25
PDE Waves on Strings and Membranes

In this chapter we explore the numerical solution of wave equations. We have two purposes in mind. First, and especially if you have skipped the discussion of the heat equation in Chap. 24, we wish to give another example of how initial conditions in time are treated with a time stepping *or* leapfrog *algorithm. Second, we wish to demonstrate that once we have a working algorithm for solving a wave equation, we can include considerably more physics than is possible with analytic treatments. Indeed, we will see that solutions can be found even after including friction, variable density, gravity, dispersion and nonlinearities.*

Problem: Recall the demonstration from elementary physics in which a string tied down at both ends is plucked "gently" at one location and a pulse is observed to travel along the string. Likewise, if the string has one end free and you shake it just right, a standing-wave pattern is set up in which the nodes remain in place and the antinodes move just up and down. Your **problem** is develop accurate models for wave propagation on a string, and to see if you can set up traveling- and standing-wave patterns.[1]

25.1
The Hyperbolic Wave Equation (Theory)

Consider a string of length L tied down at both ends (Fig. 25.1). The string has a constant density ρ per unit length, a constant tension T, is subject to no frictional forces, and the tension is high enough that we may ignore the sagging of the string due to gravity. We assume that the displacement of the string $y(x,t)$ from its rest position is in the vertical direction only, and that it is a function of the horizontal location along the string x and the time t.

To obtain a simple, linear equation of motion (nonlinear wave equations are discussed in Unit II), we assume that the string's relative displacement $y(x,t)/L$ and slope $\partial y/\partial x$ are small. We isolate an infinitesimal section Δx of the string (Fig. 25.2). We see that the difference in the vertical components of

[1] Some similar, but independent, studies can also be found in [74].

Computationyal Physics. Problem Solving with Computers (2nd edn).
Rubin H. Landau, Manuel José Páez, Cristian C. Bordeianu

ISBN: 978-3-527-40626-5

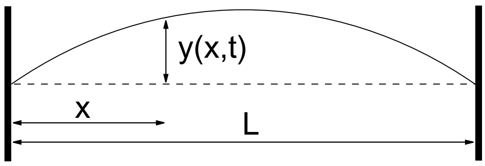

Fig. 25.1 A stretched string of length L tied down at both ends, under high enough tension to ignore gravity. The vertical disturbance of the string from its equilibrium position is $y(x,t)$.

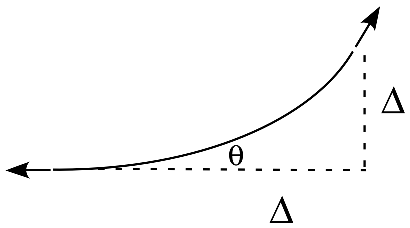

Fig. 25.2 A differential element of the string showing how the string's displacement leads to the restoring force.

the tension on either end of the string produces the restoring force that accelerates this section of the string in the vertical direction. By applying Newton's laws to this section, we obtain the familiar wave equation:

$$\sum F_y = \rho\,\Delta x\,\frac{\partial^2 y}{\partial t^2} \tag{25.1}$$

$$\sum F_y = T\sin[\theta(x+\Delta x)] - T\sin[\theta(x)] \tag{25.2}$$

$$= T\left.\frac{\partial y}{\partial x}\right|_{x+\Delta x} - T\left.\frac{\partial y}{\partial x}\right|_{x} \simeq T\frac{\partial^2 y}{\partial x^2} \tag{25.3}$$

$$\Rightarrow \qquad \frac{\partial^2 y(x,t)}{\partial x^2} = \frac{1}{c^2}\frac{\partial^2 y(x,t)}{\partial t^2} \qquad c = \sqrt{T/\rho} \tag{25.4}$$

where we have assumed that θ is small enough for $\sin\theta \simeq \tan\theta = \partial y/\partial x$. The existence of two independent variables x and t makes this a PDE. The constant c is the velocity with which a disturbance travels along the wave, and is seen to decrease for a heavier string, and increase for a tighter one. Note, this signal velocity c is *not* the same as the velocity of a string element $\partial y(x,t)/\partial t$.

The initial conditions for our problem is that the string is plucked gently and released. We assume that the "pluck" places the string into a triangular shape with the center of triangle 8/10th of the way down the string and with a height of 1:

$$y(x, t=0) = \begin{cases} 1.25x/L & x \leq 0.8L \\ (5 - 5x/L), & x > 0.8L \end{cases} \qquad \text{(initial condition 1)} \tag{25.5}$$

Because (25.4) is second-order in time, a second initial condition (beyond initial displacement) is needed to determine the solution. We interpret the "gentleness" of the pluck to mean the string is released from rest:

$$\frac{\partial y}{\partial t}(x, t=0) = 0 \qquad \text{(initial condition 2)} \tag{25.6}$$

The boundary conditions for our problem are that both ends of the string are tied down for all times:

$$y(0, t) \equiv 0 \qquad y(L, t) \equiv 0 \qquad \text{(boundary conditions)} \tag{25.7}$$

25.1.1 Solution via Normal Mode Expansion

The "analytic solution" to (25.4) is obtained via the familiar separation-of-variables technique. We assume that the solution is the product of a function of space times a function of time:

$$y(x, t) = X(x)T(t) \tag{25.8}$$

We substitute (25.8) into (25.4), divide by $y(x, t)$, and are left with an equation that has a solution only if there are solutions to the two ODEs:

$$\frac{d^2T(t)}{dt^2} + \omega^2 T(t) = 0 \tag{25.9}$$

$$\frac{d^2X(x)}{dt^2} + k^2 X(x) = 0 \qquad k \stackrel{\text{def}}{=} \frac{\omega}{c} \tag{25.10}$$

The angular frequency ω and the wave vector k are determined by demanding that the solutions satisfy the boundary conditions. Specifically, the string being attached at both ends demands

$$X(x=0, t) = X(x=l, t) = 0 \tag{25.11}$$

$$\Rightarrow \quad X_n(x) = A_n \sin k_n x \qquad k_n = \frac{\pi(n+1)}{L} \qquad n = 0, 1, \ldots \tag{25.12}$$

The time solution is

$$T_n(t) = C_n \sin \omega_n t + D_n \cos \omega_n t \qquad \omega_n = nck_0 = n\frac{2\pi c}{L} \tag{25.13}$$

where the frequency of this nth *normal mode* is also fixed. In fact, it is the single frequency of oscillation that defines a normal mode.

The *initial condition* (25.5) of zero velocity $\partial y/\partial t(t = 0) = 0$, requires the C_n values in (25.13) to be zero. Putting the pieces together, the normal-mode solutions are

$$y_n(x,t) = \sin k_n x \, \cos \omega_n t \qquad n = 0, 1, \ldots . \tag{25.14}$$

Since the wave equation (25.4) is linear in y, the principle of linear superposition holds and the most general solution for waves on a string with fixed ends can be written as the sum of normal modes:

$$y(x,t) = \sum_{n=0}^{\infty} B_n \, \sin k_n x \, \cos \omega_n t \tag{25.15}$$

The Fourier coefficient B_n is determined by using the second initial condition (25.5), which describes how the wave is plucked. We start with

$$y(x, t = 0) = \sum_{n}^{\infty} B_n \, \sin n k_0 x, \tag{25.16}$$

multiply both sides by $\sin m k_0 x$, substitute the value of $y(x, 0)$ from (25.5), and integrate from 0 to l to obtain

$$B_m = 6.25 \frac{\sin(0.8 m\pi)}{m^2 \pi^2}. \tag{25.17}$$

You will be asked to compare the Fourier series (25.15) to our numerical solution. While it is in the nature of the approximation that the precision of the numerical solution depends on the choice of step sizes, it is also revealing to realize that the precision of the "analytic" solution depends on summing an infinite number of terms, which can only be done only approximately.

25.1.2
Algorithm: Time Stepping (Leapfrog)

As with the Laplace and heat equations, we look for a solution $y(x,t)$ only for discrete values of the independent variables x and t on a grid (Fig. 25.3):

$$x = i\Delta x \qquad i = 1, \cdots N_x \qquad t = j\Delta t \qquad j = 1, \cdots N_t \tag{25.18}$$

$$y(x,t) = y(i\Delta x, \, i\Delta t) \stackrel{\text{def}}{=} y_{i,j} \tag{25.19}$$

In contrast to Laplace's equation where the grid was in two space dimensions, the grid in Fig. 25.3 is in both space and time. That being the case, moving across a row corresponds to increasing x values along the string for a fixed

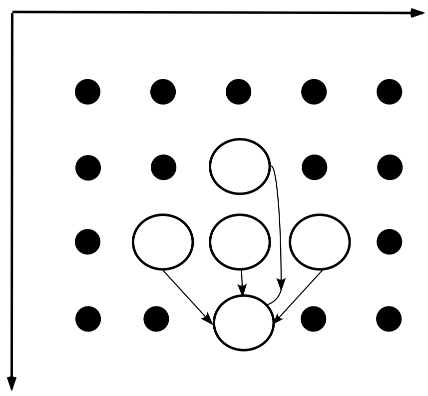

Fig. 25.3 The solutions of the wave equation for four earlier space–time points are used to obtain the solution at the present time.

time, while moving down a column corresponds to increasing time steps for a fixed position. Even though the grid in Fig. 25.3 may be square, we cannot use a relaxation technique for the solution because we do not know the solution on all four sides. The boundary conditions determine the solution along the right and left sides, while the initial time condition determines the solution along the top.

As with the Laplace equation, we use the central difference approximation to *discretize* the wave equation into a difference equation. First we express the second derivatives in terms of finite differences:

$$\frac{\partial^2 y}{\partial t^2} \simeq \frac{y_{i,j+1} + y_{i,j-1} - 2y_{i,j}}{(\Delta t)^2} \tag{25.20}$$

$$\frac{\partial^2 y}{\partial x^2} \simeq \frac{y_{i+1,j} + y_{i-1,j} - 2y_{i,j}}{(\Delta x)^2} \tag{25.21}$$

Substituting (25.20) in the wave equation (25.4) yields the difference equation

$$\frac{y_{i,j+1} + y_{i,j-1} - 2y_{i,j}}{c^2(\Delta t)^2} = \frac{y_{i+1,j} + y_{i-1,j} - 2y_{i,j}}{(\Delta x)^2}. \tag{25.22}$$

Notice that this equation contains three time value: `j+1` = the future, `j` = the present, and `j-1` = the past. Consequently, we rearrange it into a form that permits us to predict the future solution from the present and past solutions:

$$y_{i,j+1} = 2y_{i,j} - y_{i,j-1} + \frac{c^2}{c'^2}\left[y_{i+1,j} + y_{i-1,j} - 2y_{i,j}\right] \tag{25.23}$$

$$c' = \Delta x/\Delta t \tag{25.24}$$

Here c' is a combination of numerical parameters with the dimension of velocity, whose size relative to c determines the stability of the algorithm. The algorithm (25.23) propagates the wave from the two earlier times, j and $j-1$, and from three nearby positions, $i-1$, i, and $i+1$, to a later time $j+1$ and a single space position i (Fig. 25.3).

As you have seen in our discussion of the heat equation, a leapfrog method is quite different from a relaxation technique. We start with the solution along the topmost row, and them move down forward, one step at a time. If we write out the solution for present times to a file, then we need to store only three time values on the computer, which this saves memory. In fact, because the time steps must often be quite small to obtain high precision, you may only want to store the solution for every fifth or tenth times.

Initializing the recurrence relation is a bit tricky because it requires displacements from two earlier times, whereas the initial conditions are only for one time. Nonetheless, the rest condition (25.5), when combined with the *forward-difference* approximation, lets us extrapolate to negative time:

$$\frac{\partial y}{\partial t}(x,0) \simeq \frac{y(x,0) - y(x,-\Delta t)}{\Delta t} = 0 \tag{25.25}$$

$$\Rightarrow \qquad y_{i,0} = y_{i,1} \tag{25.26}$$

Here we take the initial time as $j = 1$, and so $j = 0$ corresponds to $t = -\Delta t$. Substituting this relation into (25.23) yields for the initial step

$$y_{i,2} = y_{i,1} + \frac{c^2}{c'^2}\,[y_{i+1,1} + y_{i-1,1} - 2y_{i,1}] \qquad (t = \Delta t \text{ only}) \tag{25.27}$$

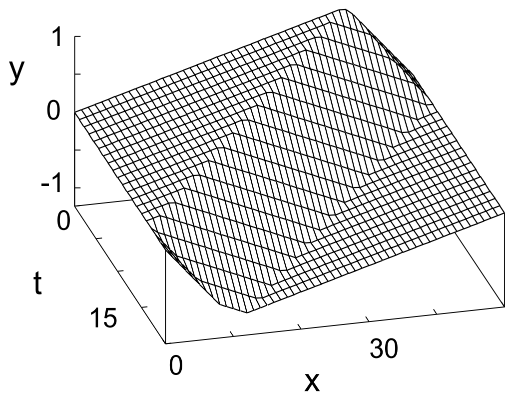

Fig. 25.4 The vertical displacement as a function of position x and time t for a string initially plucked near its right end forms a pulse that divides into waves traveling to the right and to the left.)

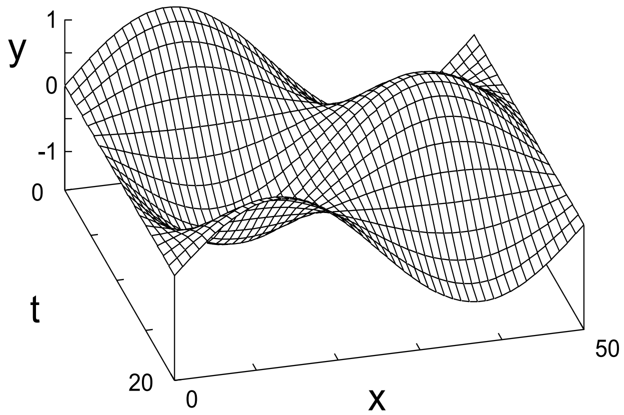

Fig. 25.5 The vertical displacement as a function of position x and time t for a string initially placed in a normal mode. Notice how the standing wave moves up and down with time. (In this and Fig. 25.4, y/L should be small, but that is harder to show.)

Equation (25.27) uses the solution throughout all of space at the initial time $t = 0$ to propagate (leapfrog) it forward to a time Δt. Subsequent time steps use (25.23) and are continued for as long as you like.

As is also true with the heat equation, the success of the numerical method depends on the relative sizes of the time and space steps. If we apply a von Neumann stability analysis to this problem by substituting $y_{m,j} = \xi^j \exp(ikm\Delta x)$, as we did in Section 24.4, a more complicated equation results. Nonetheless, [9] shows that the difference-equation solution will be stable for the general class of transport equations if

$$c \le c' = \Delta x/\Delta t \qquad \text{(Courant condition).} \tag{25.28}$$

Equation (25.28) means that the solution gets better with smaller *time* steps, but gets worse for smaller space steps (unless you simultaneously make the time step smaller). Having different sensitivities to the time and space steps may appears surprising because the wave equation (25.4) is symmetric in x and t, yet the symmetry is broken by the nonsymmetric initial and boundary conditions.

Exercise: Figure out a procedure for solving for the wave equation for all times in just one step. Estimate how much memory would be required for that. □

Exercise: Can you figure out a procedure solving for the wave motion with a relaxation technique? What would you take as your initial guess, and how would you know when the procedure has converged? □

25.1.3 Wave Equation Implementation

The program `EqString.java` in Listing 25.1 solves the wave equation for a string of length $L = 1$ m with its ends fixed and with the gently-plucked initial conditions. Note, $L = 1$ violates the assumption that $y/L << 1$, but makes it easy to display the results; you should try $L = 1000$ to be realistic. The values of density and tension are entered as constants, $\rho = 0.01$ kg/m, $T = 40$ N, with the space grid set at 101 points, corresponding to $\Delta = 0.01$ cm.

25.1.4 Assessment and Exploration

1. Run the simulation and make a surface plot of the results.
2. Explore a number of space and time step combinations. In particular, try steps that satisfy and that do not satisfy the Courant condition (25.28). Does your exploration agree with the stability condition?
3. Compare the analytic and numeric solutions, summing at least 200 terms in the "analytic" solution.
4. Use the time dependence of your graph to estimate the peak's propagation velocity c. Compare the deduced c to (26.6).
5. Our solution of the wave equation for a plucked string leads to the formation of a wave packet, which corresponds to multiple normal modes of the string. In Fig. 25.5 we show the motion of a string for the initial conditions,

$$y(x, t = 0) = 0.001 \sin 2\pi x \qquad \frac{\partial y}{\partial t}(x, t = 0) = 0 \quad (25.29)$$

   which excite just one normal mode. Modify the program to incorporate this initial condition and see if a normal mode results.
6. Observe the motion of the wave for initial conditions corresponding the sum of two adjacent normal modes. Does beating occur?
7. When a string is plucked near its end, a pulse reflects off the ends and bounces back and forth. Change the initial conditions of the model program to one corresponding to a string plucked exactly in its middle, and see if a traveling or a standing wave results.
8. (Optional) Figs. 25.6 and 25.7 show the wave packets that result as a function of time for initial conditions corresponding to the double pluck indicated on the left of the figure. Verify that initial conditions of the form

Listing 25.1: `EqString.java` solves the wave equation via time stepping for a string of length $L = 1$ m with its ends fixed and with the gently-plucked initial conditions. You will need to modify this code to include new physics.

```java
//  Eqstring.java: Solution of wave equation via time stepping
//                 Output in 3D gnuplot format

import java.io.*;

public class Eqstring  {
  final static double rho = 0.01, ten = 40., max = 100.;

  public static void main(String[] argv)
                         throws IOException, FileNotFoundException  {
    int i, k;
    double x[][] = new double[101][3];
    double ratio, c, c1;
    PrintWriter w = new PrintWriter                  // Save data in file
                        (new FileOutputStream("EqString.dat"), true);
    c = Math.sqrt(ten/rho);                          // Propagation speed
    c1 = c;                                              // CFL criteria
    ratio =  c*c/(c1*c1);
                                                // Initial configuration
    for ( i=0;  i < 81;   i++ ) x[i][0] = 0.00125*i;
    for ( i=81; i < 101;  i++ ) x[i][0] = 0.1-0.005*(i-80);
                                                      // First time step
    for ( i=1;  i < 100;  i++ ) {
      x[i][1] = x[i][0]+0.5*ratio*(x[i+1][0]+x[i-1][0]-2.*x[i][0]);}
                                                     // Later time steps
      for ( k=1;  k < max;  k++ )  {
        for ( i=1;  i < 100;  i++ )  x[i][2] = 2.*x[i][1]
                -x[i][0] + ratio*(x[i+1][1] + x[i-1][1] - 2*x[i][1]);
        for ( i=0;  i < 101;  i++ )  {
          x[i][0] = x[i][1];
          x[i][1] = x[i][2];
        }
        if ((k%5) == 0) {                         // Print every 5th point
          for ( i=0;  i < 101;  i++ ) {
            w.println("" + x[i][2] + ""); }          // Gnuplot 3D format
            w.println("");                        // Empty line for gnuplot
        }
      }
      System.out.println("data in EqString.dat, gnuplot format");
  }
}
```

$$\frac{y(x,t=0)}{0.005} = \begin{cases} 0 & 0.0 \le x \le 0.1 \\ 10x - 1 & 0.1 \le x \le 0.2 \\ -10x + 3 & 0.2 \le x \le 0.3 \\ 0, & 0.3 \le x \le 0.7 \\ 10x - 7 & 0.7 \le x \le 0.8 \\ -10x + 9 & 0.8 \le x \le 0.9 \\ 0 & 0.9 \le x \le 1.0 \end{cases} \tag{25.30}$$

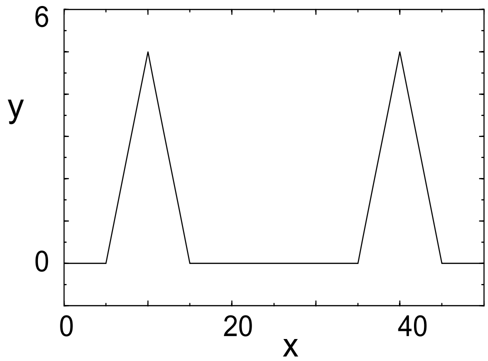

Fig. 25.6 The initial configuration of a string plucked in two places simultaneously.

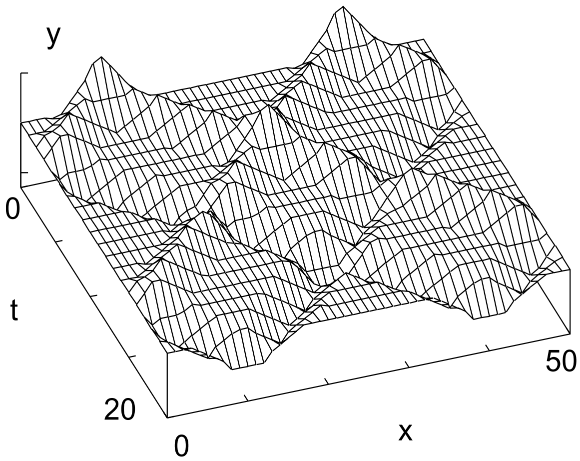

Fig. 25.7 The vertical displacement as a function of position and time of a string initially plucked simultaneously at two points, as shown on the left. Note that each initial peak breaks up into waves traveling to the right and to the left. The traveling waves invert on reflection from the fixed end. As a consequence of these inversions, the $t = 15$ wave is an inverted $t = 0$ wave.

leads to this type of a repeating pattern. In particular, observe whether the pulses move or just oscillate up and down.

25.1.5
Including Friction (Extension)

The string problem we have investigated so far can be handled by either numerical or analytic techniques. We now wish to extend the theory to include some more realistic physics. These extensions have only numerical solutions.

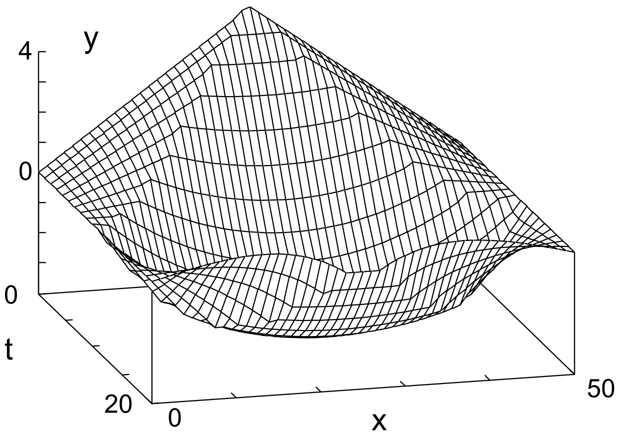

Fig. 25.8 The vertical displacement as a function of position and time of a string with friction initially plucked at its middle.

Real plucked strings do not vibrate forever because the real world contains friction. Consider again the element of string between x and $x + dx$ (Fig. 25.2), but imagine now that this elements is moving in a viscous fluid, such as air. An approximate model for the frictional force is to have it point in a direction opposite to the (vertical) velocity of the string, proportional to that velocity, as well as proportional to the length of the element:

$$F_f \simeq -2\kappa \Delta x \frac{\partial y}{\partial t} \tag{25.31}$$

Here κ is a constant that is proportional to the viscosity of the medium in which the string is vibrating. Including this force in the equation of motion changes the wave equation to

$$\frac{\partial^2 y}{\partial t^2} = c^2 \frac{\partial^2 y}{\partial x^2} - \frac{2\kappa}{\rho} \frac{\partial y}{\partial t} \tag{25.32}$$

In Figs. 25.8 and 25.9 we show the resulting motion of a string plucked in the middle when friction is included. Observe how the initial pluck breaks up into waves traveling to the right and left that get reflected and inverted by the fixed ends. Because those parts of the wave with the greatest velocity experience the greatest friction, the peak tends to get smoothed out the most as time progresses.

Exercise: Generalize the algorithm for the wave equation to include friction and observe the change in wave's behavior. Start off with $T = 40$ N, $\rho = 10$ kg/m, and pick a value of κ that is large enough to cause a noticeable effect, but not so large as to stop the oscillations. As a check, reverse the sign of κ and

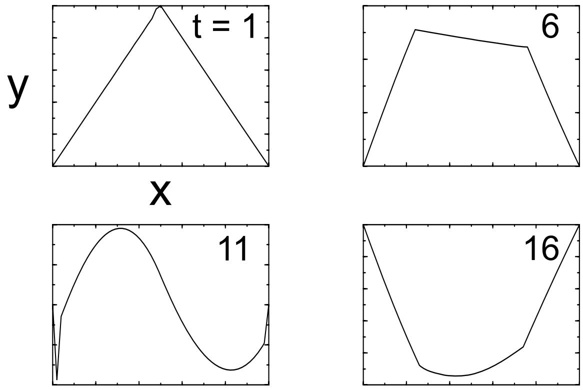

Fig. 25.9 Disturbance versus position for a string with variable density that is initially plucked at its center. The disturbances at four times are given. At $t = 6$ we see that the wave moves faster in the denser region to the right, but that its amplitude decreases because the string is heavier there.

see if the wave grows in time (which would eventually violate our assumption of small oscillations). □

25.1.6 Variable Tension and Density (Extension)

We have derived the propagation velocity for waves on a string as $c = \sqrt{T/\rho}$. This says that waves moves slower in regions of high density, and faster in regions of high tension. If the density of the string varies, to illustrate, by having the ends thicker in order to support the weight of the middle, then c will no longer be a constant and our wave equation needs fixing. In addition, if the density varies, then the tension would too because it takes a greater tension to support a greater mass. If gravity acts, then we would also expect that the tension at the ends of the string to be higher because they must support the entire weight of the string.

To derive the proper equation for wave motion, consider again an element of string (Fig. 25.2), and our derivation of the wave equation. If we ignore friction but do not assume the tension T is constant, then Newton's second law gives

$$F = ma$$

$$\Rightarrow \quad \frac{\partial}{\partial x}\left[T(x)\frac{\partial y(x,t)}{\partial x}\right]\Delta x = \rho(x)\Delta x\frac{\partial^2 u(x,t)}{\partial t^2} \tag{25.33}$$

$$\Rightarrow \quad \frac{\partial T(x)}{\partial x}\frac{\partial y(x,t)}{\partial x} + T(x)\frac{\partial^2 y(x,t)}{\partial x^2} = \rho(x)\frac{\partial^2 y(x,t)}{\partial t^2} \tag{25.34}$$

If $\rho(x)$ and $T(x)$ are known functions, then these equations can be solved with just a small modification of our algorithm.

Deducing a $T(x)$ for a given $\rho(x)$ in the presence of gravity is a hard statics problem because it also requires the solution for equilibrium shape of the string. While we solve that problem below, for those interested in an easier problem that still shows the new physics, you may assume that the density and tension are proportional:

$$\rho(x) = \rho_0 e^{\alpha x} \qquad\qquad T(x) = T_0 e^{\alpha x}. \tag{25.35}$$

While we would expect the tension to be greater in regions of higher density (more mass to move), being proportional is just an approximation. Substitution of these relations into (25.34) yields the new wave equation:

$$\frac{\partial^2 y(x,t)}{\partial x^2} + \alpha \frac{\partial y(x,t)}{\partial x} = \frac{1}{c^2}\frac{\partial^2 y(x,t)}{\partial t^2} \qquad\qquad c^2 = \frac{T_0}{\rho_0}. \tag{25.36}$$

Here c is a constant that would be the wave velocity if $\alpha = 0$. This equation is similar to the wave equation with friction, only now the first derivative is with respect to x, and not t. The corresponding difference equations follow from using the central difference approximations for the derivatives:

$$y_{i,j+1} = 2y_{i,j} - y_{i,j-1} + \frac{\alpha c^2(\Delta t)^2}{2\Delta x}[y_{i+1,j} - y_{i,j}] + \frac{c^2}{c'^2}[y_{i+1,j} + y_{i-1,j} - 2y_{i,j}]$$

$$y_{i,2} = y_{i,1} + \frac{c^2}{c'^2}[y_{i+1,1} + y_{i-1,1} - 2y_{i,1}] + \frac{\alpha c^2(\Delta t)^2}{2\Delta x}[y_{i+1,1} - y_{i,1}] \tag{25.37}$$

25.2 Realistic 1D Wave Exercises

Do these exercises for the assumed density and tension given by (25.35). Include friction in order to make the simulation realistic. Assume $\alpha = 0.5$, $T_0 = 40$ N, and $\rho_0 = 0.01$ kg/m.

1. Modify the algorithm in your program to handle variable tension and density, and friction. Run some typical cases and create surface plots of the results.

2. Explain in words how the waves dampen and how a wave's velocity appears to change. The behavior you obtain may look something like that shown in Fig. 25.8.

3. **Normal Modes:** Search for normal mode solutions of the variable-tension wave equation, that is, solutions that vary like

$$u(x,t) = A\cos(\omega t)\sin(\gamma x)$$

Try using this form to start off your algorithm and see if you can find standing waves. Use large values for ω.

4. When conducting physics demonstrations, we set up standing wave patterns by driving one end of the string periodically. Try doing the same with your algorithm; that is, build into your code the conditions that

$$y(x = 0, t) = A \sin \omega t$$

for all times. Try to vary A and ω until a normal mode (standing wave) is obtained.

5. If you were able to find standing waves, then verify that this string acts like a high-frequency filter, namely, that there is a frequency below which no waves occur. (This is for exponential density case.)

6. For the catenary problem, plot up your results showing *both* the disturbance $u(x,t)$ about the catenary and the actual height $y(x,t)$ above the horizontal for a plucked string initial condition.

7. Try the first two normal modes for a uniform string as the initial conditions for the catenary. These should be close to, but not exactly normal modes.

8. We derived the normal modes for a uniform string after assuming that $k(x) = \omega/c(x)$ is a constant. For a catenary without too much x variation of the tension, we should be able to make the approximation

$$c(x)^2 \simeq \frac{T(x)}{\rho} = \frac{T_0 \cosh(x/d)}{\rho}$$

See if you get a better representation of the first two normal modes if you include some x dependence to k.

25.3
Vibrating Membrane (2D Waves)

Problem: An elastic membrane is stretched across the top of a square box of sides π and attached securely. The tension per unit length in the membrane is T. Initially the membrane is placed in the asymmetrical shape,

$$u(x, y, t = 0) = \sin 2x \sin y \qquad 0 \le x \le \pi \quad 0 \le y \le \pi \tag{25.38}$$

where u is the vertical displacement from equilibrium. Your **problem** is to describe the motion of the membrane when it is released from rest [71].

The description of wave motion on a membrane is basically the same as that of 1D waves on a string discussed in Section 25.1, only now we have wave propagation in two directions. Consider Fig. 25.10 showing a square section of the membrane under tension T. The membrane only moves vertically in the z direction, yet because the restoring force arising from the tension in the membrane varies in both the x and y directions, there is wave motion along the surface of the membrane.

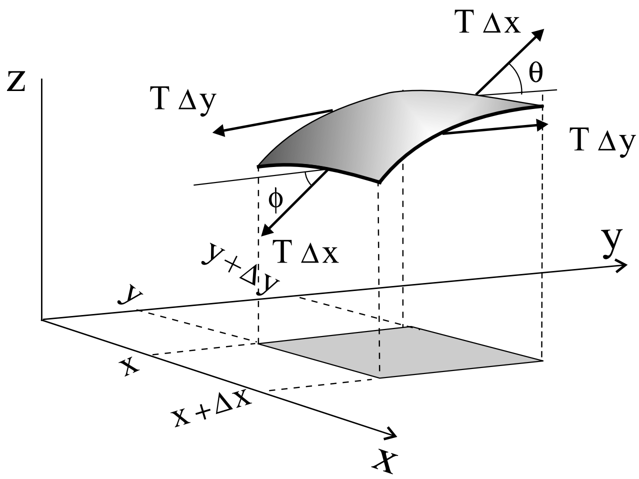

Fig. 25.10 A small part of an oscillating membrane and the forces that act on it.

Although the tension is constant over the small area in Fig. 25.10, there will be a net vertical force on the segment if the angle of incline of the membrane varies as we move through space. Accordingly, the net force on the membrane in the z direction due to the change in y is

$$\sum F_z(x = \text{constant}) \;=\; T\Delta x\, \sin\theta - T\,\Delta x\, \sin\phi \tag{25.39}$$

where θ is the angle of incline at $y + \Delta y$ and ϕ the angle at y. Yet if we assume that the angles are small (small displacements), then we can make the approximations:

$$\sin\theta \;\approx\; \tan\,\theta \;=\; \left.\frac{\partial u}{\partial y}\right|_{y+\Delta y} \tag{25.40}$$

$$\sin\phi \;\approx\; \tan\phi \;=\; \left.\frac{\partial u}{\partial y}\right|_{y}$$

$$\Rightarrow \qquad \sum F_z(x = \text{constant}) \;=\; T\Delta x \left(\left.\frac{\partial u}{\partial y}\right|_{y+\Delta y} - \left.\frac{\partial u}{\partial y}\right|_{y} \right) \;\approx\; T\Delta x\, \frac{\partial^2 u}{\partial y^2}\Delta y$$

Similarly, the net force in the z direction due to the variation in y is

$$\sum F_z(y = \text{constant}) = T\Delta y \left(\frac{\partial u}{\partial x}\Big|_{x+\Delta x} - \frac{\partial u}{\partial x}\Big|_{x} \right) \tag{25.41}$$

$$\approx T\Delta y \frac{\partial^2 u}{\partial x^2}\Delta x \tag{25.42}$$

The membrane section has mass $\rho\Delta x\Delta y$, where ρ is the membrane's mass per unit area. We now apply Newton's second law to determine the acceleration of the membrane section in the z direction motion due to the sum of the net forces arising from both the x and y variations:

$$\rho\Delta x\Delta y \frac{\partial^2 u}{\partial t^2} = T\Delta x \frac{\partial^2 u}{\partial y^2}\Delta y + T\Delta y \frac{\partial^2 u}{\partial x^2}\Delta x,$$

$$\Rightarrow \qquad \frac{1}{c^2}\frac{\partial^2 u}{\partial t^2} = \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \qquad c = \sqrt{T/\rho} \tag{25.43}$$

This is the 2D version of the wave equation (25.4) that we studied previously in one dimension. Here c, the propagation velocity, is still the square root of tension over density, only now it is tension per unit length and mass per unit area.

25.4
Analytical Solution

The analytic or numerical solution of the partial differential equation (25.43) requires us to know both the boundary conditions and the initial conditions. The boundary conditions for our problem hold for all times and are given when we are told that the membrane is attached securely to a square box:

$$u(x=0,y,t) = u(x=\pi,y,t) = 0 \tag{25.44}$$
$$u(x,y=0,t) = u(x,y=\pi,t) = 0$$

The initial conditions has two parts, the shape of the membrane at time $t = 0$, and the velocity of each point of the membrane. The initial configuration is

$$u(x,y,t=0) = \sin 2x \sin y \qquad 0 \le x \le \pi \quad 0 \le y \le \pi \tag{25.45}$$

Second, we are told that the membrane is released from rest, which means:

$$\frac{\partial u}{\partial t}\Big|_{t=0} = v(x,y,t=0) = 0, \tag{25.46}$$

where we write partial derivative because there are also spatial variations.

The analytic solution is based on the guess that because the wave equation (25.43) has the derivatives with respect to each coordinate and time separate, the full solution $u(x,y,t)$ is the product of separate functions of x, y and t:

$$u(x,y,t) = X(x)\,Y(y)\,T(t) \tag{25.47}$$

After substituting into (25.43) and dividing by $X(x)Y(y)T(t)$, we obtain:

$$\frac{1}{c^2}\frac{1}{T(t)}\frac{d^2T(t)}{dt^2} = \frac{1}{X(x)}\frac{d^2X(x)}{dx^2} + \frac{1}{Y(y)}\frac{d^2Y(y)}{dy^2} \tag{25.48}$$

The only way that the LHS of (25.48) can depends only on time while the RHS depends only on coordinates, is if both sides are constant:

$$\frac{1}{c^2}\frac{1}{T(t)}\frac{d^2T(t)}{dt^2} = -\xi^2 = \frac{1}{X(x)}\frac{d^2X(x)}{dx^2} + \frac{1}{Y(y)}\frac{d^2Y(y)}{dy^2} \tag{25.49}$$

$$\Rightarrow \quad \frac{1}{X(x)}\frac{d^2X(x)}{dx^2} = -k^2 \tag{25.50}$$

$$\frac{1}{Y(y)}\frac{d^2Y(y)}{dy^2} = -q^2 \qquad \text{where } q^2 + k^2 = \xi^2 \tag{25.51}$$

In (25.50) and (25.51) we have included the further deduction that since each term on the RHS of (25.49) depends on either x or y, then the only way their sum can be constant is if each term is a constant, in this case $-k^2$.

The solutions of these equations are standing waves in the x and y directions, which of course are all sinusoidal function,

$$\begin{aligned} X(x) &= A\sin kx + B\cos kx \\ Y(y) &= C\sin qy + D\cos qy \\ T(t) &= E\sin c\xi t + F\cos c\xi t \end{aligned}$$

We now apply the boundary conditions:

$$u(x=0,y,t) = u(x=\pi,y,z) = 0 \quad \Rightarrow \quad B = 0 \quad k = m = 1,2,\cdots$$

$$u(x,y=0,t) = u(x,y=\pi,t) = 0 \quad \Rightarrow \quad D = 0 \quad y = n = 1,2,\cdots$$

$$\Rightarrow \quad X(x) = A\sin mx \qquad Y(y) = C\sin ny$$

The fixed values for the eigenvalues m and n describing the modes for the x and y standing waves are equivalent to fixed values for the constants q^2 and k^2. Yet since $q^2 + k^2 = \xi^2$, we must also have a fixed value for ξ^2:

$$\xi^2 = q^2 + k^2 \qquad \Rightarrow \qquad \xi_{mn} = \pi\sqrt{m^2 + n^2} \tag{25.52}$$

The full space–time solution now takes the form

$$u_{mn} = [G_{mn} \cos c\xi t + H_{mn} \sin c\xi t] \sin mx \, \sin ny \tag{25.53}$$

where n and m are integers. Because the wave equation is linear in u, its most general solution is a linear combination of the eigenmodes (25.53):

$$u(x, y, t) = \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} [G_{mn} \cos c\xi t + H_{mn} \sin c\xi t] \sin mx \, \sin ny \tag{25.54}$$

While an infinite series is not generally a good algorithm, our choice of initial and boundary conditions means that only the $m = 2, n = 1$ term contributes:

$$u(x, y, t) = \cos c\sqrt{5} \, \sin 2x \, \sin y \tag{25.55}$$

where c is the wave velocity. You should verify that initial and boundary conditions are indeed satisfied.

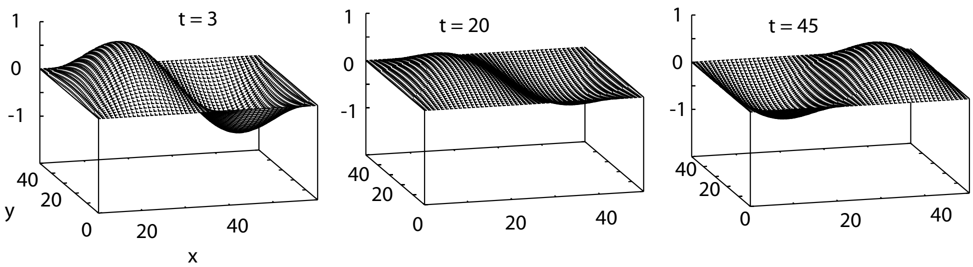

Fig. 25.11 The standing wave pattern on a square box top at three different times.

25.5 Numerical Solution for 2D Waves

The development of an algorithm for the solution of the 2D wave equation (25.43) follows that of the 1D equation in Section 25.1.2. We start by expressing the second derivatives in terms of central differences:

$$\frac{\partial^2 u(x,y,t)}{\partial t^2} = \frac{u(x,y,t+\Delta t) + u(x,y,t-\Delta t) - 2u(x,y,t)}{(\Delta t)^2} \tag{25.56}$$

$$\frac{\partial^2 u(x,y,t)}{\partial x^2} = \frac{u(x+\Delta x,y,t) + u(x-\Delta x,y,t) - 2u(x,y,t)}{(\Delta x)^2} \tag{25.57}$$

$$\frac{\partial^2 u(x,y,t)}{\partial y^2} = \frac{u(x,y+\Delta y,t) + u(x,y-\Delta y,t) - 2u(x,y,t)}{(\Delta y)^2} \tag{25.58}$$

After discretizing the variables, $u(x = i\Delta, y = i\Delta y, t = k\Delta t) \equiv u^k_{i,j}$, we obtain our time-stepping algorithm by solving for the future solution in terms of the present and past ones:

$$u^{k+1}_{i,j} = 2u^k_{i,j} - u^{k-1}_{i,j} c^2 \left(\frac{\Delta t}{\Delta x}\right)^2 \left[u^k_{i+1,j} + u^k_{i-1,j} - 4u^k_{i,j} + u^k_{i,j+1} + u^k_{i,j-1}\right] \quad (25.59)$$

Whereas the present (k) and past ($k-1$) solutions are known after the first step, to get the algorithm going we need to know the solution at $t = -\Delta t$, that is, before the initial time. To find that, we use the fact that the membrane is released from rest:

$$0 = \frac{\partial u(t=0)}{\partial t} \approx \frac{u^1_{i,j} - u^{-1}_{i,j}}{2\Delta t} \qquad \Rightarrow \qquad u^{-1}_{i,j} = u^1_{i,j} \quad (25.60)$$

After substitution into (25.59) and solving for u^1, we obtain the algorithm for the first step:

$$u^1_{i,j} = u^0_{i,j} + \frac{1}{2}c^2 \left(\frac{\Delta t}{\Delta x}\right)^2 \left[u^0_{i+1,j} + u^0_{i-1,j} - 4u^0_{i,j} + u^0_{i,j+1} + u^0_{i,j-1}\right] \quad (25.61)$$

Since the displacement $u^0_{i,j}$ is given at time $t = 0$ ($k = 0$), we compute the solution for the first time step with (25.61) and with (25.59) for subsequent steps.

Listing 25.2: Wave2D.java

```
// Wave2D.java: 2D Wave eq for vibrating membrane by Manuel J. Paez

  import java.io.*;

public class Wave2D {
  final static double den =390.0, ten =180.0;     // Density, T, step

  public static void main(String[] argv)
                   throws IOException, FileNotFoundException {
    int i, j, k;          // i,j: membrane grid positions, k: for time
    int max = 45;                          // Final time of oscillation
    double c, cprime, x, y;          // vel; cprime = delta u/delta t
    double covercp, incrx, incry;              // c, cprime, increments
    double u[][][] = new double[101][101][3];
    PrintWriter w =                          // Output in Helmholtz.dat
        new PrintWriter(new FileOutputStream("Helmholtz.dat"), true);
    double ratio;                                     // (c/cprime)^2.
    incrx = Math.PI/100.0;
    incry = Math.PI/100;
    c = Math.sqrt(ten/den);                        // Propagation speed
    cprime = c;                                         // For simplicity
    covercp = c/cprime;                                     // c / cprime
    ratio = 0.5*covercp*covercp;                   // 0.5 for stability
    System.out.println("ratio "+ratio);
    y = 0.0;
```

```
    for( j=0; j<101; j++ ) {              // Initial condition: position
      x = 0.0;
      for( i=0; i<101; i++ )  {
        u[i][j][0] = Math.sin(2.0*x)*Math.sin(y);
        x = x+incrx;
      }
      y = y+incry;
    }
    for ( j=1; j<100; j++ )  {                        // First time step
      for ( i=1; i<100; i++ )  {
        u[i][j][1] = u[i][j][0] +0.5*ratio*(u[i+1][j][0]+u[i-1][j][0]
            +u[i][j+1][0]+u[i][j-1][0]-4.0*u[i][j][0]) ;
      }
    }
    for ( k=1; k<=max; k++ )  {                          // Later times
      for ( j=1; j<100; j++ )  {
        for ( i=1; i<100; i++ )  {
          u[i][j][2] = 2.*u[i][j][1] - u[i][j][0]+ratio*(u[i+1][j][1]
          + u[i-1][j][1] + u[i][j+1][1]+u[i][j-1][1] - 4.*u[i][j][1]);
        }
      }
      for ( j=0; j<101; j++ ){
        for( i=0; i<101; i++ ){
          u[i][j][0] = u[i][j][1];                            // New past
          u[i][j][1] = u[i][j][2];                         // New present
        }
      }
      if ( k == max) {
        for ( j=0; j<101; j=j+2 ) {                  // Last time values
          for( i=0; i<101; i=i+2 ) {
            w.println("   " +u[i][j][2] );
          }   //for gnuplot
          w.println("");
        }
      }                                                            // if
    }                                                    // for k (time)
    System.out.println("data stored in Helmholtz.dat");
  }
}
```

The program `Wave2D.java` in Listing 25.2 solves the 2D wave equation using the time-stepping (leapfrog) algorithm. It continues iterating in time up to `max` steps. The shape of the membrane at three different times are shown in Fig. 25.11.