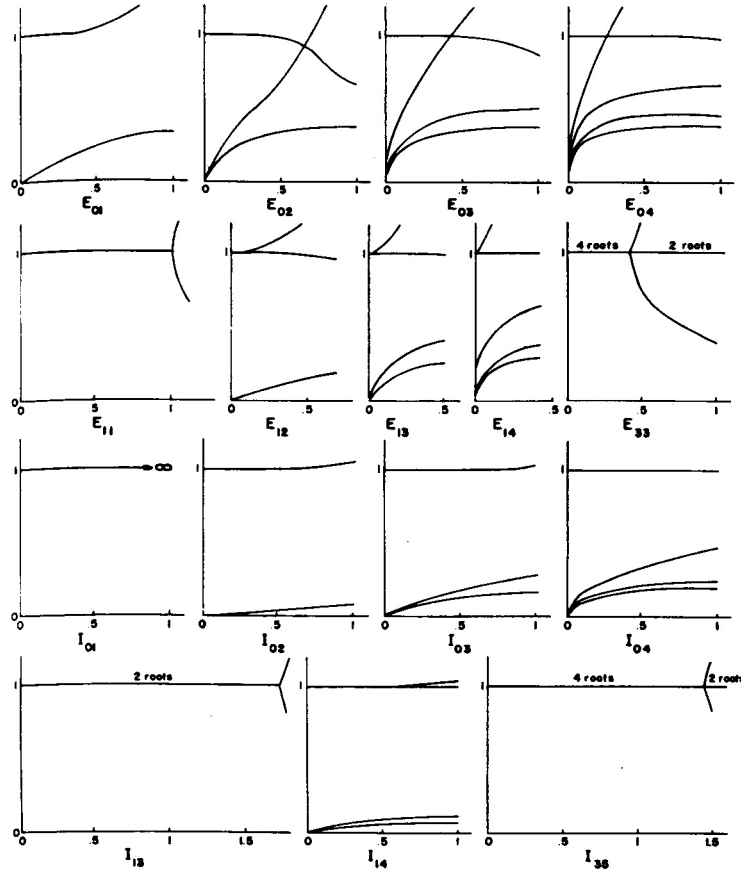


Figure 3.12: Amplification factor of various schemes for the oscillation equation (from Baer and Simons, 1970). The horizontal axis in each panel is Ω . See Table 3.3.

$$q^{n+1} - q^n = -Kq^n, \quad (3.64)$$

where $K \equiv \kappa \Delta t$. The solution is

$$q^{n+1} = (1 - K)q^n. \quad (3.65)$$

Note that $\lambda = 1 - K$ is real. For $|1 - K| < 1$, i.e. κ or Δt small enough, so that $K \leq 2$, the scheme is stable. This is, therefore, a conditionally stable scheme

By the same technique, it is easy to show that, when applied to the decay equation,

- the backward implicit scheme is *unconditionally stable*;

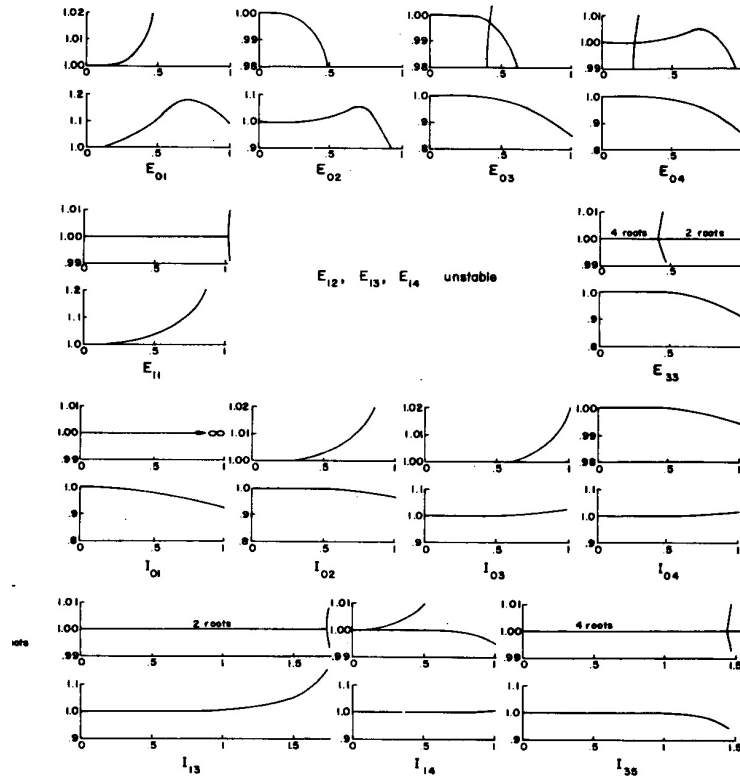


Figure 3.13: $|\lambda|$ and $\frac{\theta}{\Omega}$ (for the physical mode), plotted as functions of Ω , for the oscillation equation (from Baer and Simons, 1970). See Table 3.3.

- the trapezoidal implicit scheme is *unconditionally stable*;
- the Matsuno (Euler-Backward) scheme is *conditionally stable*;
- the Heun scheme is *conditionally stable*; and
- the second-order Adams-Bashforth scheme is *conditionally stable*.

Finally, the leapfrog scheme for the decay equation is

$$q^{n+1} - q^{n-1} = -2Kq^n, \quad (3.66)$$

and so λ satisfies

$$\lambda^2 + 2K\lambda - 1 = 0. \quad (3.67)$$

The two roots are

$$\lambda_1 = -K + \sqrt{K^2 + 1}, \quad \lambda_2 = -K - \sqrt{K^2 + 1}. \quad (3.68)$$

Since $0 \leq \lambda_1 \leq 1$, and $\lambda_1 \rightarrow 1$ as $K \rightarrow 0$, we see that λ_1 corresponds to the physical mode. However, $|\lambda_2| > 1$ always. Actually $\lambda_2 \leq -1$ ($\lambda_2 \rightarrow -1$ as $\Delta t \rightarrow 0$), so the computational mode oscillates in sign from one time level to the next, and amplifies.

Therefore, the leapfrog scheme is *unconditionally unstable* for this type of equation. *We cannot use the leapfrog scheme whenever we have any damping in the problem.* A simple interpretation of this can be given. Suppose we have $q = 0$ at $n = 0$ and $q > 0$ at $n = 1$, as shown schematically in Fig. 3.14. From (3.66) we see that the restoring effect computed at $n = 1$ is added to q^0 , resulting in a negative deviation at $n = 2$. The restoring effect computed at $n = 2$ is added to q^1 , resulting in the amplified positive deviation at $n = 3$, as graphically illustrated in Fig. 3.14. This shows why the leapfrog scheme is a very bad choice for this type of problem.

3.6 Damped oscillations

What should we do if we have an equation of the form

$$\frac{dq}{dt} = i\omega q - \kappa q = (-\kappa + i\omega)q? \quad (3.69)$$

As an example, we can mix the leapfrog and forward (or backward) schemes in the following manner. We write the finite difference analogue of (3.69) as

$$q^{n+1} - q^{n-1} = 2i\Omega q^n - 2Kq^{n-1}, \quad (3.70)$$

(decay term forward differenced), or as

$$q^{n+1} - q^{n-1} = 2i\Omega q^n - 2Kq^{n+1}. \quad (3.71)$$

(decay term backward differenced). The oscillation term on the right-hand sides of both (3.70) and (3.71) is in “centered” form, whereas the damping term is an uncentered form. These schemes are conditionally stable.

3.7 Nonlinear damping

In real applications, it is quite typical that κ depends on q , so that (3.62) becomes nonlinear. Kalnay and Kanamitsu (1988) studied the behavior of ten time-differencing schemes for a nonlinear version of (3.62) given by

$$\frac{dq}{dt} = -(\kappa q^P)q + S. \quad (3.72)$$

$$q^0 = 0$$

$$q^1 > 0$$

$$q^2 = q^0 - 2Kq^1 = 0 - 2Kq^1 < 0$$

$$q^3 = q^1 - 2Kq^2 = q^1 - 2K(q^0 - 2Kq^1) = q^1(1 + 4K^2) > q^1$$

$$q^4 = q^2 - 2Kq^3 < q^2$$

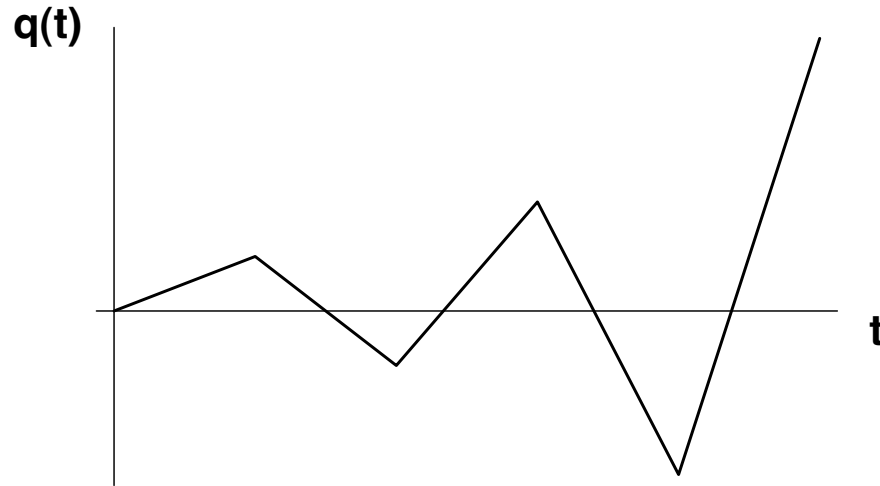


Figure 3.14: An example illustrating how the leapfrog scheme leads to instability with the decay equation. The solution shown here represents the computational mode only and would be superimposed on the physical mode.

where P is a non-negative exponent, and S is a source or sink whose form is unspecified. The reason for introducing S is simply to allow non-zero equilibrium values of q . In real applications, there is usually a term corresponding to S . In case $p = 0$ and $S = 0$, (3.72) reduces to (3.62).

An example of a real application that gives rise to an equation of the form (3.72) is boundary-layer parameterization. The soil temperature, T_g , satisfies an equation roughly of the form

$$C \frac{dT_g}{dt} = -\rho_a c_T (T_g - T_a) V (T_g - T_a) + S_g, \quad (3.73)$$

where C is the heat capacity of the soil layer, ρ_a is the density of the air, $c_T(T_g - T_a)$ is a “transfer coefficient” that depends on $(T_g - T_a)$, V is the wind speed at a level near the ground (often taken to be 10 m above the soil surface), T_a is the temperature of the air at some level near the ground (often taken to be 2 m above the soil surface), and S_g represents all other processes that affect the soil temperature, e.g. solar and infrared radiation, and the latent heat flux, and the conduction of heat through the soil.

The air temperature is governed by a similar equation:

$$\rho_a D c_P \frac{dT_a}{dt} = \rho_a c_T (T_g - T_a) V (T_g - T_a) + S_a \quad (3.74)$$

Here c_P is the heat capacity of air at constant pressure, and D is the depth of the layer of air whose temperature is represented by T_a .

Comparing (3.73) with (3.74), we find that

$$\frac{d(T_g - T_a)}{dt} = -\rho_a c_T (T_g - T_a) V (T_g - T_a) \left(\frac{1}{C} + \frac{1}{\rho_a D c_P} \right) + \left(\frac{S_g}{C} - \frac{S_a}{\rho_a D c_P} \right) \quad (3.75)$$

The correspondence between (3.75) and (3.72) is obvious. The two equations are essentially the same if the transfer coefficient $c_T(T_g - T_a)$ has a power-law dependence on $T_g - T_a$. Virtually all realistic atmospheric models involve equations something like (3.73) and (3.74), so this is a very practical example.

From what we have already discussed, it should be clear that an implicit scheme would be a good choice for (3.72), i.e.

$$\frac{q^{n+1} - q^n}{\Delta t} = -\kappa (q^{n+1})^{P+1} + S. \quad (3.76)$$

Such a scheme is in fact unconditionally stable, but for arbitrary P it must be solved iteratively, which can be quite expensive. For this practical reason, (3.76) would not be considered a viable choice, except where P is a small integer, in which case, (3.76) can be solved analytically.

Let q denote an equilibrium solution of (3.72). Then q satisfies

$$\kappa q^{P+1} = S. \quad (3.77)$$

Let q' denote a departure from this equilibrium, so that $q = \bar{q} + q'$. Then (3.72) can be linearized as follows:

$$\frac{d}{dt}(\bar{q} + q') = -\kappa \bar{q}^{P+1} - \kappa(P+1)\bar{q}^P q' + S, \quad (3.78)$$

which reduces to

$$\frac{dq'}{dt} = -\kappa(P+1)\bar{q}^P q'. \quad (3.79)$$

This linearized equation can be analyzed using von Neumann's method.

As an example, the forward time-differencing scheme, applied to (3.79), gives

$$q^{n+1} - q^n = -\alpha(P+1)q^n, \quad (3.80)$$

where

$$\alpha \equiv \kappa \bar{q}^P \Delta t, \quad (3.81)$$

and we have dropped the “prime” notation for simplicity. We can rearrange (3.80) to

$$q^{n+1} = [1 - \alpha(P+1)]q^n, \quad (3.82)$$

from which we see that

$$\lambda = 1 - \alpha(P+1). \quad (3.83)$$

Table 3.4 summarizes the ten schemes that Kalnay and Kanamitsu analyzed, and

Table 3.4: Schemes for the nonlinear decay equation, as studied by Kalnay and Kanamitsu (1988).

Name of scheme	Form of scheme	Amplification factor	Linear stability criterion
<i>Forward explicit</i>	$\frac{q^{n+1} - q^n}{\Delta t} = -\kappa(q^n)^{P+1} + S$	$1 - \alpha(P+1)$	$\alpha(P+1) < 2$
<i>Backward implicit</i>	$\frac{q^{n+1} - q^n}{\Delta t} = -\kappa(q^{n+1})^{P+1} + S$	$\frac{1}{1 + \alpha(P+1)}$	Unconditionally stable

Table 3.4: Schemes for the nonlinear decay equation, as studied by Kalnay and Kanamitsu (1988).

Name of scheme	Form of scheme	Amplification factor	Linear stability criterion
<i>Centered implicit</i>	$\frac{q^{n+1} - q^n}{\Delta t} = -\kappa \left(\frac{q^n + q^{n+1}}{2} \right)^{P+1} + S$	$\frac{1 - \alpha(P+1)/2}{1 + \alpha(P+1)/2}$	Unconditionally stable
<i>Explicit coefficient, implicit q</i>	$\frac{q^{n+1} - q^n}{\Delta t} = -\kappa(q^n)^P q^{n+1} + S$	$\frac{1 - \alpha P}{1 + \alpha}$	$\alpha(P-1) < 2$
<i>Predictor-corrector coefficient, implicit q</i>	$\frac{\hat{q} - q^n}{\Delta t} = -\kappa(q^n)^P \hat{q} + S$ $\frac{q^{n+1} - q^n}{\Delta t} = -\kappa(\hat{q})^P q^{n+1} + S$	$\frac{1 - \alpha(P-1) + (\alpha P)^2}{(1 + \alpha)^2}$	$\alpha(P-1) < 1$
<i>Average coefficient, implicit q</i>	$\frac{\hat{q} - q^n}{\Delta t} = -\kappa(q^n)^P \hat{q} + S$ $\frac{q^{n+1} - q^n}{\Delta t} = -\kappa \left[\frac{(q^n)^P + (\hat{q})^P}{2} \right] q^{n+1} + S$	$\frac{1 - \alpha(P-1) - \frac{\alpha^2 P}{2} + \frac{(\alpha P)^2}{2}}{(1 + \alpha)^2}$	$\alpha(P-2) < 2$
<i>Explicit coefficient, extrapolated q</i>	$\frac{q^{n+1} - q^n}{\Delta t} = -\kappa(q^n)^P [\gamma q^{n+1} + (1 - \gamma)q^n] + S$	$\frac{1 - \alpha(P+1-\gamma)}{1 + \alpha\gamma}$	$\alpha(P+1-2\gamma) < 2$
<i>Explicit coefficient, implicit q, with time filter</i>	$\frac{\hat{q} - q^n}{\Delta t} = -\kappa(q^n)^P \hat{q} + S$ $q^{n+1} = (1 - A)\hat{q} + Aq^n$	$\frac{(1 - A)(1 - \alpha P)}{1 + \alpha} + A$	$\alpha[P(1 - A) - 1 - A] < 2$
<i>Double time step, explicit coefficient, implicit q with time average filter</i>	$\frac{\hat{q} - q^n}{2\Delta t} = -\kappa(q^n)^P \hat{q} + S$ $q^{n+1} = \frac{\hat{q} + q^n}{2}$	$\frac{1 - \alpha(P-1)}{1 + 2\alpha}$	$\alpha(P-3) < 2$
<i>Linearization of backward implicit scheme</i>	$\frac{q^{n+1} - q^n}{\Delta t} = -\kappa(q^n)^P [(P+1)q^{n+1} - Pq^n] + S$	$\frac{1 + \alpha P}{1 + \alpha(P+1)}$	Unconditionally stable

gives the amplification factors and stability criteria for each. In the table, A is a parameter used to adjust the properties of a time filter; and $\gamma > 1$ is an “extrapolation” parameter. For a detailed discussion, see the paper by Kalnay and Kanamitsu (1988), which you should find quite understandable at this stage. As a recommendation, the “Double time step, explicit coefficient, implicit temperature with time average filter” scheme (the second-last scheme in the table) is quite nice.

3.8 Summary

It is possible to construct time-differencing schemes of arbitrary accuracy by including enough time levels, and/or through iteration. Schemes of very high accuracy (e.g. tenth order) can be constructed quite easily, especially using symbolic algebra programs, but highly accurate schemes involve a lot of arithmetic and so are expensive. In addition they are complicated. An alternative approach to obtain high accuracy is to use a simpler low-order scheme with a smaller time step. This also involves a lot of arithmetic, but on the other hand the small time step makes it possible to represent the temporal evolution in more detail.

More accurate schemes are not always better. For example, the second-order leapfrog scheme is unstable when applied to the decay equation, while the first-order backward implicit scheme is unconditionally stable and well behaved for the same equation. A stable but less accurate scheme is obviously preferable to an unstable but more accurate scheme.

For the advection and oscillation equations, truncation errors can be separated into amplitude errors and phase errors. Neutral schemes, like the leapfrog scheme, have only phase errors.

Computational modes are permitted by differencing schemes that involve three or more time levels. To control these modes, there are four possible approaches:

- 1) Choose a scheme that involves only two time levels;
- 2) Choose the computational initial condition well, and periodically re-start;
- 3) Choose the computational initial condition well, and use a time filter (e.g. Asselin, 1972) to suppress the computational mode;
- 4) Choose the computational initial condition well, and choose a scheme that intrinsically damps the computational mode more than the physical mode (e.g., an Adams-Bashforth scheme).

Appendix to Chapter 3

A Proof that the Fourth-Order Runge-Kutta Scheme has Fourth-Order Accuracy

We wish to obtain an approximate numerical solution of the ordinary differential equation

$$\frac{dq}{dt} = f(q, t) . \quad (3.84)$$

As discussed earlier, the fourth-order Runge-Kutta scheme is given by:

$$q^{(n+1)} - q^{(n)} = \Delta t \frac{1}{6} (k_1 + 2k_2 + 2k_3 + k_4) , \quad (3.85)$$

where

$$\begin{aligned} k_1 &= f(q^{(n)}, n\Delta t) , & k_2 &= f\left[q^{(n)} + \frac{k_1\Delta t}{2}, \left(n + \frac{1}{2}\right)\Delta t\right] , \\ k_3 &= f\left[q^{(n)} + \frac{k_2\Delta t}{2}, \left(n + \frac{1}{2}\right)\Delta t\right] , & k_4 &= f(q^{(n)} + k_3\Delta t, (n+1)\Delta t) . \end{aligned} \quad (3.86)$$

To prove that this scheme has fourth-order accuracy, we construct the right-hand side of (3.85), and then to show that it is equal to the left-hand side, with an error of order $(\Delta t)^5$.

Define an operator δ by

$$\delta \equiv k_1 \frac{\partial}{\partial q} + \frac{\partial}{\partial t} = f \frac{\partial}{\partial q} + \frac{\partial}{\partial t} , \quad (3.87)$$

where we drop the notation that indicates the time level, for convenience. If we apply the operator δ to a function (such as f) which depends on both q and t , then δ returns the total time rate of change of the function, taking into account the part of this tendency that comes from the change in q and also the part that comes from the change in t

$$\delta(f) \equiv \frac{df}{dt} . \quad (3.88)$$

We now write

$$k_1 = f , \quad (3.89)$$

$$\begin{aligned} k_2 &= f \left[q^{(n)} + \frac{k_1 \Delta t}{2}, \left(n + \frac{1}{2} \right) \Delta t \right] \\ &= f + \frac{\Delta t}{2} \delta(f) + \frac{1}{2!} \left(\frac{\Delta t}{2} \right)^2 \delta^2(f) + \frac{1}{3!} \left(\frac{\Delta t}{2} \right)^3 \delta^3(f) + O[(\Delta t)^4] \\ &= \left[1 + \frac{\Delta t}{2} \delta + \frac{1}{2!} \left(\frac{\Delta t}{2} \right)^2 \delta^2 + \frac{1}{3!} \left(\frac{\Delta t}{2} \right)^3 \delta^3 \right] f + O[(\Delta t)^4] , \end{aligned} \quad (3.90)$$

and

$$\begin{aligned}
k_3 &= f \left[q^{(n)} + \frac{k_2 \Delta t}{2}, \left(n + \frac{1}{2} \right) \Delta t \right] \\
&= f + \left(\frac{\Delta t}{2} k_2 \frac{\partial}{\partial q} + \frac{\Delta t}{2} \frac{\partial}{\partial t} \right) f + \frac{1}{2!} \left(\frac{\Delta t}{2} k_2 \frac{\partial}{\partial q} + \frac{\Delta t}{2} \frac{\partial}{\partial t} \right)^2 f + \frac{1}{3!} \left(\frac{\Delta t}{2} k_2 \frac{\partial}{\partial q} + \frac{\Delta t}{2} \frac{\partial}{\partial t} \right)^3 f + O[(\Delta t)^4] \\
&= f + \frac{\Delta t}{2} \left\{ \left[1 + \frac{\Delta t}{2} \delta + \frac{1}{2!} \left(\frac{\Delta t}{2} \right)^2 \delta^2 \right] f \frac{\partial}{\partial q} + \frac{\partial}{\partial t} \right\} f \\
&\quad + \frac{1}{2!} \left(\frac{\Delta t}{2} \right)^2 \left[\left(1 + \frac{\Delta t}{2} \delta \right) f \frac{\partial}{\partial q} + \frac{\partial}{\partial t} \right]^2 f \\
&\quad + \frac{1}{3!} \left(\frac{\Delta t}{2} \right)^3 \left(f \frac{\partial}{\partial q} + \frac{\partial}{\partial t} \right)^3 f + O[(\Delta t)^4] \\
&= f + \frac{\Delta t}{2} \left[\delta(f) + \frac{\Delta t}{2} \delta(f) f_q + \frac{1}{2!} \left(\frac{\Delta t}{2} \right)^2 \delta^2(f) f_q \right] \\
&\quad + \frac{1}{2!} \left(\frac{\Delta t}{2} \right)^2 \left[\delta^2(f) + \frac{\Delta t}{2} \delta(f) f_q \right] \\
&\quad + \frac{1}{3!} \left(\frac{\Delta t}{2} \right)^3 \delta^3(f) + O[(\Delta t)^4] \\
&= f + \Delta t \frac{\delta(f)}{2} + \frac{(\Delta t)^2}{2!} \left[\frac{\delta^2(f)}{4} + \frac{\delta(f) f_q}{2} \right] \\
&\quad + \frac{(\Delta t)^3}{3!} \left[\frac{\delta^3(f)}{8} + \frac{3}{4} \delta(f) f_q + \frac{3}{8} \delta^2(f) f_q \right] + O[(\Delta t)^4] .
\end{aligned} \tag{3.91}$$

In (3.91), we have simplified along the way by suppressing higher-order terms whenever possible, and we have used the notation $f_q \equiv \frac{\partial f}{\partial q}$. It remains to assemble k_4 :

$$\begin{aligned}
k_4 &= (q^{(n)} + k_3 \Delta t, (n+1)\Delta t) \\
&= f + \left(k_3 \Delta t \frac{\partial}{\partial q} + \Delta t \frac{\partial}{\partial t} \right) f + \frac{1}{2!} \left(k_3 \Delta t \frac{\partial}{\partial q} + \Delta t \frac{\partial}{\partial t} \right)^2 f + \frac{1}{3!} \left(k_3 \Delta t \frac{\partial}{\partial q} + \Delta t \frac{\partial}{\partial t} \right)^3 f \\
&= f + \Delta t \left\{ \left[f + \Delta t \frac{\delta(f)}{2} + \frac{(\Delta t)^2}{2!} \left[\frac{\delta^2(f)}{4} + \frac{\delta(f)f_q}{2} \right] \right] \frac{\partial}{\partial q} + \frac{\partial}{\partial t} \right\} f \\
&\quad + \frac{(\Delta t)^2}{2!} \left\{ \left[f + \Delta t \frac{\delta(f)}{2} \right] \frac{\partial}{\partial q} + \frac{\partial}{\partial t} \right\}^2 f + \frac{(\Delta t)^3}{3!} \left(f \frac{\partial}{\partial q} + \frac{\partial}{\partial t} \right)^3 f + O[(\Delta t)^4] \\
&= f + \Delta t \left\{ \delta(f) + \Delta t \frac{\delta(f)}{2} f_q + \frac{(\Delta t)^2}{2!} \left[\frac{\delta^2(f)}{4} + \frac{\delta(f)f_q}{2} \right] f_q \right\} \\
&\quad + \frac{(\Delta t)^2}{2!} [\delta^2(f) + \Delta t \delta(f) \delta(f_q)] + \frac{(\Delta t)^3}{3!} \delta^3(f) + O[(\Delta t)^4] \\
&= f + \Delta t \delta(f) + \frac{(\Delta t)^2}{2!} [\delta^2(f) + \delta(f)f_q] \\
&\quad + \frac{(\Delta t)^3}{3!} \left[\delta^3(f) + 3\delta(f)\delta(f_q) + \frac{3}{4}\delta^2(f)f_q + \frac{3}{2}\delta(f)(f_q)^2 \right] + O[(\Delta t)^4] .
\end{aligned} \tag{3.92}$$

Finally, we combine terms to obtain

$$\begin{aligned}
\Delta t \frac{1}{6} (k_1 + 2k_2 + 2k_3 + k_4) &= f \Delta t + \frac{(\Delta t)^2}{2!} \delta(f) + \frac{(\Delta t)^3}{3!} [\delta^2(f) + \delta(f)f_q] \\
&\quad + \frac{(\Delta t)^4}{4!} [\delta^3(f) + 3\delta(f)\delta(f_q) + \delta^2(f)f_q + \delta(f)(f_q)^2] + O[(\Delta t)^5] .
\end{aligned} \tag{3.93}$$

This can be simplified using

$$f = \frac{dq}{dt} \tag{3.94}$$

$$\delta(f) = \frac{df}{dt} = \frac{d^2 q}{dt^2} , \tag{3.95}$$

$$\delta^2(f) + \delta(f)f_q = \frac{d^2 f}{dt^2} = \frac{d^3 q}{dt^3} , \tag{3.96}$$

$$\delta^3(f) + 3\delta(f)\delta(f_q) + \delta^2(f)f_q + \delta(f)(f_q)^2 = \frac{d^3 f}{dt^3} = \frac{d^4 q}{dt^4} . \quad (3.97)$$

By substituting from (3.94)-(3.97), we can rewrite (3.93) as

$$\Delta t \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) = \frac{dq}{dt}\Delta t + \frac{(\Delta t)^2}{2!}\frac{d^2 q}{dt^2} + \frac{(\Delta t)^3}{3!}\frac{d^3 q}{dt^3} + \frac{(\Delta t)^4}{4!}\frac{d^4 q}{dt^4} + O[(\Delta t)^5] \quad (3.98)$$

Problems

1. a) Find the exact solution of

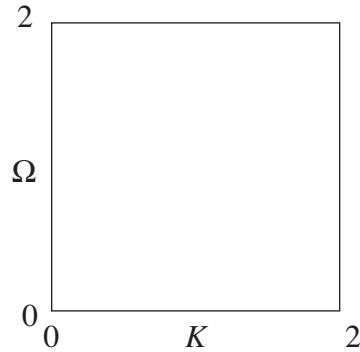
$$\frac{dq}{dt} = i\omega q - \kappa q. \quad (3.99)$$

Let $q(t=0) = 100$, $\frac{\omega}{2\pi} = 0.1$, $\kappa = 0.1$. Plot the real part of the solution for $0 \leq t \leq 100$.

- b) Find the stability criterion for the scheme given by

$$q^{n+1} - q^{n-1} = 2i\Omega q^n - 2Kq^{n+1}. \quad (3.100)$$

Plot the neutral stability boundary (where $|\lambda| = 1$) as a curve in the (K, Ω) plane, for K and Ω in the range 0 to 2, as in the sketch below. Here $\Omega \equiv \omega\Delta t$, $K \equiv \kappa p\Delta$. Indicate which part(s) of the (K, Ω) plot correspond to instability.



- c) Code the equation given in part b) above. Use a forward time step for the first step only. Use $q(t=0) = 100$, and $\Delta t = 1$. Plot the solution out to $t = 100$ for the following cases:

- a) $\frac{\omega}{2\pi} = 0.1$, $\kappa = 0$
- b) $\frac{\omega}{2\pi} = 0$, $\kappa = 0.1$
- c) $\frac{\omega}{2\pi} = 0.1$, $\kappa = 0.1$

For each case, plot $Re\{q\}$ for $0 \leq t \leq 100$ and compare with the exact solution. Discuss the numerical results as they relate to the stability analysis of part).

d) Derive an equation satisfied by the amplification factor for the second-order Adams-Bashforth scheme applied to Eq. (3.99). (The result is quite complicated.) Contour plot $|\lambda|$ as a function of both ω and κ . Find an approximate solution valid for sufficiently small Δt .

2. For the oscillation equation, compare the phase change per time step of:

- a) the leapfrog scheme's physical mode;
- b) the trapezoidal implicit scheme.

Plot the phase change per time step as a function of ω , for both schemes. Discuss the phase errors of the two schemes, as functions of ω .

3. The trapezoidal-implicit scheme for the oscillation equation is given by

$$q^{(n+1)} - q^{(n)} = \frac{i\omega\Delta t}{2}(q^{(n)} + q^{(n+1)}). \quad (3.101)$$

- a) Analyze the stability of this scheme using von Neumann's method.
- b) Find the phase change per time step, and compare with the phase change per Δt in the exact solution.

4. Determine the order of accuracy of the Matsuno scheme.

5. Find the stability criterion for the fourth-order Runge-Kutta scheme applied to the oscillation equation.