13

Unusual Dynamics of Nonlinear Systems

13.1 PROBLEM: VARIABILITY OF BUG POPULATIONS

The population of insects and the patterns of weather do not appear to follow any simple laws.[1] At times they appear stable, at other times they vary periodically, and at other times they appear chaotic, only to settle down to something simple again. Your **problem** is to deduce if a simple law might be producing such complicated behavior.

13.2 THEORY: NONLINEAR DYNAMICS

In many ways nonlinear dynamics is the glory of computational science. The computer helps us solve equations that are otherwise inaccessible, and because it is rather painless, it encourages exploration. And it works! The computed solutions have led to the discovery of new phenomena such as *solitons, chaos*, and *fractals*.

While volumes have been written on nonlinear dynamics, we will spend only two short chapters studying some simple systems. Nevertheless, you will uncover unusual properties on your own, and in the process cannot help but getting convinced that simple systems can have very complicated behaviors.

In some cases these complicated behaviors will be *chaotic*, but unless you

[1] Other than in Oregon, where storm clouds come to spend their weekends.

have a bug in your program, they will not be random.[2] We define *chaos* as the deterministic behavior of a system displaying no discernible regularity. This may seem contradictory; if a system is deterministic, it must have step-to-step correlations (which when added up means long-range correlations), but if it is chaotic, this means that it is too complicated to understand.

In an operational sense, a chaotic system is one with an extremely high sensitivity to parameters or initial conditions. This sensitivity to even minuscule changes is so high that, in practice, it is impossible to predict the long-range behavior unless the parameters are known to infinite precision (which they never are, in practice).

13.3 MODEL: NONLINEAR GROWTH, THE LOGISTIC MAP

Imagine a bunch of insects reproducing, generation after generation. We start with N_0 bugs, then in the next generation we have to live with N_1 of them, and after i generations there are N_i bugs to bug us. We want to model the time dependence of N_i, with time measured discretely in steps or generations.

For guidance, we look to the radioactive decay simulation in Chapter 7, *Monte Carlo Applications*, where the discrete decay law, $\Delta N/\Delta t = -\lambda N$, led to exponential decay when the numbers were large and the time steps short. So, if we assume that the bug breeding rate is proportional to the number of bugs:

$$\frac{\Delta N_i}{\Delta t} = \lambda' N_i, \tag{13.1}$$

we know that this will lead to unbounded exponential growth with a rate parameter λ'.

We can improve the model by realizing that bugs cannot live on love alone. They must also eat. But bugs, not being farmers, do not produce more food if they need it to sustain their growing numbers, and so the competition for a finite food supply tends to limit their number to a maximum N_*. We build this into our model by defining a new birth-rate parameter λ' that is proportional to the difference of the present and maximum populations:

$$\lambda' = \lambda(N_* - N_i), \tag{13.2}$$

$$\Rightarrow \quad \frac{\Delta N_i}{\Delta t} = \lambda'(N_* - N_i)N_i. \tag{13.3}$$

Physically, we expect that when their numbers are small, the bugs will grow exponentially, and as their numbers increase, we will see some type of moderation in the growth.

[2] You may recall from Chapter 6, *Deterministic Randomness*, that a random sequence of events does not even have step-by-step correlations.

13.3.1 The Logistic Map

Equation (13.3) is known mathematically as the *logistic map*. It is usually written in dimensionless form as an equation for the number of bugs in generation $i+1$:

$$N_{i+1} = N_i + \lambda'\Delta t(N_* - N_i)N_i, \tag{13.4}$$

$$= N_i(1 + \lambda'\Delta t N_*)\left[1 - \frac{\lambda'\Delta t}{1 + \lambda'\Delta t N_*}N_i\right]. \tag{13.5}$$

We define a dimensionless growth parameter μ and a dimensionless population variable x_i:

$$\mu \stackrel{\text{def}}{=} 1 + \lambda'\Delta t N_*, \tag{13.6}$$

$$x_i \stackrel{\text{def}}{=} \frac{\lambda'\Delta t}{\mu}N_i \simeq \frac{N_i}{N_*}. \tag{13.7}$$

Observe that the *growth rate* μ equals 1 when the breeding rate $\lambda' = 0$, and is otherwise expected to be larger than 1. If the number of bugs born per generation $\lambda'\Delta t$ is large, then $\mu \approx \lambda'\Delta t N_*$ and $x_i \approx N_i/N_*$. That is, x_i is essentially the fraction of the maximum population N_*. Consequently, we consider x values in the range

$$0 \le x_i \le 1, \tag{13.8}$$

where the value $x = 0$ corresponds to no bugs and $x = 1$ to the maximum population.

Dressed in these natural variables, the difference equation (13.5) assumes the standard form for the *logistic map*:

$$\boxed{x_{i+1} = \mu x_i(1 - x_i).} \tag{13.9}$$

In general, a map may use any function $f(x)$ to map one number in a sequence to the next:

$$x_{i+1} = f(x_i). \tag{13.10}$$

For the logistic map, $f(x) = \mu x(1-x)$. The quadratic dependence of f on x makes this a nonlinear map. The dependence of (13.9) on only the one variable x makes it a *one-dimensional* map.

We have developed a discrete model for the bug population and have expressed it as a *difference equation*. We will study this equation and see that its nonlinear dependence on x leads to some unusual behaviors. Similar behaviors are found in nonlinear differential equations, as we will see in Chapter 14, *Differential Chaos in Phase Space*; Chapter 28, *Solitons, the KdeV Equation*; and Chapter 29, *Sine-Gordon Solitons*.

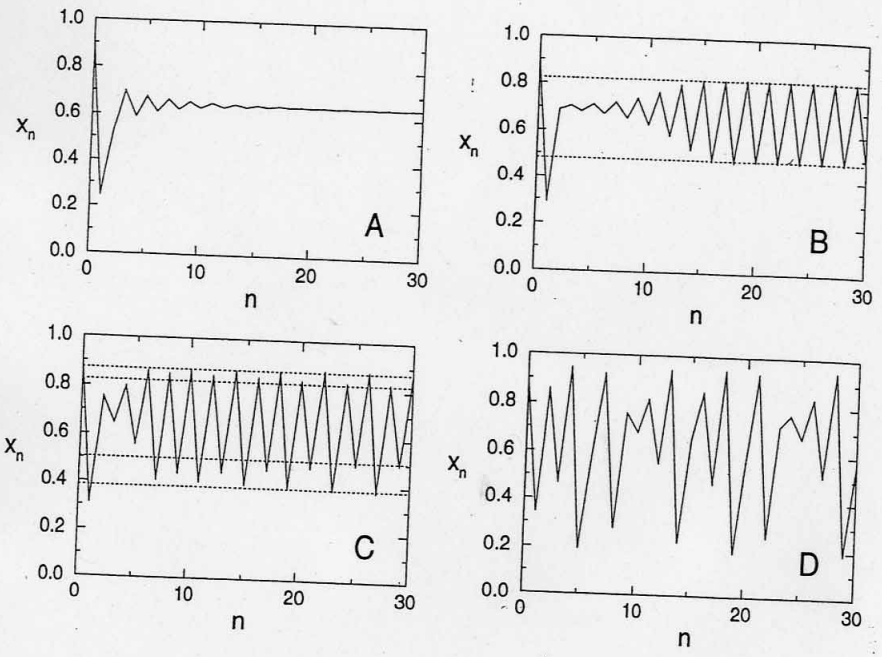


Fig. 13.1 The insect population x_n versus generation number n for various growth rates: (A) $\mu = 2.8$, a period-one cycle; (B) $\mu = 3.3$, a period-two cycle; (C) $\mu = 3.5$, a period-four cycle; (D) $\mu = 3.8$, a chaotic regime.

13.4 THEORY: PROPERTIES OF NONLINEAR MAPS

Some typical nonlinear behavior is shown in Fig. 13.1. The initial population x_0 is known as the *seed*, and as long as it is not equal to zero, its exact value generally has little effect on the population dynamics (similar to what we found when generating pseudo-random numbers). In contrast, the dynamics are controlled by the value of the growth parameter μ. For those values of μ at which the dynamics are chaotic, there is an extreme sensitivity to the initial condition, x_0, as well as on the exact value of μ.

13.4.1 Fixed Points

An important property of the map (13.9) is the possibility of a sequence of x_i values reaching a *fixed* point at which x_i remains or keeps returning to. We denote fixed points by x_*. At a one-cycle fixed point there is no change in the population from generation i to generation $i+1$, and so it must satisfy the equation

If we now use the logistic map (13.9) to relate x_{i+1} to x_i, we obtain an algebraic equation to solve

$$\mu x_*(1 - x_*) = x_* \tag{13.12}$$

$$\Rightarrow \quad x_* = 0, \text{ or } \quad x_* = \frac{\mu - 1}{\mu}. \tag{13.13}$$

The nonzero fixed point $x_* = (\mu - 1)/\mu$ corresponds to a stable population with an equilibrium between birth and death that is reached regardless of the initial population. In contrast, the $x_* = 0$ point is unstable and the population remains static only as long as no bugs exist; if even a few bugs are introduced, exponential growth occurs.

Further analysis [Rash 90] tells us that the stability of a population is determined by the magnitude of the derivative of the mapping function $f(x_i)$ at the fixed point:

$$\left|\frac{df}{dx}\right|_{x_*} < 1 \quad \text{(stable)}. \tag{13.14}$$

For the logistics map (13.9) this means that

$$\left.\frac{df}{dx}\right|_{x_*} = \mu - 2\mu x_* = \begin{cases} \mu, & \text{stable at } x_* = 0 \text{ if } \mu < 1, \\ 2 - \mu, & \text{stable at } x_* = \frac{\mu - 1}{\mu} \text{ if } \mu < 3. \end{cases} \tag{13.15}$$

13.4.2 Period Doubling, Attractors

Equation (13.15) tells us that while the equation for fixed points (13.13) may be satisfied for all values of μ, the points will not be stable if $\mu > 3$. For $\mu \geq 3$, the system's long-term population *bifurcates* into two populations (*period doubling*). Because the system then acts as if it were attracted now to two populations, these populations are called *attractors* or cycle points. We can easily predict the x values for these two-cycle attractors by requiring that generation $i + 2$ have the same population as generation i:

$$x_* = x_i = x_{i+2} = \mu x_{i+1}(1 - x_{i+1}), \tag{13.16}$$

$$\Rightarrow \quad x_* = \frac{1 + \mu \pm \sqrt{\mu^2 - 2\mu - 3}}{2\mu}. \tag{13.17}$$

We see that as long as $\mu > 3$, the square root produces a real number and thus physical solutions (complex or negative x_* values are unphysical).

We leave it for your computer explorations to discover how the system continues to double periods as the value of μ gets larger. In all cases the

13.5 IMPLEMENTATION AND ASSESSMENT: EXPLICIT MAPPING

Program the logistic map to produce a sequence of population values x_i as a function of the generation number i. These are called *map orbits*. The assessment consists of confirmation of Feigenbaum's observations [Feig 79] of the different behavior patterns shown in parts A, B, C, and D of Fig. 13.1. These occur for growth parameter $\mu = (0.4, 2.4, 3.2, 3.6, 3.8304)$ and seed population $x_0 = 0.75$. Notice the following on your graphs of x_i versus i:

1. **Transients:** Each map undergoes a *transient behavior* before reaching a steady state. These transients differ for different seeds.

2. **Asymptotes:** In some cases the steady state is reached after only 20 generations, while for larger μ values, hundreds of generations may be needed. These steady-state populations are independent of the seed.

3. **Extinction:** As shown in Fig. 13.1A, if the growth rate is too low, the population dies off.

4. **Stable states:** The single-population stable states attained for $\mu < 3$ agree with the prediction (13.13).

5. **Period doubling:** Examine the map orbits for a growth parameter μ increasing continuously through 3. Try to discover how the system continues to double periods as μ increases. For example, in Fig. 13.1C with $\mu = 3.5$, we notice a steady state in which the population alternates among four attractors (a four-cycle point).

6. **Intermittency:** Try to find solutions for $3.8264 < \mu < 3.8304$. Here the system appears stable for a while but then jumps all around, only to become stable again.

7. **Chaos:** The chaotic region is critically dependent on the value of μ, with the x_i values obtained critically dependent on x_0.

   (a) Verify this by running the logistics map with $\mu = 4$ and what should essentially be two identical seeds:

   $$x_0 = 0.75, \text{ and } x_0' = 0.75(1+\epsilon). \tag{13.18}$$

   Here ϵ should be a number very close to machine precision, for example; $\epsilon \simeq 2 \times 10^{-14}$ for double precision.

   (b) Now repeat the experiment with $x_0 = 0.75$ but with what should essentially be two identical growth parameters:

   $$\mu = 4.0, \text{ and } \mu' = 4.0(1-\epsilon). \tag{13.19}$$

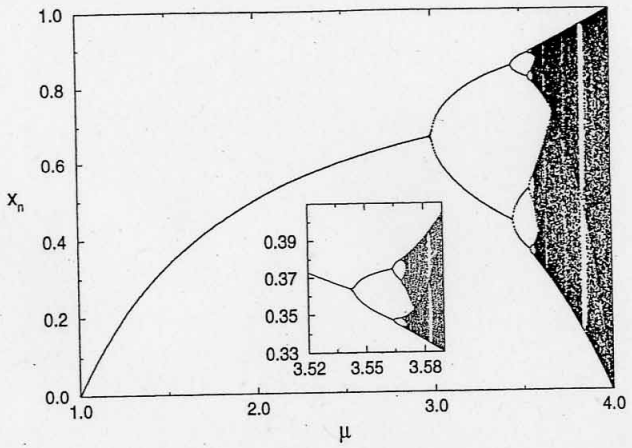


Fig. 13.2 The bifurcation plot, attractor populations versus growth rate, for the logistics map.

13.6 ASSESSMENT: BIFURCATION DIAGRAM

Computing and watching the population change with each generation gives a good idea of the basic phenomena, at least until the time dependence gets so complicated that it is hard to discern patterns. In particular, as the number of bifurcations keeps increasing and the system becomes chaotic, it is hard for our minds to see a simple underlying structure within the complicated behavior.

One way to visualize what is going on is to concentrate on the attractors; that is, those populations that appear to attract the solutions and to which the solutions continuously return. A plot of these attractors (long-term iterates) of the logistics map as a function of the growth parameter μ is an elegant way to summarize the results of extensive computer simulations.

One such *bifurcation diagram* is given in Fig. 13.2 (another is given in Fig. 13.3). For each value of μ, hundreds of iterations were made to make sure that all transients died out. Then the points (μ, x_*) were written to a file for hundreds of iterations after that. To be sure to reach all attractors, various values for the initial populations x_0 are stepped throughout.

13.7 IMPLEMENTATION: BIFURCATION DIAGRAM, BUGS.F (.C)

You should reproduce Fig. 13.2.[3] While the best way to make a visualization of this sort would be with a visualization program that permits you to vary the intensity of each individual point on the screen, we outline a method in which you control the density at each point in your plot by varying the number of pixels plotted.

Notice that your screen resolution may be ~100 dots per inch and your laser printer resolution ~300 dots per inch. This means that you need to plot ~3000 × 3000 ≃ 10 million elements. But *beware* for this can take some time to print and enough memory space on your hard disk and printer to choke them.

1. Break up the range $1 \le \mu \le 4$ into a 1000 steps and loop through them.
2. To be complete, loop through a range of x_0 values as well.
3. Wait at least 200 generations for the transients to die and then print out the next several hundred values of (μ, x_*) to a file.
4. Print out your x_* values to no more than 3–4 decimal places (you will not be able to resolve more places on your plot).
5. Sort[4] your (μ, x_*) file to remove any duplicate points (this may be a slow process, but it takes much less time than waiting for a choked printer).
6. Plot up your file of x_* versus μ. Use small symbols for the points and do not connect them.
7. Enlarge sections of your plot and notice that a similar bifurcation diagram tends to be contained within portions of the original (*self similarity*).
8. Notice the series of bifurcations at

$$\mu_k \simeq 3, 3.449, 3.544, 3.5644, 3.5688, 3.569692, 3.56989, \ldots. \tag{13.20}$$

   The end of this series is chaotic behavior.
9. Notice that after the sequence ends, others begin, only to end in chaos again. (The changes are rather quick, and plots with an enlarged μ scale are illuminating.)
10. Close examination of Fig. 13.2 shows regions containing very few populations (which are not artifacts of the video display). These are *windows*

[3] You can listen to a sonification of it on the Web.
[4] For example, with the Unix command *sort -u*.

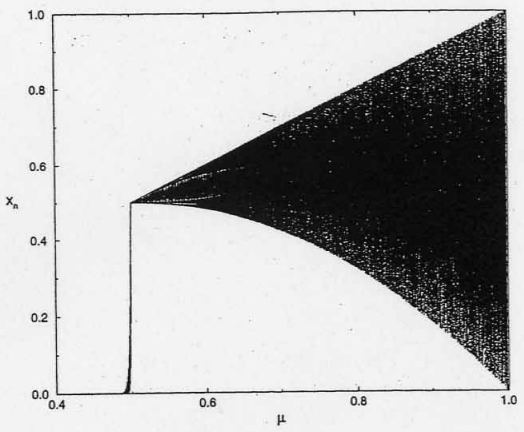


Fig. 13.3 A bifurcation plot for the tent map showing fixed points versus growth rate.

in which a slight increase in the growth rate μ changes a chaotic population into one with a finite number of fixed points. Check that at $\mu = 3.828427$ chaos turns into a three-cycle population.

13.8 EXPLORATION: RANDOM NUMBERS FROM LOGISTIC MAP?

There are claims [P&R 95] that the logistics map in the chaotic region

$$x_{i+1} = 4x_i(1 - x_i), \tag{13.21}$$

can be used to generate random numbers. While we know that successive bug populations are correlated, if the population for every ~ 6th generation is examined, the correlations die out and random numbers result. To help make the sequence uniform, a trigonometric transformation is used

$$y_i = \frac{1}{\pi} \cos^{-1}(1 - 2x_i). \tag{13.22}$$

Use the random-number tests discussed in Chapter 6, *Deterministic Randomness*, to confirm this claim.

13.9 EXPLORATION: FEIGENBAUM CONSTANTS

It was discovered by [Feig 79] that the sequence of μ_k values (13.20) converges geometrically when expressed in terms of the variable δ, which is the distance

Table 13.1 Several nonlinear maps to explore

Name	$f(x)$
Tent	$\mu(1-2\lvert x-1/2\rvert)$
Ecology	$xe^{\mu(1-x)}$
Quartic	$\mu[1-(2x-1)^4]$

between bifurcations:

$$\mu_k \;\to\; \mu_\infty - \frac{c}{\delta^k}, \tag{13.23}$$

$$\delta \;=\; \lim_{k\to\infty} \frac{\mu_k - \mu_{k-1}}{\mu_{k+1} - \mu_k}. \tag{13.24}$$

Use your sequence of μ_k values to determine the constants in (13.23) and compare to those found by Feigenbaum:

$$\mu_\infty \simeq 3.56995, \quad c \simeq 2.637, \quad \delta \simeq 4.6692. \tag{13.25}$$

Amazingly, the value of the *Feigenbaum constant* δ is a universal constant for all second order maps.

13.10 EXPLORATION: OTHER MAPS

Only nonlinear systems exhibit unusual behavior like chaos. Yet systems can be nonlinear in any number of ways. Table 13.1 lists three maps that you may use to generate x_i sequences and bifurcation plots. The tent map is illustrated in Fig. 13.3, which makes clear the origin of the name.

14

Differential Chaos in Phase Space

14.1 PROBLEM: A PENDULUM BECOMES CHAOTIC

In Fig. 14.1 we see a realistic pendulum *driven* through viscous air by an external, sinusoidal force. When the driving force is turned off, the system is observed to oscillate at the pendulum's natural frequency ω_0. When the driving force is strong, the system is observed to oscillate at the driver's frequency ω (*mode locking*). When ω_0 and ω are equal, or nearly equal, and the driving force is not too strong, a slow and periodic variation of the pendulum's amplitude is observed (*beating*). As the initial conditions are changed, the motion gets very complicated and appears chaotic. Your **problem** is to describe this behavior with a simple equation of motion, and to determine whether even the most complicated motion has some simple structure underlying it all.

In Chapter 13, *Unusual Dynamics of Nonlinear Systems*, we discovered that a simple nonlinear *difference* equation yields solutions that may be simple, complicated, or chaotic. In this chapter we search for similar behavior in *differential* equations. We also reveal the beauty and simplicity underlying chaotic systems by observing their flow in phase space.

Our study is based on the description given by [Rash 90]. An excellent, analytic discussion of the related *parametric oscillator* is given in [L&L 69, §25–30]. A similar system [G&T 96] has a vibrating pivot—in contrast to our periodic driving torque.

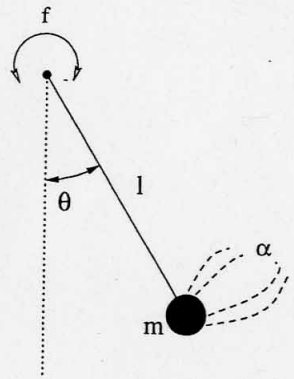


Fig. 14.1 A pendulum of length l driven through air by an external, sinusoidal torque. The strength of the torque is given by f and that of air resistance by α.

14.2 THEORY AND MODEL: THE CHAOTIC PENDULUM

The basic theory for the chaotic pendulum is the same as given in our discussion of the realistic pendulum in Chapter 11, *Anharmonic Oscillators*. Newton's laws of rotational dynamics tell us that the sum of the gravitational torque τ_g, the frictional torque τ_f, and the external torque τ_{ext} equals the moment of inertia of the pendulum times its angular acceleration:

$$\tau_g + \tau_f + \tau_{\text{ext}} = I\frac{d^2\theta}{dt^2}. \tag{14.1}$$

If we assume that the external torque is sinusoidal and that the viscous friction is proportional to the velocity of the pendulum's bob, we obtain the nonlinear differential equation

$$-\frac{mgl}{I}\sin\theta - \frac{\beta}{I}\frac{d\theta}{dt} + \frac{\tau_0}{I}\cos\omega t = \frac{d^2\theta}{dt^2}, \tag{14.2}$$

where l is the distance to the center of mass and the nonlinearity arises from the $\sin\theta$, as opposed to θ, dependence of the gravitational torque. In a more standard form, (14.2) is

$$\boxed{\frac{d^2\theta}{dt^2} = -\omega_0^2\sin\theta - \alpha\frac{d\theta}{dt} + f\cos\omega t,} \tag{14.3}$$

where

$$\omega_0 \stackrel{\text{def}}{=} \frac{mgl}{I}, \quad \alpha \stackrel{\text{def}}{=} \frac{\beta}{I}, \quad f \stackrel{\text{def}}{=} \frac{\tau_0}{I}. \tag{14.4}$$

In review, ω_0 is the natural frequency of the system arising from the gravitational restoring force, the α term arises from friction, and the f term arises from the driving force.

Equation (14.3) is a second order, time-dependent, nonlinear differential equation. In the standard form we use to solve differential equations,[1] it is the two simultaneous first-order equations:

$$y^{(1)} \stackrel{\text{def}}{=} \theta, \quad y^{(2)} \stackrel{\text{def}}{=} \frac{d\theta}{dt}, \tag{14.5}$$

$$\frac{dy^{(1)}}{dt}(t) = y^{(2)}(t), \tag{14.6}$$

$$\frac{dy^{(2)}}{dt}(t) = -\omega_0^2\sin y^{(1)}(t) - \alpha y^{(2)}(t) + f\cos\omega t. \tag{14.7}$$

14.3 THEORY: LIMIT CYCLES AND MODE LOCKING

It is easy to solve (14.6)–(14.7) on the computer. The difficulty is that the types of solutions that occur are so rich that it is not easy to figure out what's going on. Accordingly, we will examine some simple limiting cases where the motion is easy to understand.

When the chaotic pendulum is driven by a weak driving force (small value for f) with frequency ω equal to the natural frequency ω_0, it is possible to have a periodic steady-state motion with frequency of ω_0. In this case it is possible to pick the magnitude for the external torque such that after the initial transients die off, the average energy put into the system during one period exactly balances the average energy dissipated by friction during that period:

$$\langle f\cos\omega t\rangle = \langle\alpha\frac{d\theta}{dt}\rangle = \langle\alpha\frac{d\theta}{dt}(0)\cos\omega t\rangle, \tag{14.8}$$

$$\Rightarrow \quad f = \alpha\frac{d\theta}{dt}(0). \tag{14.9}$$

This leads to a *limit cycle* in which the motion is stable even in the presence of friction.

A somewhat opposite extreme occurs when the magnitude f of the driving torque is made much larger than that in (14.9). In this case the driving torque overpowers the natural oscillations of the pendulum and the steady-state motion is at the frequency of the driver. This is an example of *mode locking*. While mode locking can occur for linear or nonlinear systems, something unusual occurs for nonlinear systems. We have already seen how the oscillations

[1] See Chapter 9, *Differential Equations and Oscillations*.

of a nonlinear system may contain higher Fourier components (overtones). This means that under the right conditions, the driving force may lock onto the system by exciting its overtones. In this case the driving frequency and natural frequency are rationally related:

$$\frac{\omega}{\omega_0} = \frac{n}{m}, \tag{14.10}$$

where n and m are integers.

14.4 IMPLEMENTATION 1: SOLVE ODE, RK4.F (.C)

Take your solution to the harmonic or anharmonic oscillator and extend it to (14.6)–(14.7). This means adding velocity and time-dependent terms to the derivative function subroutine. Make α and f input parameters. Run some checks before you attack the full problem; namely, verify that

1. If $\alpha = f = 0$, you get the realistic pendulum studied previously.
2. If $f = 0$ but $\alpha \neq 0$, you get a uniformly decaying solution.

14.5 ASSESSMENT AND VISUALIZATION: PHASE-SPACE ORBITS

The conventional solution to an equation of motion is the position $x(t)$ and the velocity $v(t)$ as functions of time. In contrast, it is illuminating to go to an abstract space, *phase space*, where the ordinate is the velocity $v(t)$ and the abscissa is the position $x(t)$. As we can see in Figs. 14.2–14.6, the solutions of the equations of motion of classical dynamics form geometric objects in phase space. In this way, periodic motion, which is rather complicated to describe in terms of the time dependence of a position and a velocity, becomes recognizable geometric objects. Likewise, chaos, which seems beyond our descriptive abilities in ordinary space, becomes easily recognized structures in phase space.

Phase-space plots are useful in visualizing the solutions to the equations of motion. To understand why, we look at the nondriven one-dimensional harmonic oscillator with no friction. The position and velocity of the oscillating mass as functions of time are

$$x(t) = A\sin(\omega t), \tag{14.11}$$

$$v(t) = \frac{dx}{dt} = \omega A\cos(\omega t). \tag{14.12}$$

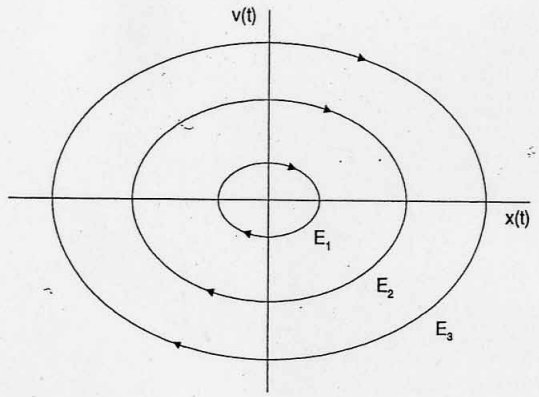


Fig. 14.2 A phase-space plot of velocity versus position for a harmonic oscillator. Because the ellipses close, the system must be periodic. The different ellipses correspond to different energies.

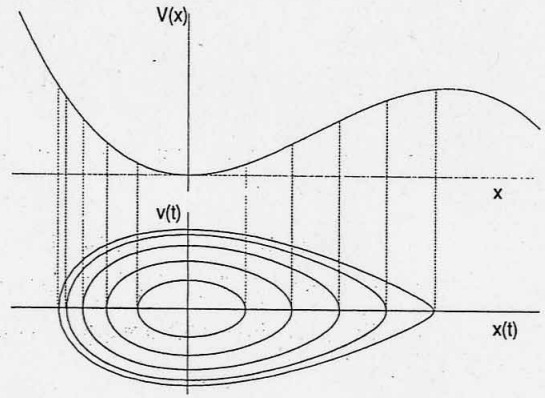


Fig. 14.3 (*Upper*) A potential-energy plot for a nonharmonic oscillator; (*lower*) a phase-space plot for the same nonharmonic oscillator. The ellipse-like figures are neither ellipses nor symmetric with respect to the v axis. The different orbits in phase space correspond to different energies.

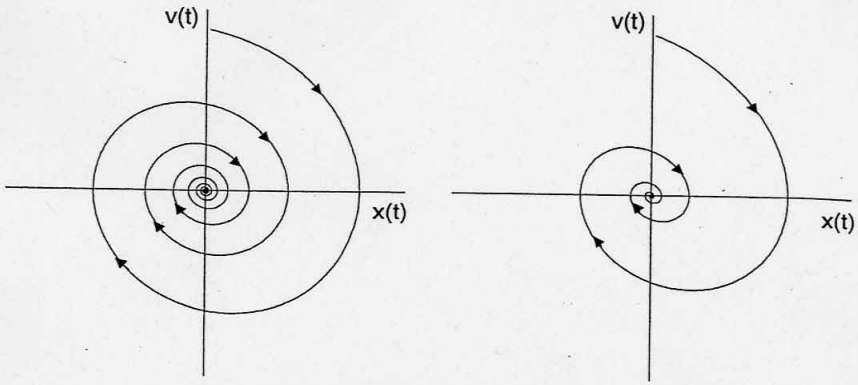


Fig. 14.4 Two phase-space orbits for oscillators with friction. The system eventually comes to rest at the origin. The damping is greater for the oscillator on the right.

When we use these solutions to form the total energy, we obtain a constant:

$$E = K(v) + V(x) = \tfrac{1}{2}mv^2 + \tfrac{1}{2}\omega^2 m^2 x^2 \tag{14.13}$$

$$= \frac{\omega^2 m^2 A^2}{2m}\cos^2(\omega t) + \tfrac{1}{2}\omega^2 m^2 A^2 \sin^2(\omega t) \tag{14.14}$$

$$= \tfrac{1}{2}m\omega^2 A^2. \tag{14.15}$$

Equation (14.13) implies that the harmonic oscillator follows closed elliptical orbits in phase space, with the size of the ellipse increasing with the system's energy. Different initial conditions produce the same ellipse if they have the same energy.

In Figs. 14.2–14.6 we show some typical structures encountered in phase space. Some generalities to note are

- For nonharmonic oscillators, the orbits will still be ellipse-like but with "corners" that become more distinct with increasing nonlinearity.
- Closed trajectories describe periodic oscillations [the same (x, v) occur again and again].
- The closed figures in the plots are stable *predictable attractors* or *stable attractors*. A nearby orbit in phase space is attracted to these (which means that they are stable).
- Phase-space orbits move clockwise for restoring forces (negative v after maximum x).
- Open orbits correspond to nonperiodic or "running" motion (a pendulum rotating like a propeller).

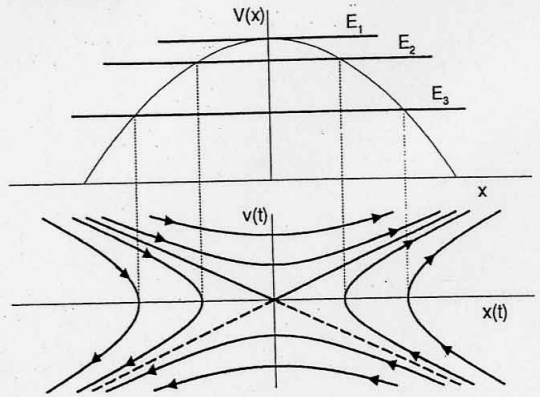


Fig. 14.5 Phase-space trajectories for a particle in a repulsive potential. Notice the absence of trajectories in the regions forbidden by energy conservation.

- Friction may cause the energy to decrease with time and the phase-space orbit to spiral into the a *fixed point*. However, for certain parameters, the energy pumped in by the external force exactly balances that lost by friction and a closed orbit results.
- For given initial conditions, different orbits do not cross because solutions are unique. Nonetheless, open orbits do come together at the points of unstable equilibrium (*hyperbolic points*).

14.6 IMPLEMENTATION 2: FREE OSCILLATIONS IN PHASE SPACE

You may recall the assessment in §11.10 in which you investigated phase-space orbits for anharmonic oscillations. That is the best place to start visualizing the chaotic pendulum.

1. Take your solution to the harmonic pendulum (θ, not $\sin\theta$, restoring torque) and plot the phase-space orbits for a variety of initial conditions:

   (a) $\theta(0) = 0$, $d\theta/dt(0) \neq 0$,

   (b) $\theta(0) \neq 0$, $d\theta/dt(0) = 0$,

   (c) $\theta(0) \neq 0$, $d\theta/dt(0) \neq 0$.

   Make sure that some of the initial conditions correspond to energies large enough to send the bob over the top ($E > 2ml$).

2. The phase-space flow should follow closed elliptic orbits. Determine whether the motion is clockwise or counterclockwise.

3. Plot the phase-space orbits for the same initial conditions but this time for the *realistic* pendulum ($\sin\theta$ not θ).

4. Deduce how the *shapes* of the orbits differ as the amplitude of oscillation becomes larger and larger.

5. Include friction into your model and describe the changes.

14.7 THEORY: CHAOTIC AND RANDOM MOTION IN PHASE SPACE

Random motion, when viewed in phase space, appears as a diffuse cloud filling all accessible regions of phase space. Periodic motion appears as smooth closed curves, ellipse-like in shape. As shown in Fig. 14.7, chaotic motion falls someplace in between. In particular, if viewed for long times and many initial conditions, the *flow* in phase space may contains dark *bands* rather than lines. The continuity within a band means that there is a continuum of solutions possible, and this is what causes the coordinate space solutions to appear chaotic. So even though the motion may be chaotic, the definite shape of the band means that there is a well-defined and simple structure within the chaos.

14.8 IMPLEMENTATION 3: CHAOTIC PENDULUM

The challenge with the computer study of the chaotic pendulum (14.6)–(14.7) is that the 4-D parameter space is immense. For normal behavior, sweeping through ω should show us resonances and beating; sweeping through α should show us underdamping, critical damping, and overdamping; sweeping through f should show us mode locking (at least for certain values of ω). These behaviors can be found here, yet they get mixed together.

Worse yet, when in the chaotic region, a minuscule change of a parameter or an initial condition may drastically change the solution. Accordingly, the exact locations of the characteristic regions in phase space are highly sensitive. For that reason, we are specific as to what parameters to use in the implementation to follow. You should not be surprised to require slight variations to obtain results similar to ours.

1. In this project you will reproduce the phase-space diagrams in Fig. 14.7. The different behaviors in this figure correspond to the different initial conditions (from top to bottom): $x(0) = -0.0885$, $v(0) = 0.8$; $x(0) = -0.0883$, $v(0) = 0.8$; $x(0) = -0.0888$, $v(0) = 0.8$. (The Web tutorials give animations showing an actual pendulum and sonifcations of these and other motions.)

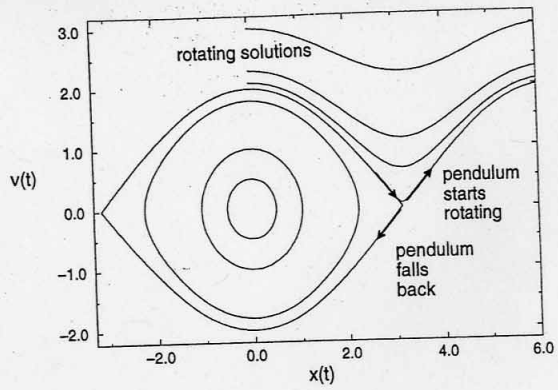


Fig. 14.6 Phase-space trajectories for a plane pendulum including "over the top" or rotating solutions. (Although not shown completely, the trajectories are symmetric with respect to vertical and horizontal reflections through the origin.)

   (a) For this first step, set $\alpha = 0.2$ and $\omega_0 = 1$.

   (b) To save time and storage, you may use a larger time step for plotting the orbits than used to solve the differential equations (a step size h equal to approximately $T/100$ is usually good). Plot $[\theta(t), d\theta/dt(t)]$ for ever-increasing time steps and see where your plots start losing detail.

   (c) Indicate which part of the orbits are transients.

   (d) Correlate phase-space structures with the behavior of $\theta(t)$ by plotting θ versus t on the same page as $d\theta/dt$ versus θ.

   (e) Gain some physical intuition about the flow in phase space by watching how it builds up with time.

2. For the second part of the study, use the same parameters as in first part, but now sweep through a range of ω values.

   (a) Use initial conditions: $d\theta/dt(0) = 0.8$, and $\theta(0) = -0.0888$.

   (b) For $\omega \simeq 0.6873$, you should find a period-three limit cycle where the pendulum jumps between three major orbits in phase space. (More precisely, there are three dominant Fourier components.)

   (c) For $\omega \simeq 0.694 - 0.695$, you should find running solutions where the pendulum keeps going over the top. Try to determine how many vibrations are made before the pendulum rotates.

   (d) For $\omega \simeq 0.686$, you should find chaotic motion in which the paths in phase space become bands of motion and the Fourier spectrum becomes broad (if you let the solution run long enough). Try to

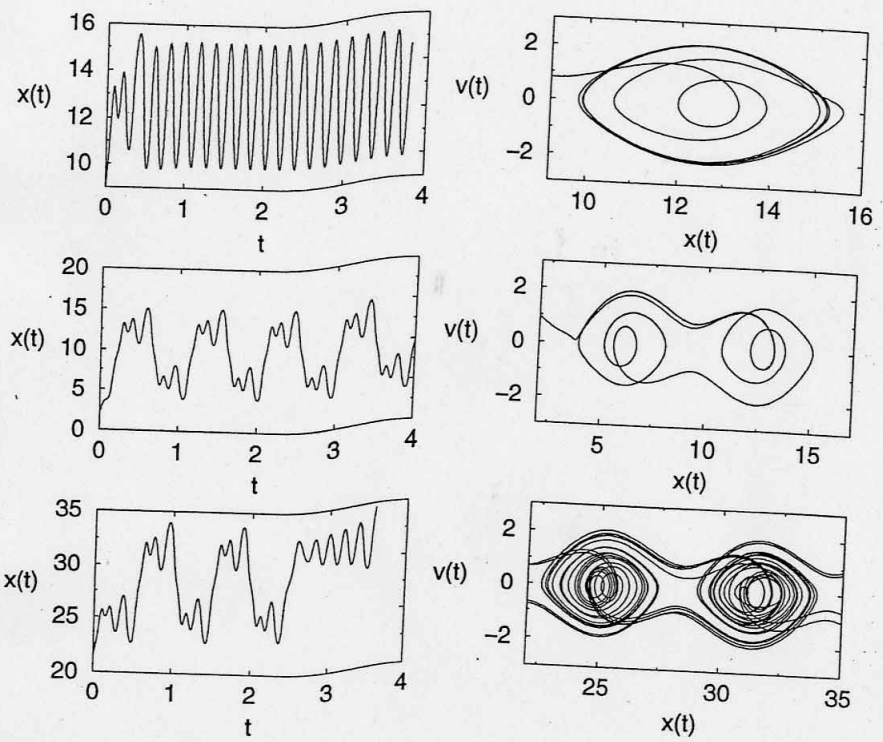


Fig. 14.7 A phase-space plot for the chaotic pendulum with $\omega_0 = 1$, $\alpha = 0.2$, $f = 0.52$, and $\omega = 0.666$. The angular position is $x(t)$ and the angular velocity is $v(t)$. The chaotic regions are the dark bands in the bottom figure.

determine just how small a difference in ω values separates the regular and the chaotic behaviors.

(e) Decrease your time step and try to determine how the bands get filled. Try to distinguish short-term and long-term behaviors in phase space.

14.9 ASSESSMENT: CHAOTIC STRUCTURE IN PHASE SPACE

Look at the plots you have produced and identify the following characteristic structures:

1. **Limit cycles:** ellipse-like figures with frequencies greater than ω_0. There may be sporadic changes among the limit cycles.

2. **Strange attractors:** well-defined, yet complicated semiperiodic behaviors that appear to be uncorrelated to the motion at an earlier time. These are highly sensitive to initial conditions. Even after millions of observations, the motion remains *attracted* to these paths.

3. **Predictable attractors:** well-defined, yet fairly simple periodic behaviors that are not particularly sensitive to initial conditions. These are orbits, such as fixed points and limit cycles, into which the system settles. If your location in phase space is near a predictable attractor, ensuing times will bring you to it.

4. **Chaotic paths:** regions of phase space that appear as filled-in bands rather than lines. The continuity within the bands implies some very complicated $\theta(t)$ behaviors, yet the general motion still has some underlying structure.

14.10 ASSESSMENT: FOURIER ANALYSIS OF CHAOTIC PENDULUM

We have seen that a realistic pendulum contains nonlinear terms in the restoring torque that lead to overtones; that is, frequencies other than just the fundamental. In addition, when a realistic pendulum is driven by an external sinusoidal force, the pendulum and driver may mode lock, and this leads to the pendulum moving at a frequency that is rationally related to the driver's. Consequently, the behavior of our chaotic pendulum is expected to be a combination of various periodic behaviors, sometimes occurring simultaneously and sometimes occurring sequentially.

In this assessment you will determine the Fourier components present in the pendulum's complicated and chaotic behaviors. This should show that a

"three-cycle structure," for example, contains three major Fourier components. You should also notice that when the pendulum goes over the top, its spectrum contains a steady-state ("dc") component.

1. Dust off your program which analyzes a $y(t)$ into Fourier components. (Alternatively, you may use a Fourier analysis tool contained in your graphics program or system library.)

2. Apply your analyzer to the solution of the forced, damped pendulum for the cases where there are one-, three-, and five-cycle structures in phase space. Deduce the major frequencies contained in these structures. (Try *not* to analyze the transient behavior.)

3. Try to deduce a relation between the Fourier components, the natural frequency ω_0, and the driving frequency ω.

4. A classic signal of chaos is a broadband, although not necessarily flat, Fourier spectrum. Examine your system for parameters that give chaotic behavior and verify this statement.

14.11 EXPLORATION: PENDULUM WITH VIBRATING PIVOT

A pendulum with a vibrating pivot point is an example of a *parametric resonance*. It is similar to our chaotic pendulum (14.3), but with the driving force depending on $\sin\theta$:

$$\frac{d^2\theta}{dt^2} = -\alpha\frac{d\theta}{dt} - (\omega_0^2 + f\cos\omega t)\sin\theta. \tag{14.16}$$

One way of understanding the physics of this equation is to go to the rest frame of the pivot (an accelerating reference frame) where you would say that there is a fictitious force that effectively leads to a sinusoidal variation of g or ω_0^2.

Analytic [L&L 69, §25-30] as well as numeric [DeJ 92, G&T 96] studies of this system exist. A fascinating aspect of this system is that the excitation of its modes of vibration (overtones) occurs through a series of *bifurcations*. In fact, when the instantaneous angular velocity $d\theta/dt$ is plotted as a function of the strength of the driving force, the bifurcation diagram in Fig. 14.8 results. Although the physics is very different, this behavior is manifestly similar to the bifurcation diagram for bug populations studied in Chapter 13, *Unusual Dynamics of Nonlinear Systems*. This behavior is, apparently, the result of mode locking and beating with the overtones.

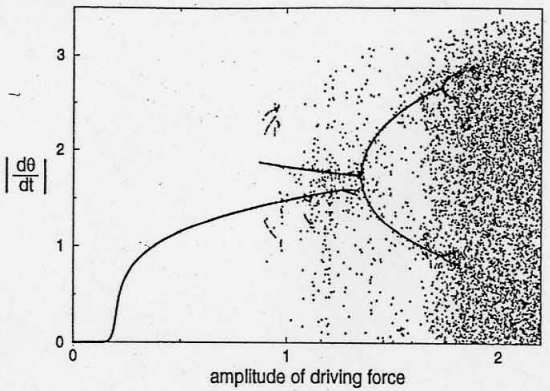


Fig. 14.8 Bifurcation diagram for the damped pendulum with a vibrating pivot. The ordinate is $|d\theta/dt|$, the absolute value of the instantaneous angular velocity at the beginning of the period of the driver, and the abscissa is the magnitude of the driving force d. The heavy line results from the overlapping of points, not from connecting the points. (Produced with the assistance of Melanie Johnson and Hans Kowallik.)

14.11.1 Implementation: Bifurcation Diagram of Pendulum

We obtained the bifurcation diagram of Fig. 14.8 by following these steps (a modification of those in [DeJ 92]):

1. Set $\alpha = 0.1, \omega_0 = 1, \omega = 2$, and let f vary through the range in Fig. 14.8.

2. Use the initial conditions: $\theta(0) = 1$ and $\frac{d\theta}{dt}(0) = 1$.

3. Sample (record) the instantaneous angular velocity $\frac{d\theta}{dt}$ whenever the driving force passes through zero.

4. Wait 150 periods before sampling to permit transients to die off.

5. Sample $\frac{d\theta}{dt}$ for 150 times and plot the results.

6. Plot $\left|\frac{d\theta}{dt}\right|$ versus f.

7. Repeat the procedure for each new value of f.

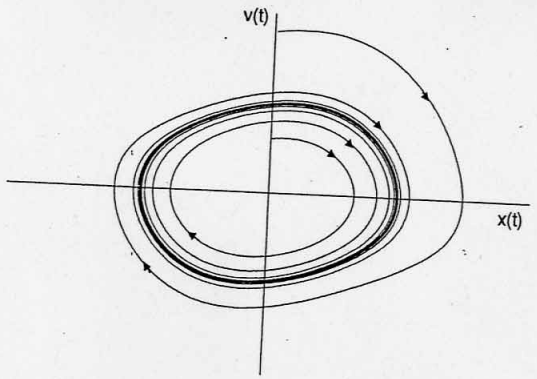


Fig. 14.9 Two phase-space trajectories corresponding to solutions of the van der Pol equation. One trajectory approaches the limit cycle (the dark curve) from the inside, while the other approaches it from the outside.

14.12 FURTHER EXPLORATIONS

1. The nonlinear behavior in once-common objects such as vacuum tubes and metronomes are described by the **van der Pool equation**

$$\frac{d^2x}{dt^2} + \mu(x^2 - x_0^2)\frac{dx}{dt} + \omega_0^2 x = 0. \tag{14.17}$$

The behavior predicted for these systems is *self-limiting* because the equation contains a limit cycle that is also a predictable attractor. You can think of (14.17) as describing an oscillator with x-dependent damping (the μ term). If $x > x_0$, friction slows the system down; if $x < x_0$, friction speeds the system up. Some phase-space orbits are shown in Fig. 14.9. The heavy curve is the *limit cycle*. Orbits internal to the limit cycle spiral out until they reach the limit cycle; orbit external to it spiral in.

2. **Duffing oscillator:** Here we have a nonlinear oscillator that is damped and driven,

$$\frac{d^2\theta}{dt^2} - \tfrac{1}{2}\theta(1-\theta^2) = -\alpha\frac{d\theta}{dt} + f\cos\omega t. \tag{14.18}$$

This is similar to the chaotic pendulum we studied, but has some advantage in the ease with which the multiple attractor sets can be found. It has been studied by [M&L 85].

3. **Lorenz attractor:** In 1962 Lorenz was looking for a simple model for weather prediction and simplified the heat-transport equations to the three equations [Tab 89]:

$$\frac{dx}{dt} = 10(y-x), \tag{14.19}$$

$$\frac{dy}{dt} = -xz + 28x - y, \tag{14.20}$$

$$\frac{dz}{dt} = xy - \tfrac{8}{3}z. \tag{14.21}$$

The solution of these simple nonlinear equations gave the complicated behavior that has led to the modern interest in chaos (after considerable doubt regarding the numerical solutions).

4. **A 3-D computer fly:** Plot, in 3-D space, the equations

$$x = \sin ay - z\cos bx, \tag{14.22}$$

$$y = z\sin cx - \cos dy, \tag{14.23}$$

$$z = e\sin x. \tag{14.24}$$

Here the parameter e controls the degree of randomness.

5. **Hénon–Heiles potential:** The potential and Hamiltonian

$$V(x,y) = \tfrac{1}{2}x^2 + \tfrac{1}{2}y^2 + x^2y - \tfrac{1}{3}y^3, \tag{14.25}$$

$$H = \tfrac{1}{2}p_x^2 + \tfrac{1}{2}p_y^2 + V(x,y), \tag{14.26}$$

are used to describe three astronomical objects interacting. They bind the objects near the origin, but releases them if they move far out. The equations of motion for this problem follow from the Hamiltonian equations

$$\frac{dp_x}{dt} = -x - 2xy, \qquad \frac{dp_y}{dt} = -y - x^2 + y^2, \tag{14.27}$$

$$\text{where } \frac{dx}{dt} = p_x, \qquad \frac{dy}{dt} = p_y. \tag{14.28}$$

(a) Numerically solve for the position $[x(t), y(t)]$ for a particle in the Hénon–Heiles potential.

(b) Plot $[x(t), y(t)]$ for a number of initial conditions. Check that the initial condition $E < \frac{1}{6}$ leads to a bounded orbit.

(c) Produce a Poincare section in the (y, p_y) plane. Plot (y, p_y) each time an orbit passes through $x = 0$.

22

Thermodynamic Simulations: The Ising Model

22.1 PROBLEM: HOT MAGNETS

Ferromagnetic materials such as bar magnets contain *domains* that are magnetized even in the absence of an external magnetic field. When an external magnetic field is applied, the different domains align and the internal fields become very high. Yet, as the temperature of the ferromagnet is raised, the magnetism decreases and in some cases a *phase transition* occurs in which the magnetism decreases precipitously. Your **problem** is to explain the thermal behavior and the phase transitions of ferromagnets.

We will solve this problem with the quantum mechanical Ising model and the simulated annealing (Metropolis) algorithm. The model is simple but contains much physics, and the simulation gives a visualization of thermal equilibrium that is absent from formal studies. The same algorithm and theory is used for lattice quantum mechanics, and so the present project should be completed before attempting Chapter 23, *Functional Integration on Quantum Paths*.

22.2 THEORY: STATISTICAL MECHANICS

When we say that an object is at a temperature T, we mean that the atoms composing this object are in a state of thermodynamic equilibrium with an environment at temperature T. While this may be an equilibrium state, it is also a dynamic state (it is thermo*dynamics* after all). The system's

energy is continually changing as it exchanges energy with the environment. If the system is at temperature T, then its atoms have an average kinetic energy proportional to T, with larger and larger fluctuations from this average occurring as the temperature increases.

An example of how the equilibrium state changes with temperature is the annealing process (it's one we will simulate on the computer). Let's say that we are making a blade for a sword and are hammering away at it while it's red hot to get its shape just right. At this high a temperature there is a lot of internal motion and not the long-range order needed for a stiff blade. So, as part of the process, we *anneal* the blade; that is, we heat and slow cool it in order to reduce brittleness and increase strength. Too rapid a cooling would not permit long-range equilibration (and ordering) of the blade, and this would lead to brittleness.

In the present problem we deal with the thermal properties of magnetized materials. The magnetism arises from the alignment of the quantum mechanical spins of the atoms. The spin of each atom, in turn, arises from its electrons. When the number of electrons is large, the problem is too difficult to really solve, and so statistical methods are used to obtain average quantities (in most cases that is all we can measure, anyway). If the system is described microscopically by classical or quantum mechanics, then this method is called *statistical mechanics.*

Statistical mechanics starts with the elementary interactions among a system's particles and constructs the macroscopic thermodynamic properties such as temperature T and internal energy U. The essential assumption is that all microscopic configurations of the system consistent with the constraints are equally probable. This leads to a distribution for the states of the system in which state α_j with energy $E(\alpha_j)$ occurs with a probability given by the Boltzmann factor:

$$P(\alpha_j) = \frac{e^{-E(\alpha_j)/kT}}{Z(T)}, \tag{22.1}$$

$$Z(T) = \sum_{\alpha_j} e^{-E_j/kT}. \tag{22.2}$$

Here k is Boltzmann's constant, T is the temperature, and the partition function $Z(T)$ is a weighted sum over states.

Notice that the Boltzmann distribution (22.1) does not require a thermal system to be in the state of lowest energy. Rather, it states that it is less likely for the system to have a high energy. Of course, as $T \to 0$ only the $E = 0$ state has a nonvanishing probability, yet for finite temperatures we expect the system's energy to have fluctuation on the order of kT.

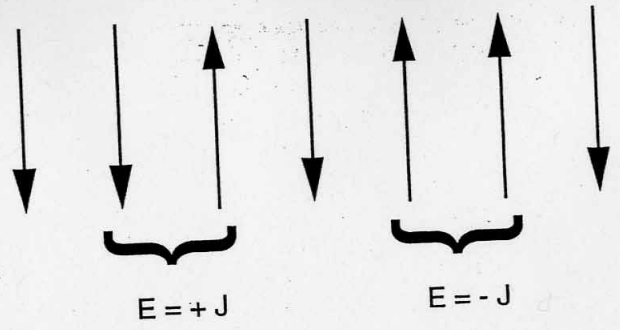


Fig. 22.1 A 1-D lattice of N spins. The interaction energy $V = \pm J$ between nearest-neighbor pairs is shown for aligned and opposing spins.

22.3 MODEL: AN ISING CHAIN

As our model, we consider N magnetic dipoles fixed on the links of a linear chain as shown in Fig. 22.1. (It is a straightforward generalization to handle two- and three-dimensional lattices, and indeed we do it as an exploration.) Because the particle are fixed, their positions and momenta are not dynamical variables, and we need to only worry about their spins.

We assume that the particle at site i has spin s_i, which can be either up or down:

$$s_i \equiv s_{z,i} = \pm \tfrac{1}{2}. \tag{22.3}$$

Each possible configuration or state of the N particles is described by the quantum state vector

$$|\alpha_j\rangle = |s_1, s_2, \ldots, s_N\rangle \tag{22.4}$$

$$= \{\pm \tfrac{1}{2}, \pm \tfrac{1}{2}, \ldots\}, \quad j = 1, 2^N. \tag{22.5}$$

Since the spin of each particle can assume that any one of *two* values, there are 2^N different possible states of the N particles in the system. We do not worry about the effect on the state vector of interchanging identical particles because they are all fixed in place and so can't be confused with each other.

The energy of the system arises from the interaction of the spins with each other and with the external magnetic field B. We know from quantum mechanics that an electron's spin and magnetic moment are proportional to each other, so a "dipole–dipole" interaction is equivalent to a "spin–spin" interaction. We assume that each dipole interacts with the external magnetic field and with its nearest neighbor through the potential:

$$V_i = -J\mathbf{s}_i \cdot \mathbf{s}_{i+1} - g\mu \mathbf{s}_i \cdot \mathbf{B}. \tag{22.6}$$

Here the constant J is called the *exchange energy* and is a measure of the strength of the spin–spin interaction. The constant g is the gyromagnetic ratio; that is, the proportionality constant between the angular momentum and magnetic moment. The constant μ is the Bohr magneton, the unit for magnetic moments.

In our model with N particles, there are 2^N possible configurations. Since $2^{20} > 10^6$, we see that even a small number of particles can lead to a prohibitively large number of configurations to consider. For a small number of particles, the computer can examine all possible spin configurations, yet as N gets larger, or as we go to two and three dimensions, a less exact statistical approach is used. We will consequently apply equations (22.1)–(22.2). In addition, we know from our previous studies that Monte Carlo techniques work well on the computer, and so we expect that the simulation may be realistic. Just how large N must be for this to occur depends somewhat on how good a description is needed; for our purposes $N > 200$ should appear statistical, with $N \simeq 2000$ being reliable.

The energy of the system to be used in the Boltzmann distribution (22.1) is the expectation value of the sum of V over the spins of the particles:

$$E(\alpha) = \langle\alpha| \sum_i V_i |\alpha\rangle \tag{22.7}$$

$$= -J \sum_{i=1}^{N-1} s_i s_{i+1} - B\mu \sum_{i=1}^{N} s_i. \tag{22.8}$$

For simplicity of notation, and to be able to compare with an analytic result, we now turn off the magnetic field, that is, assume that $B = 0$. Nonetheless, it is easy to add it back into the final expression for the energy, and we recommend that you do it because giving a preferred direction to space stabilizes the configuration.

The equilibrium alignment of the spins depends critically on the sign of the exchange energy J. If $J > 0$, the lowest energy state will tend to have neighboring spins aligned. If the temperature is low enough, the ground state will be a *ferromagnet* with essentially all spins aligned. If $J < 0$, the lowest energy state will tend to have neighbors with opposite spins. If the temperature is low enough, the ground state will be a *antiferromagnet* with alternating spins.

Unfortunately, a simple model such as this has its limits. Its approach to thermal equilibrium is qualitatively but not quantitatively correct. Further, when $B = 0$ there is no preferred direction in space, which means that the average magnetization may vanish, and this may lead to instabilities in which the spins spontaneously reverse. These instabilities are a type of Bloch-wall transitions in which regions of different spin orientations change size. In addition, the phase transition at the Curie temperature, a characteristic of magnetic materials, does not occur in the 1-D version of the model.

As you will verify in your simulation, a system described by the Boltzmann distribution (22.1) does not have a single configuration. Rather, there is a continual and random interchange of thermal energy with the environment that leads to fluctuations in the total energy. Even at equilibrium, the system fluctuates with the fluctuations getting larger as the temperature rises.

22.4 SOLUTION, ANALYTIC

For very large numbers of particles, the thermodynamic properties of the 1-D Ising model can be calculated analytically [P&B 89]. This tells us that the average energy (in J units) is

$$\frac{U}{J} = -N \tanh \frac{J}{kT} = -N \frac{e^{J/kT} - e^{-J/kT}}{e^{J/kT} + e^{-J/kT}}, \tag{22.9}$$

$$\Rightarrow \begin{cases} N, & \text{for } kT \to 0, \\ 0, & \text{for } kT \to \infty. \end{cases} \tag{22.10}$$

The analytic result for the specific heat per particle is

$$C(kT) = \frac{1}{N}\frac{dU}{dT} = \frac{(J/kT)^2}{\cosh^2(J/kT)}. \tag{22.11}$$

While the specific heat for the 1-D model does have a maximum as a function of temperature, it does not exhibit the characteristic discontinuity of a phase transition (the 2-D and 3-D models do). For the 1-D Ising model, the magnetization is

$$M(kT) = \frac{N e^{J/kT} \sinh(B/kT)}{\sqrt{e^{2J/kT} \sinh^2(B/kT) + e^{-2J/kT}}}. \tag{22.12}$$

22.5 SOLUTION, NUMERICAL: THE METROPOLIS ALGORITHM

We need an algorithm to evaluate the sums that appear in the partition function (22.2). This is analogous to a 2^N-dimensional numerical integration, and we know from §7.13 that a Monte Carlo approach is best for such high dimensions. Yet we do not want to generate random configurations uniformly for a system in thermal equilibrium because the Boltzmann factor essentially vanishes for those configurations whose energies are not close to the minimum energy. In other words, the vast majority of the terms we sum over to determine the thermodynamic properties of the system make hardly any contribution, and it would be quicker to weight the random numbers we generate such that most of the sum is over large terms.

In their simulation of neutron transmission through matter, Metropolis,

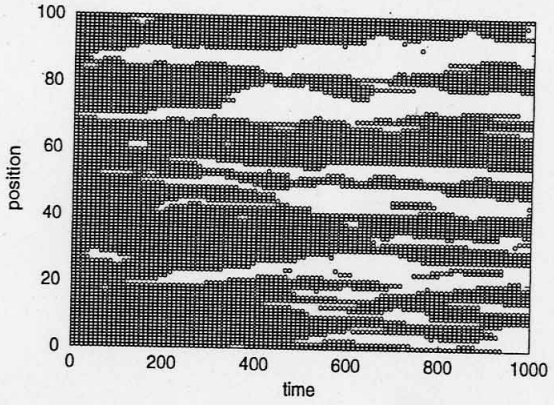


Fig. 22.2 A 1-D lattice of 100 spins. As indicated by the dark circles, initially all the spins are pointing up (a "cold" start). Magnetic domains are seen to form as the system heats up.

Rosenbluth, Teller, and Teller [Metp 53] found an algorithm to improve the Monte Carlo calculation of averages. This *Metropolis algorithm* has now become a cornerstone of computational physics. The sequence of configurations it produces is an example of a *Markov chain*, and in our case simulates the fluctuations in spin configurations that occur during thermal equilibrium. Although not simple or particularly illuminating to prove, this algorithm changes randomly the individual spins in such a way that on the average, the probability of any one configuration occurring follows a Boltzmann distribution.

The Metropolis algorithm involves two steps. First we start off with an arbitrary initial condition and repeatedly apply the algorithm until thermal equilibrium for a specific temperature is reached. For example, in Fig. 22.2 we have an initially "cold" system, whereas the spins would be random for have an initially "hot" system. Once the system reaches thermal equilibrium, the algorithm generates the statistical fluctuations about equilibrium that determine the thermodynamic quantities.

Because the 2^N configurations of N spins can be a lot, the amount of computer time needed can be very long (yet the program is simple). The hope is that the configurations obtained after a small number ($\simeq 10N$) of iterations will be close to those that produce minimum energy. While the answers do get better if we sample more and more configurations, there is a limit to improvement because roundoff error increases as more calculations are made.

Explicitly, the Metropolis algorithm is used to generate a nonuniform, random distribution of spin configurations α_j values, (22.5), with each α_j having

probability

$$\mathcal{P}(\alpha_j) = \frac{1}{Z} e^{-E(\alpha_j)/kT}. \tag{22.13}$$

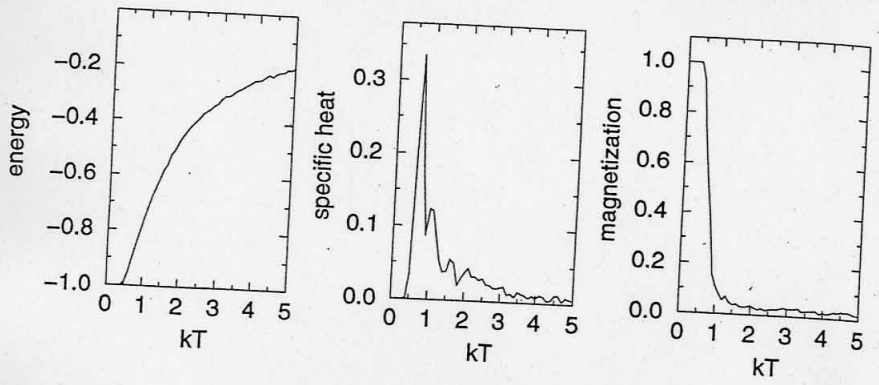
The technique is a variation of von Neumann rejection (stone throwing of §7.8) in which we start with an initial configuration and vary it randomly to obtain a *trial* configuration. The Boltzmann factor tells us that the relative probability of this trial configuration is proportional to $\Delta\mathcal{P} = \exp(-\Delta E/kT)$, where ΔE is the difference in energy between the previous and the trial configuration. If the trial configuration has a lower energy, ΔE will be negative, the relative probability will be greater than one, and we accept the trial configuration as the new configuration with no further ado. If, on the other hand, the trial configuration has a higher energy, we do not reject it out of hand, but accept it with the relative probability $\Delta\mathcal{P} = \exp(-\Delta E/kT) < 1$.

To accept a configuration with a probability, we pick a uniform random number between 0 and 1, and if the relative probability is greater than this number, we accept the trial configuration; if the Boltzmann factor is smaller than the chosen random number, we reject it. When the trial configuration is not accepted, the next configuration is identical to the preceding one.

The key aspects of the Metropolis algorithm is that the weight given to a trial configuration depends on how far it is from the minimum-energy configuration. Those configurations that stray far from the minimum-energy configuration are deemphasized but not completely discarded. By permitting the system to deviate away from the minimum-energy configuration (go "uphill" for a while), this technique is successful at finding a global extremum for situations in which other techniques are successful at finding only local ones. Its success relies on it not being too quick in "cooling" to the minimum-energy configuration; for this reason the algorithm is sometimes called *simulated annealing*. The algorithm is expected to be good at simulating the fluctuations about minimum energy, and gives explicit results for the thermodynamic quantities like those in Fig. 22.3.

The explicit **rules** for the Metropolis algorithm are

1. Start with an arbitrary spin configuration $\alpha_k = \{s_1, s_2, \ldots, s_N\}$.
2. To generate a new configuration α_{k+1}:
   (a) Pick particle i randomly.
   (b) Reverse i's spin direction to create trial configuration α_{tr}.
   (c) Calculate the energy $E(\alpha_{\text{tr}})$ of the trial configuration.
   (d) If $E(\alpha_{\text{tr}}) \le E(\alpha_k)$, accept the trial; that is, set $\alpha_{k+1} = \alpha_{\text{tr}}$.
   (e) If $E(\alpha_{\text{tr}}) > E(\alpha_k)$, accept with probability $\mathcal{P} = \exp(-\Delta E/kT)$:
      i. Choose a uniform random number: $0 \le r \le 1$,
      ii. Let $\alpha_{k+1} = \begin{cases} \alpha_{\text{tr}}, & \text{if } \mathcal{P} \ge r \quad \text{(accept)}, \\ \alpha_k, & \text{if } \mathcal{P} < r \quad \text{(reject)}. \end{cases}$



g. 22.3 Simulated results for the energy, specific heat, and magnetization of a 1-D tice of 100 spins as a function of temperature.

How do you start? One possibility, clearly a good choice for high temratures, is to start with random values of the spins. Another possibility, arly a good choice for low temperatures, is to start with a ferromagnetic or iferromagnetic configuration (for positive and negative J, respectively). In ieral, one tries to remove the importance of the starting configuration by ing the calculation "run a while" ($\simeq N$ rearrangements) before calculating rages. Then you repeat the calculation for different starting configurations, l take the average of the results.

5 IMPLEMENTATION, METROPOLIS ALGORITHM, ISING.F (.C)

Write a program that implements the Metropolis algorithm, that is, that produces a new configuration α_{k+1} from the present configuration α_k. (Alternatively, use the program supplied on the diskette or the Web.)

Make the key data structure in your program an array `s(1:N)` containing the values of s_i. For debugging, print out + and - to give the spins at each lattice point and the trial number.

The value for the exchange energy J fixes the scale for energy. Keep it fixed at $J = 1$ (or -1 for an antiferromagnet).

The thermal energy kT in units of J is the independent variable that your program should treat as an input parameter. Use $kT = 1$ for debugging.

Use periodic boundary conditions on your chain to minimize end effects. (This means that make the first and last spins the same.)

6. Try $N \simeq 20$ for debugging and larger values for production runs.
7. Use the printout to check that the system equilibrates for
   (a) A totally ordered initial configuration (cold start).
   (b) A random initial configuration (hot start).

Your cold start simulation should resemble Fig.22.2.

22.7 ASSESSMENT: APPROACH TO THERMAL EQUILIBRIUM

1. Watch a chain of N atoms attain thermal equilibrium when in contact with a heat bath. At high temperatures, or for small numbers of atoms, you should see large fluctuations, while at lower temperatures you should see smaller fluctuations.
2. The largest kT may be unstable as the system can absorb enough energy to flip all its spin. This is related to the fact that we have eliminated the magnetic field and in this way have no preferred direction to space. Introducing an external magnetic field B will stabilize the system but will also change the total energy and the analytic results.
3. Note how at thermal "equilibrium" the system is still quite dynamic with spins flipping all the time. It is the energy exchange that determines the thermodynamic properties.
4. You may well find that simulations at small kT (say that $kT \sim 0.1$ for $N = 200$) are slow to equilibrate. Higher kT values equilibrate faster yet have larger fluctuations.

22.8 ASSESSMENT: THERMODYNAMIC PROPERTIES

For a given spin configuration α_j, the energy and magnetization are given by

$$E_j = -J \sum_{i=1}^{N-1} s_i s_{i+1}, \tag{22.14}$$

$$\mathcal{M}_j = \sum_{i=1}^{N} s_i. \tag{22.15}$$

At high temperatures we expect a random assortment of spins and so a vanishing magnetization. At low temperature we expect $\mathcal{M}$ to approach $N/2$ as all the spins get aligned.

/hile the specific heat can be computed from the elementary definition

$$C = \frac{1}{n}\frac{dU}{dT}, \tag{22.16}$$

g a numerical differentiation of a fluctuating variable is not expected to be rate. A better way is to first calculate the fluctuations in energy occurring ıg a number of simulations

$$U_2 = \frac{1}{M}\sum_{t=1}^{M}(E_t)^2, \tag{22.17}$$

:hen determine the specific heat from the energy fluctuations:

$$C = \frac{U_2 - (U)^2}{kT^2}. \tag{22.18}$$

Extend your program to calculate the total internal energy U (22.14) and the magnetization $\mathcal{M}$ (22.15) for the chain. Notice that you do not have to recalculate entire sums for each new configuration because only one spin changes.

Make sure you wait for your system to attain thermal equilibrium before you calculate thermodynamic quantities. (You can check that U is fluctuating about its average.) Your results should resemble those shown in Fig.22.3.

The large statistical fluctuations are reduced by running the simulation a number of times with different seeds and taking the average of the :esults.

The simulations you run for small N may be realistic but may not agree vith statistical mechanics, which assumes $N \simeq \infty$ (you may assume that V $\simeq$ 2000 is close to infinity). Check that agreement with the analytic esults for the thermodynamic limit is better for large N than small N.

Check that the simulated thermodynamic quantities are independent f initial conditions (within statistical uncertainties). In practice, your old and hot start results should agree.

Iake a plot of the internal energy U as a function of kT and compare) the analytic result (22.9).

Iake a plot of the magnetization $\mathcal{M}$ as a function of kT and compare) the analytic result. Does this agree with how you expect a heated ıagnet to behave?

ompute the fluctuations of the energy U_2 (22.17), and the specific heat (22.18). Make a graph of your simulated specific heat compared to ıe analytic result (22.11).

22.9 EXPLORATION: BEYOND NEAREST NEIGHBORS

Extend the model so that the spin–spin interaction (22.6) extends to next-nearest neighbors as well as nearest neighbors. For the ferromagnetic case, this should lead to less fluctuation because we have increased the couplings among spins and thus an increased thermal inertia.

22.10 EXPLORATION: 2-D AND 3-D ISING MODELS

Extend the model so that the ferromagnetic spin–spin interaction (22.6) extends to nearest neighbors in two and then three dimensions. Continue using periodic boundary conditions and keep the number of particles small, at least to start [G&T 96].

1. Form a square lattice and place $\sqrt{N}$ spins on each side.
2. Examine the mean energy and magnetization of the system as it equilibrates.
3. Is the temperature dependence of the average energy qualitatively different from the 1-D model?
4. Print out the spin configurations for small N ($\simeq$ 16–25) and identify the domains.
5. Once your system appears to be behaving properly, calculate the heat capacity of the 2-D Ising model with the same technique used for the 1-D model. Use a total number of particles $100 \leq N \leq 1000$.
6. Look for a phase transition from an ordered to unordered configuration by examining the heat capacity as a function of temperature. It should diverge at the phase transition (you may get only a peak).

25

Electrostatic Potentials

25.1 INTRODUCTION: TYPES OF PDE'S

Natural quantities such as temperature and pressure vary continuously in both space and time. Such being our world, the function or *field* $U = U(x, y, z, t)$ used to describe these quantities must contain space and time coordinates as independent variables. The independence of each variable means that the derivatives in the equations must be partial derivatives. The equations are then *partial differential equations* (PDEs) in contrast to ordinary differential equations (ODEs).

Solving PDEs differs from solving ODEs in a number of ways. In particular, the *initial condition* (the $t = 0$ solution) must be known not just at one point, but throughout all of space. In addition, we must constrain the solution for all times by requiring it to have a specified form in some region of space [the *boundary conditions*, e.g., $U(x = a, y, z, t) = 12$]. This makes the algorithms for the solutions of PDEs more complicated than those for ODEs.

As time evolves, the changes in the field $U(x, y, z, t)$ at any one position affects the field at neighboring points. While the time evolution is similar to *ordinary differential equations*, there are now couplings to simultaneous variations in the space dimensions, and so our algorithms will make *finite-difference* steps in both time and space. For a realistic problem requiring a reliable level of precision, this may lead to an incredibly large number of coupled equations. In the next few chapters we will investigate various problems leading to different types of partial differential equations and correspondingly different approaches to solving them.

The most general form of a 2-D time-independent PDE is

$$A(x,y)\frac{\partial^2 U}{\partial x^2} + 2B(x,y)\frac{\partial^2 U}{\partial x \partial y} + C(x,y)\frac{\partial^2 U}{\partial y^2} = F(x,y,U,\frac{\partial U}{\partial x},\frac{\partial U}{\partial y}), \quad (25.1)$$

where A, B, C, and F are general functions. For the special case where

$$B^2(x,y) = A(x,y)C(x,y) \quad (25.2)$$

for all x and y, the equation is called *parabolic*. An example with $B = C = 0$, is the 1-D heat equation:

$$\frac{\partial U(x,t)}{\partial t} = \frac{k}{c\rho}\frac{\partial^2 U(x,t)}{\partial x^2}. \quad (25.3)$$

When $B^2 > AC$ for all x and y, the equation is called *hyperbolic*. An example with $B = 0$ and $AC < 0$ is the 2-D wave equation:

$$\frac{\partial^2 \psi(x,y,t)}{\partial x^2} + \frac{\partial^2 \psi(x,y,t)}{\partial y^2} = \frac{1}{c^2}\frac{\partial^2 \psi(x,y,t)}{\partial t^2}. \quad (25.4)$$

Here the y of (25.1) has become the time variable. When $AC > B^2$ for all x and y, the equation is *elliptic*. An example is Laplace's equation (25.7).

25.2 PROBLEM: DETERMINING AN ELECTROSTATIC POTENTIAL

Your **problem** is to find the electric potential for all points *inside* the charge-free square shown in Fig. 25.1. The bottom and sides of the region are made up of wires that are "grounded" (kept at 0 V). The top has a different wire running across it, connected to a battery that keeps it at a constant 100 V. Once the electric potential is known, you should also determine the nature of the electric field **E**.

25.3 THEORY: LAPLACE'S EQUATION (ELLIPTIC PDE)

It is known from classical electrodynamics [Jack 75] that the electric potential $U(\mathbf{x})$ satisfies Poisson's PDE:

$$\nabla^2 U(\mathbf{x}) = -4\pi\rho(\mathbf{x}). \quad (25.5)$$

Here $\rho(\mathbf{x})$ is the charge density at the spatial location $\mathbf{x}$, and we leave off its time dependence because we are dealing with an electrostatics problem. In charge-free regions of space, that is, regions where $\rho(\mathbf{x}) = 0$, the scalar

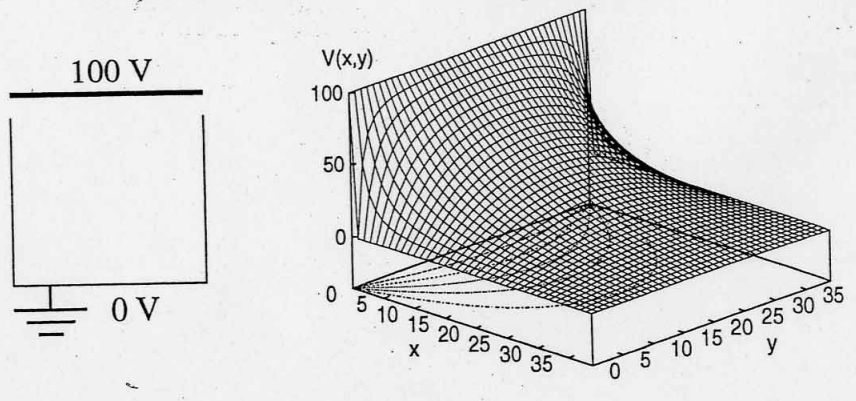


Fig. 25.1 (*Left*) The region of space within a square in which we want to determine the electric potential. There is a wire at the top kept at a constant 100 V and a grounded wire at the sides and bottom. (*Right*) The electric potential V ($\equiv U$) for this geometry. The projection onto the xy plane gives the equipotential lines.

potential satisfies *Laplace's equation*:

$$\nabla^2 U(\mathbf{x}) = 0. \quad (25.6)$$

This equation is an elliptic PDE with physical applications beyond electrostatics. In 2-D rectangular coordinates it takes the form

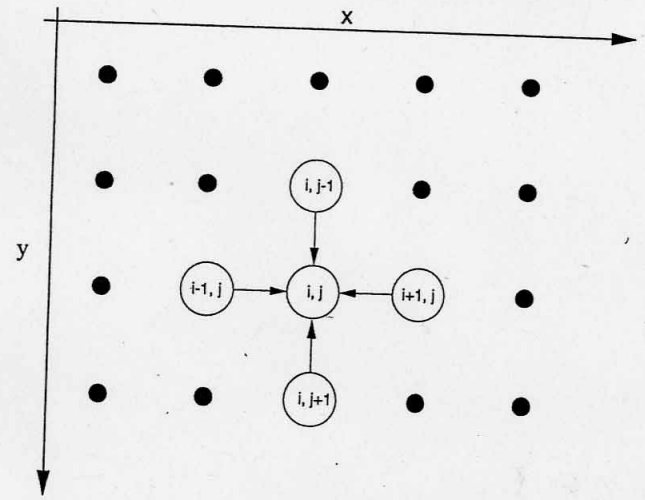
$$\boxed{\frac{\partial^2 U(x,y)}{\partial x^2} + \frac{\partial^2 U(x,y)}{\partial y^2} = 0,} \quad (25.7)$$

which shows that the potential depends simultaneously on x and y. The mathematical problem is to find $U(x,y)$ within a boundary, given its values along the boundary.

25.4 METHOD, NUMERICAL: FINITE DIFFERENCE

Mathematical theorems tell us that if the potential $U(x,y)$ is known along a specific boundary, solutions must exist inside the boundary.[1]

[1]The solution outside the boundary cannot always be found analytically, although its existence is physically obvious. It can be found numerically. Of course, we view infinity as our "boundary," and make the potential zero there. Then being outside the square is equivalent to being "between" two conductors, which is a problem we can solve.



25.2 The algorithm for Laplace's equation in which the potential at the point $) = (i, j)\Delta$ equals the average of the potential values at the four nearest-neighbor :s. The boundary conditions are imposed by having the potentials along the ıeter remain at fixed values.

› deduce an algorithm (illustrated in Fig. 25.2) for the numeric solution 5.7), we follow similar steps to those used in §8.1 to derive the forward- ·ence algorithm for differentiation. We expand the potential at the points $\Delta x, y)$ as a Taylor series in the x variable:

$$U(x+\Delta x,y) = U(x,y)+\frac{\partial U}{\partial x}\Delta x+\frac{1}{2}\frac{\partial^2 U}{\partial x^2}(\Delta x)^2+\cdots, \quad (25.8)$$

$$U(x-\Delta x,y) = U(x,y)-\frac{\partial U}{\partial x}\Delta x+\frac{1}{2}\frac{\partial^2 U}{\partial x^2}(\Delta x)^2+\cdots. \quad (25.9)$$

ı we add these equations and keep terms of order $(\Delta x)^2$, the linear terms : cancel and we obtain a central-difference approximation to the second ıtive for the partial derivative $\partial^2 U(x,y)/\partial x^2$:

$$\frac{\partial^2 U(x,y)}{\partial x^2} \simeq \frac{U(x+\Delta x,y)+U(x-\Delta x,y)-2U(x,y)}{(\Delta x)^2}. \quad (25.10)$$

ise, we expand $U(x, y \pm \Delta y)$ as a Taylor series in y to obtain

$$\frac{\partial^2 U(x,y)}{\partial y^2} \simeq \frac{U(x,y+\Delta y)+U(x,y-\Delta y)-2U(x,y)}{(\Delta y)^2}. \quad (25.11)$$

We expect the error in these finite-difference equations to be proportional to Δ^4. Substituting these approximations into (25.7) leads to a finite-difference approximation (Fig. 25.2) to Laplace's PDE:

$$\frac{U(x+\Delta x,y)+U(x-\Delta x,y)-2U(x,y)}{(\Delta x)^2} + \frac{U(x,y+\Delta y)+U(x,y-\Delta y)-2U(x,y)}{(\Delta y)^2} \simeq 0. \quad (25.12)$$

The reader will notice that (25.12) is a relation among the solutions at five points in space. To utilize it, we divide space up into a lattice, and solve for U at each site on the lattice. This results in a set of linear algebraic equations. One approach is to solve these linear equations explicitly as a (big) matrix problem, using the computer to do the matrix algebra. This is attractive as a direct solution, but it requires a great deal of computing time and memory.

The approach we follow is a simple one. We assume equal grid spacings, $\Delta x = \Delta y = \Delta$ in (25.12) to obtain

$$\boxed{U(x,y) \simeq \tfrac{1}{4}\left[U(x+\Delta,y)+U(x-\Delta,y)+U(x,y+\Delta)+U(x,y-\Delta)\right].} \quad (25.13)$$

Equation (25.13) is our basic algorithm for Laplace's equation in two dimensions. It approximates the potential at point (x, y) as the average of the potential values at the four nearest neighbors, as shown in Fig. 25.2. It is applied as part of an iterative scheme in which we start with the solution along the boundaries and an initial guess for the rest of the solution, use (25.13) to obtain an improved solution, and keep repeating the algorithm until stability is attained.

As is often true in the numerical solution of PDEs, this algorithm is not of as high an order as those used to solve ordinary differential equations (e.g., *rk4*), and so for a given step size Δ, is not as accurate. We will see the effect of this crudeness in the large number of iterations needed to obtain convergence. For serious calculations that use large amounts of computer time, a more accurate, "industrial-strength" algorithm is probably worth the programming complexity.

25.5 METHOD, ANALYTIC: POLYNOMIAL EXPANSIONS

We want the analytic solution of Laplace's equation

$$\frac{\partial^2 U(x,y)}{\partial x^2}+\frac{\partial^2 U(x,y)}{\partial y^2}=0, \quad (25.14)$$

with the boundary conditions given along a square of side L. We assume that the potential is the product of independent functions of x and y:

$$U(x,y) = X(x)Y(y), \tag{25.15}$$

and substitute this product into (25.14). After dividing the resulting equation by $X(x)Y(y)$ we obtain

$$\frac{d^2X(x)/dx^2}{X(x)} + \frac{d^2Y(y)/dy^2}{Y(y)} = 0. \tag{25.16}$$

Because $X(x)$ is a function of only x and $Y(y)$ of only y, the derivatives in (25.16) are *ordinary* as opposed to *partial* derivatives. Since $X(x)$ and $Y(y)$ are assumed to be independent, the only way (25.16) can be valid for *all* values of x and y is for each term in (25.16) to be equal to a constant:

$$\frac{d^2Y(y)/dy^2}{Y(y)} = -\frac{d^2X(x)/dx^2}{X(x)} = k^2. \tag{25.17}$$

We now have two, noncoupled ordinary differential equations:

$$\frac{d^2X(x)}{dx^2} + k^2X(x) = 0, \tag{25.18}$$

$$\frac{d^2Y(y)}{dy^2} - k^2Y(y) = 0. \tag{25.19}$$

We shall see that this choice of sign for the constant matches the boundary conditions and gives us periodic behavior in x. The other choice of sign would give periodic behavior in y, and that would not work. The solutions for $X(x)$ are periodic and those for $Y(y)$ are exponential:

$$X(x) = A\sin kx + B\cos kx, \tag{25.20}$$

$$Y(y) = Ce^{ky} + De^{-ky}. \tag{25.21}$$

The $x = 0$ boundary condition, $U(x = 0, y) = 0$, can be met only if $B = 0$. The $x = L$ boundary condition, $U(x = L, y) = 0$, can be met only for values of k for which

$$kL = n\pi, \quad n = 1, 2, \ldots. \tag{25.22}$$

Accordingly, for each value of n there is a solution for X that we label as $X_n(x)$:

$$X_n(x) = A_n \sin\left(\frac{n\pi}{L}x\right). \tag{25.23}$$

For each value of k_n which satisfies the x boundary conditions, the y solution $Y(y)$ must satisfy the boundary condition $U(x, y = 0) = 0$. This requires

$D = -C$ in (25.21), and so

$$Y_n(y) = C(e^{k_n y} - e^{-k_n y}) \equiv 2C\sinh\left(\frac{n\pi}{L}y\right). \tag{25.24}$$

Because we are solving linear equations, the principle of linear superposition holds and this means that the most general solution is the sum of the products $X_n(x)Y_n(y)$:

$$U(x,y) = \sum_{n=1}^{\infty} E_n \sin\left(\frac{n\pi}{L}x\right)\sinh\left(\frac{n\pi}{L}y\right). \tag{25.25}$$

The E_n values are arbitrary constants and are fixed by requiring the solution to satisfy the remaining boundary condition at $y = L$. In general, this boundary condition could be that the potential has some specific functional dependence. For our **problem** the boundary condition is $U(x, L) = 100$ V, and so

$$\sum_{n=1}^{\infty} E_n \sin\frac{n\pi}{L}x \sinh n\pi = 100 \text{ V}. \tag{25.26}$$

We determine the constants E_n by projection. We multiply both sides of the equation by $\sin m\pi/Lx$, with m an integer, and integrate from 0 to L:

$$\sum_{n}^{\infty} E_n \sinh n\pi \int_0^L dx \sin\frac{n\pi}{L}x \sin\frac{m\pi}{L}x = \int_0^L dx 100 \sin\frac{m\pi}{L}x. \tag{25.27}$$

Yet the integral on the LHS is nonzero only for $n = m$, in which case we can solve for E_n:

$$E_n = \begin{cases} 0, & \text{for } n \text{ even}, \\ \frac{4(100)}{n\pi \sinh n\pi}, & \text{for } n \text{ odd}. \end{cases} \tag{25.28}$$

Finally, we obtain the potential at any point (x, y),

$$U(x,y) = \sum_{n=1,3,5,\ldots}^{\infty} \frac{400}{n\pi} \sin\left(\frac{n\pi x}{L}\right)\frac{\sinh(n\pi y/L)}{\sinh(n\pi)}. \tag{25.29}$$

At this point it is interesting to observe that the solution via the numerical algorithm (25.13) starts with the values of the potential on the boundaries and then propagates them through all space via many iterations. The numerical solution keeps repeating the algorithm until a stable solution results. In contrast, the analytic solution has the x and y dependence explicit via a double Fourier series, but must keep summing terms until a stable solution results.

It is important to notice when evaluating the analytic solution, that the sinh functions in (25.29) may overflow for large n. Some of these overflows can be avoided by expressing the quotient of the two hyperbolic sine functions

in terms of exponentials:

$$\frac{\sinh(n\pi y/L)}{\sinh(n\pi)} = \frac{e^{n\pi(y/L-1)} - e^{-n\pi(y/L+1)}}{1 - e^{-2n\pi}}. \qquad (25.30)$$

While the e^n/n term still gets large, the sum of term with alternating signs n (25.29) converges.

5.6 IMPLEMENTATION: SOLUTION ON LATTICE, LAPLACE.F

We divide the square into a lattice with equal spacing Δ in both in the x and directions. The x and y variables are now discrete:

$$x = x_0 + i\Delta, \quad y = y_0 + j\Delta, \quad (i, j = 0, N_{\max} = L/D). \qquad (25.31)$$

We represent the potential by the array $U(N_{\max}, N_{\max})$. The finite-difference gorithm (25.13) is now

$$U(i,j) = \tfrac{1}{4}\left[U(i+1,j) + U(i-1,j) + U(i,j+1) + U(i,j-1)\right], \qquad (25.32)$$

th the following boundary conditions:

$$\begin{array}{lcll} U(i, N_{\max}) & = & 100, & \text{(top)}, \\ U(1,j) & = & 0, & \text{(left)}, \\ U(N_{\max}, j) & = & 0, & \text{(right)}, \\ U(i,1) & = & 0, & \text{(bottom)}. \end{array} \qquad (25.33)$$

e average (25.32) is used to find the potential at each position (i, j), starting n an edge and working inwards. Successive iterations are obtained by using .32) on the output of the previous iteration, the iterations ending when the nges in the potential are insignificant.

- Write a program or modify the one on the diskette or Web to find the solution of Laplace's equation within a square of side L.
- Increase the number of iterations until you see there is no significant change in the potential throughout the entire region. Because many computations are involved, do the calculation in double precision.
- Impose the boundary conditions on the four sides of the square. There is an ambiguity in the top corners because they can be 100 V or 0.
- Repeat the iteration several hundred times and observe the potential, row by row. Check whether the boundary conditions are always met.
- After you have the numerical solution debugged and stable, compare with the analytic solution (25.29). Do not be surprised if you need to sum thousands of terms before the "analytic" solution converges!

25.7 ASSESSMENT AND VISUALIZATION

1. Observe how the solution converges with successive iterations.
2. Observe how your choice of potential values for the corners leads to inaccuracies near the corners.
3. Repeat the process for different step sizes and judge if it is stable and convergent.
4. You will need hundreds or possibly thousands of iterations to obtain stable and accurate answers (we warned you that you pay a price for simplicity of algorithm).
5. Which solution is more precise, the analytic or numerical?
6. Use a plotting program to draw lines of constant U (potential). These are equipotential surfaces.
7. Either by hand or in some clever way, draw curves orthogonal to the equipotential lines, beginning and ending on the boundaries (where charges lie). The regions of high line density are regions of high electric force.

25.8 EXPLORATION

The numerical solution to the PDE can be used for any boundary conditions. The computer just calculates different numbers on the boundary and then iterates as before. Once you have a running program, have some fun and try out other boundary conditions. Two boundary conditions to try out are triangular

$$U(x,a) = \begin{cases} 200\frac{x}{L}, & \text{for } x \le L/2, \\ 100(1 - \frac{x}{L}), & \text{for } x \ge L/2, \end{cases} \qquad (25.34)$$

and sinusoidal

$$U(x,a) = 100 \sin\left(\frac{2\pi x}{L}\right). \qquad (25.35)$$

Unfortunately, you will have to recalculate the Fourier coefficients for these boundary conditions, or have confidence in your numerical solution and live without an analytic comparison.

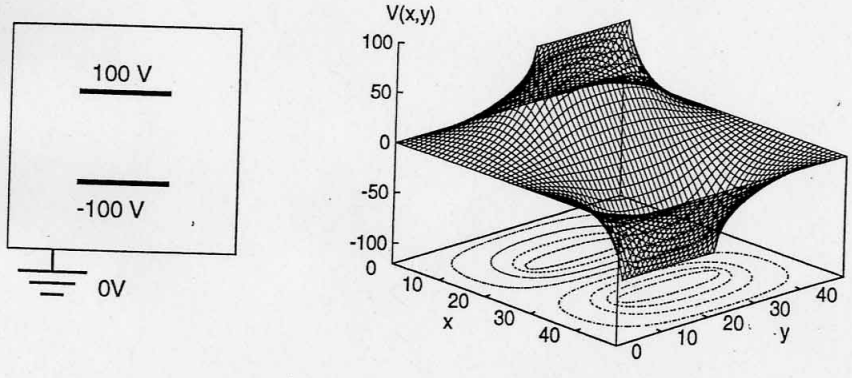


Fig. 25.3 (*Left*) the geometry of a parallel-plate capacitor within a box. A realistic capacitor would have the plates closer together in order to condense the field. (*Right*) The electric potential for this geometry. The projection on the xy plane gives the equipotential lines.

25.9 EXPLORATION: PARALLEL-PLATE CAPACITOR

The standard solution for a capacitor's field is for the region between two infinite plates. We want to see the edge effects and the exterior field when a finite capacitor is placed in a grounded box, as shown in Fig. 25.3. Modify the given program to satisfy these boundary conditions.

Plot the potential and equipotential surfaces. Sketch in the electric field lines (always orthogonal to the equipotential surfaces and beginning and ending on charges). Where is the electric field most intense, and how does it differ from that for an infinite capacitor? Results of our simulation are shown in Fig. 25.3.

25.10 EXPLORATION: FIELD BETWEEN SQUARE CONDUCTORS

You have designed a piece of equipment that is essentially a small metal box at 100 V within a larger, grounded one. You find that sparking occurs inside the box, which indicates too large an electric field. You need to determine where the field is greatest so that you can change the geometry and eliminate the sparking.

Modify the program to satisfy these boundary conditions and to determine the field between the boxes (Gauss's law tells us that the electric field vanishes within the inner box because it contains no charge). Plot the potential and equipotential surfaces, and sketch in the electric field lines (always orthogonal to the equipotential surfaces and beginning and ending on charges). Deduce

where the electric field is most intense and try redesigning the equipment to reduce the field.

26

Heat Flow

26.1 PROBLEM: HEAT FLOW IN A METAL BAR

In Fig. 26.1 we see a metal bar of length $L = 100$ cm and width w located on the x axis. It is insulated along its sides but not its ends. Initially the bar is at a uniform temperature of 100°C, and then both ends are placed in contact with ice water. Heat flows out of the noninsulated ends only. Your **problem** is to determine how the temperature will vary as we move along the bar at any instant of time, and how this variation changes with time.

26.2 MODEL: THE HEAT (PARABOLIC) PDE

A basic fact of nature is that heat flows from hot to cold; that is, from regions of high temperature to regions of low temperature. The rate of heat flow $\mathbf{H}$ through some material is proportional to the gradient of the temperature T within the material:

$$\mathbf{H} = -K\nabla T(\mathbf{x}, t), \tag{26.1}$$

where K is the thermal conductivity of the material. The total amount of heat energy $Q(t)$ in the material at any one time is proportional to the integral of the temperature over the volume of the material:

$$Q(t) = \int d\mathbf{x} C\rho(\mathbf{x}) T(\mathbf{x}, t), \tag{26.2}$$

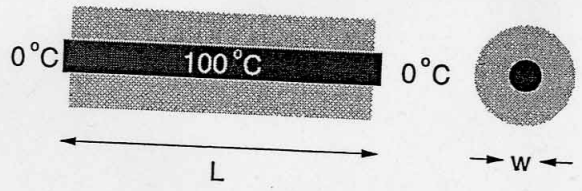


Fig. 26.1 A metallic bar insulated along its length with its ends in contact with heat reservoirs.

where C is the specific heat and ρ the density of the material. Because energy is conserved, the rate of decrease of Q with time must equal the amount of heat flowing out of the material. When this energy balance is struck and the divergence theorem applied, the *heat equation* is the result:

$$\frac{\partial T(\mathbf{x},t)}{\partial t} = \frac{K}{C\rho}\nabla^2 T(\mathbf{x},t), \tag{26.3}$$

where we assume that a constant density ρ.

Equation (26.3) is a parabolic PDE with space and time as independent variables. The setup of this problem implies that there is no temperature variation in directions perpendicular to the bar, and so we have one spatial coordinate to consider in our PDE:

$$\boxed{\frac{\partial T(x,t)}{\partial t} = \frac{K}{C\rho}\frac{\partial^2 T(x,t)}{\partial x^2}.} \tag{26.4}$$

We are given the initial temperature of the bar

$$T(x, t=0) = 100^o\text{C}, \tag{26.5}$$

and must solve equation (26.4) to determine how the temperature changes with time and along the bar. The solution is constrained by the boundary conditions

$$T(x=0,t) = T(x=L,t) = 0, \tag{26.6}$$

that at the ends of the bar are in ice water.

26.3 METHOD, ANALYTIC: POLYNOMIAL EXPANSIONS

The analytic approach is again based on the assumption that a solution exists in which the time and space dependences occur as separate functions:

$$T(x,t) = X(x)\mathcal{T}(t)$$

When (26.7) is substituted into (26.4), and the resulting equation is divided by the product $X(x)\mathcal{T}(t)$, there results two, noncoupled ODEs:

$$\frac{d^2X(x)}{dx^2} + \lambda^2 X(x) = 0, \tag{26.8}$$

$$\frac{d\mathcal{T}(t)}{dt} + \lambda^2\frac{K}{C\rho}\mathcal{T}(t) = 0, \tag{26.9}$$

where λ is a constant to be determined. The boundary condition that the temperature equals zero at $x = 0$ demands the sine function:

$$X(x) = A\sin\lambda x. \tag{26.10}$$

The requirement that the temperature vanish at $x = L$ determines the possible values for the constant λ:

$$\sin\lambda L = 0 \Rightarrow \lambda = \lambda_n = \frac{n\pi}{L}, \quad n = 1,2,\ldots, \tag{26.11}$$

$$\mathcal{T}(t) = e^{-\lambda_n^2 t/C\rho}. \tag{26.12}$$

The analytic solution is in this way

$$T(x,t) = A_n \sin\left(\frac{n\pi x}{L}\right) e^{-\lambda_n^2 t/C\rho}, \tag{26.13}$$

where n can be any odd integer and A_n is an arbitrary constant.

Because the principle of linear superposition holds, the most general solution to (26.4) can be written as a linear superposition of (26.13) using all values of n:

$$T(x,t) = \sum_{n=1}^{\infty} A_n e^{-\lambda_n^2 t/C\rho}\sin(\lambda_n x). \tag{26.14}$$

The Fourier expansion coefficients A_n are determined by the initial condition that at time $t = 0$ the entire bar has a temperature of $T = 100^o$C:

$$T(x,t=0) = T_0 \quad \Rightarrow \quad \sum_{n=1}^{\infty} A_n \sin(\lambda_n x) = T_0. \tag{26.15}$$

As before, we use the orthogonality of different sine functions to determine A_n by projection:

$$A_n = \frac{4T_0}{n\pi}, \quad n = 1,3,5,\ldots, \tag{26.16}$$

The full solution is consequently the infinite series

$$\boxed{T(x,t) = \sum_{n=1,3}^{\infty} \frac{4T_0}{n\pi} e^{-n^2\pi^2 Kt/(L^2 C\rho)} \sin\left(\frac{n\pi x}{L}\right).} \tag{26.17}$$

26.4 METHOD, NUMERICAL: FINITE DIFFERENCE

As with Laplace's equation, the numerical solution is based on converting a differential equation into an approximate finite-difference one. A custom algorithm is then used to generate the solution. The algorithm is derived by expanding $T(x,t+\Delta t)$ and $T(x+\Delta x,t)$ in Taylor series and keeping terms of lowest order in Δ:

$$T(x,t+\Delta t) \simeq T(x,t)+\frac{\partial T(x,t)}{\partial t}\Delta t, \tag{26.18}$$

$$T(x+\Delta x,t) \simeq T(x,t)+\frac{\partial T}{\partial x}\Delta x, \tag{26.19}$$

$$\Rightarrow \quad \frac{\partial T(x,t)}{\partial t} \simeq \frac{T(x,t+\Delta t)-T(x,t)}{\Delta t}, \tag{26.20}$$

$$\frac{\partial^2 T(x,t)}{\partial x^2} \simeq \frac{T(x+\Delta x,t)+T(x-\Delta x,t)-2T(x,t)}{(\Delta x)^2}. \tag{26.21}$$

The PDE (26.4) becomes the finite-difference equation:

$$\frac{T(x,t+\Delta t)-T(x,t)}{\Delta t} \simeq \frac{K}{C\rho}\frac{T(x+\Delta x,t)+T(x-\Delta x,t)-2T(x,t)}{(\Delta x)^2},$$

$$\Rightarrow \quad T(x,t+\Delta t) \simeq T(x,t)+\frac{\Delta t}{(\Delta x)^2}\frac{K}{C\rho} \times [T(x+\Delta x,t)+T(x-\Delta x,t)-2T(x,t)]. \tag{26.22}$$

In discrete form, this is

$$\boxed{T(i,j+1)=T(i,j)+\frac{K\Delta t[T(i+1,j)+T(i-1,j)-2T(i,j)]}{C\rho(\Delta x)^2},} \tag{26.23}$$

where $x=i\Delta x$ and $t=j\Delta t$.

The algorithm described by (26.23) is pictured in Fig. 26.2. We see that the temperature at the point $[x=i\Delta x, t=(j+1)\Delta t]$ [the LHS of (26.23)] is computed from the temperature values at three points of an earlier time [the RHS of (26.23)]. The boundary conditions are imposed by having the temperature along the perimeter fixed. The initial conditions are imposed by giving the initial temperature distribution used to generate the temperature at time Δt. Once the distribution at time Δt is known, it is used to generate the temperature distribution at time $2\Delta t$, and so forth.

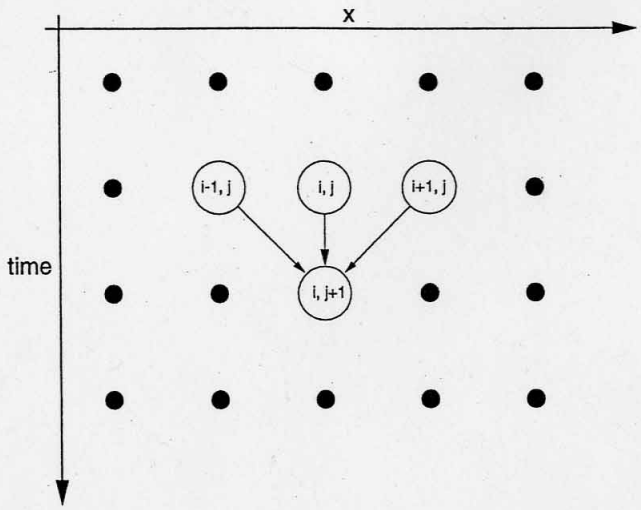


Fig. 26.2 The algorithm for the heat equation in which the temperature at the point $[x=i\Delta x, t=(j+1)\Delta t]$ is computed from the temperature values at three points of an earlier time.

26.5 ANALYTIC ASSESSMENT: ALGORITHM

Generally, PDEs are solved by converting them into finite-difference equations and then using the computer to find a numerical solution to the finite-difference equations. Sometimes we are able to find analytic solutions to the PDEs and test the numerics by comparing these to the analytic solution.[1] The heat equation is an exceptional case in which there *also* exits an analytic solution to the finite-difference equation (26.23) [C&P 88]:

$$T(i,j)=A\left[1-4\frac{K}{C\rho}\frac{\Delta t}{(\Delta x)^2}\sin^2\left(\frac{i\pi\Delta x}{2l}\right)\right]^j \sin\left(\frac{i\pi\Delta x}{l}\right). \tag{26.24}$$

While this analytic solution of the finite difference equation is not a valid solution to the PDE, it is helpful to provide understanding of the algorithm.

Results of the numerical simulation are shown in Fig. 26.3. If we compare the analytic solution of the PDE (26.17) with (26.24), we see that the valid solution decays exponentially with time, but this will not be true[2] for the

[1] In more realistic problems, or for more complicated boundary conditions, finding analytic solutions may be difficult or impossible. In those cases you may be able to find a "test case" similar enough to your problem to permit a check of your numerics.

[2] The argument is based on $\lim_{n\to\infty}(1+x/n)^n = e^x$, and is a little tricky. For example, a

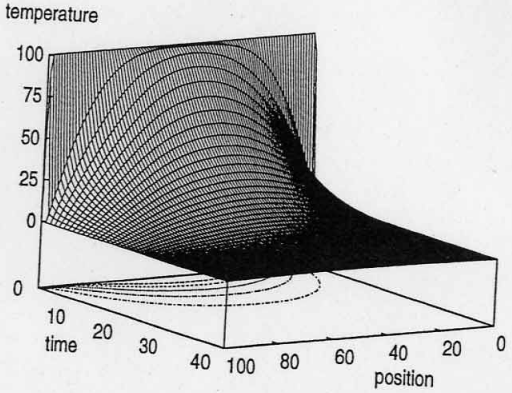


Fig. 26.3 The temperature versus position along the bar and versus time. The initial temperature is constant throughout the bar. The projected contours give the isotherms.

numeric solution unless

$$\frac{K}{C\rho}\frac{\Delta t}{(\Delta x)^2} \leq \tfrac{1}{4}. \tag{26.25}$$

If this condition is not met, the numeric solution will not decay in time and so, clearly, will be wrong. Equation (26.25) tells us that if we make the time step smaller we improve convergence, but if we decrease the space step without a simultaneous quadratic increase in the time step, we worsen convergence. (The lack of symmetry arises because the heat equation is second-order in space but only first-order in time.) Of course, for most problems you would not know the analytic solution of the finite-difference equation, and you would have to try out different combinations of Δx and Δt until a stable and reasonable solution is obtained. You may expect, nonetheless, that there may be choices for Δx and Δt for which the numeric solution fails, and that a separate decrease of Δx or Δt may not improve the solution.

26.6 IMPLEMENTATION, HEAT EQUATION, EQHEAT.F (.C)

Recall, we want to solve for the temperature distribution within an iron bar of length $L = 50$ cm with the boundary conditions

$$T(x = 0, t) = T(x = L, t) = 0, \tag{26.26}$$

$\frac{1}{2}$ on the RHS of (26.25) will give a stable solution, but it will not decay with time.

and initial conditions

$$T(x, t = 0) = 100°\text{C}. \tag{26.27}$$

The constants appropriate to iron are

$$C = 0.113 \text{ cal/(g°C)}, \quad K = 0.12 \text{ cal/(sg°C)}, \quad \rho = 7.8 \text{ g/cc}. \tag{26.28}$$

1. Write or modify the program given on the diskette and Web to solve the heat equation.

2. Define a 2-D array `T(101,2)` for the temperature as a function of space and time. The first index is for the 100 space divisions of the bar, and the second index for present and past times (because thousands of time steps may be made, we save memory by saving only two times).

3. For time $t = 0$, (*j=1*), initialize `T` so that all points on the bar except the end points are at 100°C. Set the temperatures of the ends to 0°C.

4. Apply equation (26.21) to obtain the temperature at the next time.

5. Assign the present-time values of the temperature to the past values:

$$T(i, 1) = T(i, 2), \quad i = 1, \ldots, 101. \tag{26.29}$$

6. Start running with 50 time steps. Once you are confident the program is running properly, use thousands of steps to see the bar cool with time. For every ~500 time steps, print the time and temperature along the bar.

26.7 ASSESSMENT: CONTINUITY, NUMERIC VERSUS ANALYTIC

Extend your program to evaluate the analytic solution at the times and points used for in the numerical solution. You may have to sum the analytic solution over thousands of terms for stability and precision.

1. Make sure your program gives a temperature distribution that varies smoothly along the bar and which agrees with the boundary conditions.

2. Make sure your program gives a temperature distribution that varies smoothly with time and attains equilibrium. You may have to vary the time and space steps to obtain well behaved solutions.

3. Compare the analytic and numeric solutions (and the times needed to compute them). If the solutions differ, suspect the one which does not appear smooth and continuous.

26.8 ASSESSMENT: VISUALIZATION

1. Make 2-D plots of the temperature versus position along the bar.
2. Better yet, make a 3-D plot of the temperature versus position versus time.
3. Make a plot of the contours of constant temperature, the *isotherms*.

26.9 EXPLORATION

Stability test: Check that the temperature diverges with time if the constant C in equation (26.25) is made larger than 0.5.

Material dependence: Repeat the calculation for aluminum, $C = 0.217$ cal/(g°C), $K = 0.49$ cal/(g°C), $\rho = 2.7$ g/cc. Take note that the stability condition requires you to change the size of the time step.

Scaling: The shape of the temperature versus time curve may be the same for different materials, but not the scale. Which of the two bars cools faster?

Sinusoidal initial distribution: $\sin(\pi x/L)$. (This may seem somewhat artificial, but it leads to attractive graphs.) Use the same constants as in the example, and go out to 3000 time steps, printing out results every 150 steps to watch the bar cool. In this case we can compare to the analytic solution,

$$T(x,t) = \sin\frac{\pi x}{L} e^{-\pi^2 Rt/L^2}, \quad R = \frac{k}{C\rho}. \tag{26.30}$$

Two bars in contact: Assume that there are two identical bars, each 25 cm long, as shown in Fig. 26.4. One bar is kept in a heat bath at 100°C, and the other at 50°C. They are put in contact along one of their ends with their other ends kept at 0°C. Determine how the temperature varies with time and location.

Radiating bar (Newton's cooling): Imagine now, that instead of being insulated along its length, a single bar is in contact with an environment at a temperature T_e different from the initial temperature of the bar. In this case, Newton's law of cooling (radiation) says that the rate of temperature change due to radiation is

$$\frac{\partial T}{\partial t} = -h(T - T_e), \tag{26.31}$$

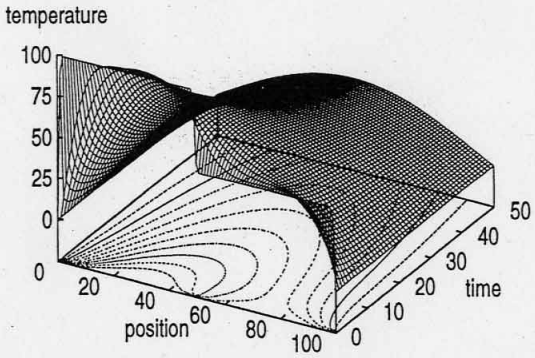


Fig. 26.4 The temperature versus position and time when two bars of differing temperature are placed in contact at $t = 0$. The contours projected onto the position-time plane give the isotherms.

where h is a positive constant. The heat equation is now modified to

$$\frac{\partial T(x,t)}{\partial t} = \frac{k}{C\rho}\frac{\partial^2 T}{\partial^2 x} - hT(x,t). \tag{26.32}$$

Modify the algorithm and program to include Newton's cooling along the length of the bar. Compare the cooling of this bar with that of the insulated bar.

27

Waves on a Strin

27.1 PROBLEM: A VIBRATING STRING

You pluck a string and some pattern of waves follow. Your **problem** is t
predict this pattern when the pluck is 1 mm in height. You may have see
that if you pluck at one location and let go gently, a pulse or traveling wave
observed on the string. And if you shake the string just right, a standing-wa
pattern, in which the nodes remain in the same place for all times, may resul
Actually, we want to solve the problem for all of these possibilities.[1]

27.2 MODEL: THE WAVE EQUATION (HYPERBOLIC PDE)

As our model, we consider a string of length l, tied down at both ends a
shown in Fig. 27.1. The string has a constant density per unit length ρ,
constant tension τ, and is subject to no frictional or gravitational forces. Th
vertical displacement of the string from its rest position is described by
function of two variables $y(x,t)$, where x is the horizontal location along th
string and t the time. We assume that the displacement of the string y is onl
in the vertical direction.

To obtain a linear equation of motion (nonlinear PDEs are discussed i
Chapters 28 and 29), we assume that the displacement and slope are small
If we isolate an infinitesimal section Δx of the string, we know from Newton'

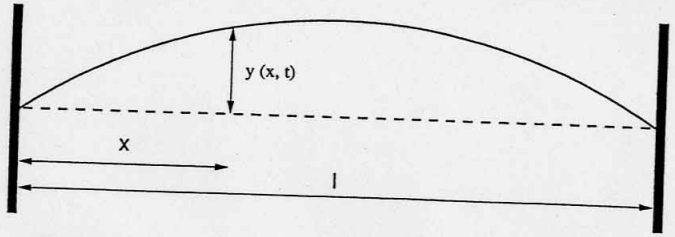


Fig. 27.1 A stretched string of length l tied down at both ends. The horizontal position along the string is given by x and the vertical disturbance of the string from its equilibrium position by $y(x,t)$.

second law of motion that the sum of the vertical forces on the string section must equal the mass times the vertical acceleration of the section:

$$\sum F_y = \rho \Delta x \frac{\partial^2 y}{\partial t^2}. \tag{27.1}$$

Here the forces are the components of the string's tension τ. The vertical components of the tension on each end of the segment change as the angle of the string changes, and we obtain those components by relating the slope of the string to $\partial y/\partial x$:

$$\sum F_y = \tau \left[\left(\frac{\partial y}{\partial x} \right)_{x+\Delta x} - \left(\frac{\partial y}{\partial x} \right)_x \right] = \tau \frac{\partial^2 y}{\partial x^2}, \tag{27.2}$$

$$\Rightarrow \quad \frac{\partial^2 y(x,t)}{\partial x^2} = \frac{1}{c^2} \frac{\partial^2 y(x,t)}{\partial t^2}. \tag{27.3}$$

Here the propagation speed is denoted by

$$c = \sqrt{\tau/\rho}. \tag{27.4}$$

Observe in these equations that y, the height of the string, is the dependent variable, and that the position along the string x and the time t are *both* independent variables. The existence of two independent variables makes this a PDE.

Because both ends of the string are tied down, the *boundary conditions* are that the displacements must vanish for all times at the end of the string:

$$y(0,t) = y(l,t) \equiv 0, \quad \text{(boundary conditions)}. \tag{27.5}$$

As stated in the **problem** specification, the *initial condition* is that at $t = 0$ the right side of the string is "plucked," that is, the string is lifted 1 mm at

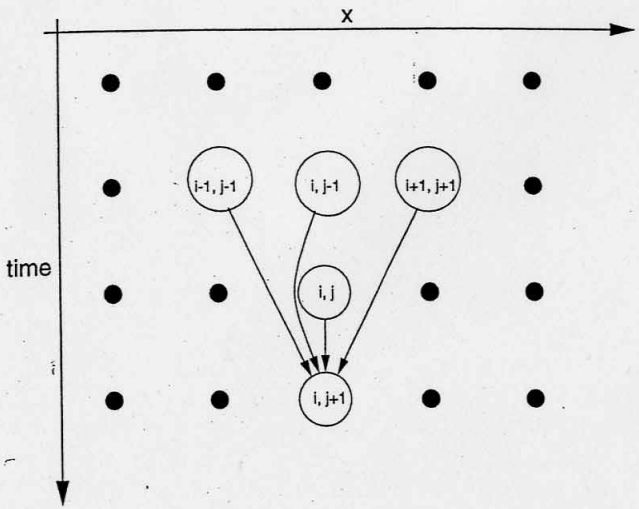


Fig. 27.2 Time steps in the algorithm for the vibrating string.

$x = 0.8l$. We model the "pluck" with the mathematical function:

$$y(x, t=0) = \begin{cases} 1.25x/l, & \text{for } x \le 0.8l, \\ 5.0(1 - x/l), & \text{for } x > 0.8l, \end{cases} \quad \text{(initial condition 1)}. \tag{27.6}$$

Because (27.3) is a second-order equation in time, a second initial condition (beyond initial displacement) is needed to determine the solution. We take that second condition to be that the plucked string is released from *rest*:

$$\frac{\partial y}{\partial t}(x, t=0) = 0, \quad \text{(initial condition 2)}. \tag{27.7}$$

27.3 METHOD, NUMERICAL: TIME STEPPING

As was done with Laplace's equation in Chapter 25, *Electrostatic Potentials*, we look for a solution to our PDE on the 2-D grid shown in Fig. 27.2. In the present case the horizontal axis (first index) represents the position x along the string and the vertical axis (second index) represents time. We assign discrete variables to x and t:

$$x = i\Delta x, \quad t = j\Delta t, \tag{27.8}$$

and represent y as $y(i,j)$. We convert the wave equation (27.3) into a finite-difference equation by expressing the second derivatives in terms of finite

differences:

$$\frac{\partial^2 y(x,t)}{\partial t^2} \simeq \frac{y(i,j+1)+y(i,j-1)-2y(i,j)}{(\Delta t)^2}, \tag{27.9}$$

$$\frac{\partial^2 y(x,t)}{\partial x^2} \simeq \frac{y(i+1,j)+y(i-1,j)-2y(i,j)}{(\Delta x)^2}. \tag{27.10}$$

After substituting (27.9) and (27.10), we obtain final, discrete equation:

$$\boxed{y(i,j+1) = 2y(i,j) - y(i,j-1) + \frac{c^2}{c'^2}\left[y(i+1,j)+y(i-1,j)-2y(i,j)\right],} \tag{27.11}$$

where $c' = \Delta t/\Delta x$, is just a combination of numerical parameters with the dimension of velocity.

As shown in Fig. 27.2, (27.11) is a recurrence relation that propagates the wave from the two earlier times, j and $j-1$, and three nearby positions, $i-1$, i, and $i+1$, to a later time $j+1$ and a single position i (we are using Δx and Δt units). First starting the recurrence relation is a bit tricky because we need to know displacements from two earlier times, whereas the initial conditions are for only one time. To alleviate that difficulty, we convert the initial conditions (27.6) and (27.7) to finite-difference form, and use that to step backward in time! Explicitly, the *central-difference* approximation gives

$$\frac{\partial y}{\partial t}(x, t=0) = 0 \Rightarrow \frac{y(x,\Delta t)-y(x,-\Delta t)}{2\Delta t} = 0, \tag{27.12}$$

$$\Rightarrow \quad y(i,-1) = y(i,1). \tag{27.13}$$

Imposing this condition onto (27.11) for the initial time yields

$$y(i,2) = y(i,1) + \frac{1}{2}\left(\frac{\Delta t}{\Delta x}\right)^2 c^2\left[y(i+1,1)+y(i-1,1)-2y(i,1)\right]. \tag{27.14}$$

We see that (27.14) takes the solution throughout all of space at the initial time $t = 0$ ($j = 1$), and propagates it forward to time Δt. Subsequent advances in time are produced by (27.11).

As we have seen in the finite-difference method applied to heat conduction, the success of the numerical method depends on the relative sizes of the time and space steps. For the present problem there is a similar stability criterion that tells us that the finite-difference solution will be stable if [Cour 28]

$$c \le c' = \frac{\Delta x}{\Delta t}. \tag{27.15}$$

This means that the solution gets better with smaller *time* steps, but gets worse for smaller space steps. This appears somewhat surprising because the wave equation (27.3) is symmetric in x and t. Yet the symmetry is broken by the nonsymmetric way we specify the initial and boundary conditions. A typical numerical solution is shown in Fig. 27.3.

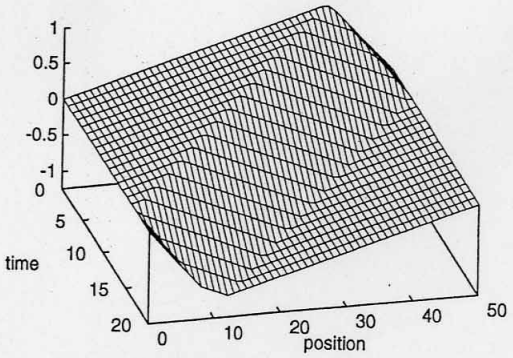


Fig. 27.3 The vertical displacement as a function of position and time of a string initially plucked near its right end. Observe how the initial pulse divides into waves traveling to the right and to the left, and how each traveling wave inverts after its reflection from a fixed end. Notice, too, that the traveling wave moving to the right hits the end first.

27.4 METHOD, ANALYTIC: NORMAL MODES

The analytic solution to (27.3) is obtained via the familiar separation-of-variables technique. We assume that a solution to the wave equation exists that is the product of a function of space times a function of time:

$$y(x,t) = X(x)T(t). \tag{27.16}$$

We substitute (27.16) into (27.3), divide the resulting equation by $y(x,t)$, and are left with an equation that has a solution only if there exists solutions to the two ODEs:

$$\frac{d^2T(t)}{dt^2} + \omega^2 T(t) = 0, \tag{27.17}$$

$$\frac{d^2X(x)}{dt^2} + k^2 X(x) = 0, \tag{27.18}$$

$$k \stackrel{\text{def}}{=} \frac{\omega}{c}. \tag{27.19}$$

Here the angular frequency ω and the wave vector k are determined by demanding that the solutions satisfy the boundary conditions.

The solution for $X(x)$ is required to satisfy the *boundary conditions* that the string is attached at both ends:

$$\begin{aligned} X(x=0,t) &= X(x=l,t)=0 && (27.20)\\ \Rightarrow \quad X_n(x) &= A_n \sin k_n x, && (27.21)\\ k_n &= \frac{2\pi(n+1)}{l}, \quad n=0,1,\ldots. && (27.22) \end{aligned}$$

The corresponding solution for the time equation is

$$\begin{aligned} T_n(t) &= C_n \sin\omega_n t + D_n \cos\omega_n t, && (27.23)\\ \omega_n &= n\omega_0, \quad \omega_0 \stackrel{\text{def}}{=} ck_0 = \frac{2\pi c}{l}. && (27.24) \end{aligned}$$

Solutions of the form (27.16) and (27.23) are the nth *normal modes*, where, by definition, each mode oscillates at a single frequency.

The *initial condition* (27.7) requires the C_n values in (27.23) to be zero. Putting the pieces together, this means that for a string with its ends fixed and initially at rest, there are solutions of the wave equation of the form

$$y_n(x,t) = \sin k_n x \cos\omega_n t, \quad (n=0,1,\ldots). \tag{27.25}$$

Since (27.3) is a linear equation in y, the principle of linear superposition holds and the most general solution can be written as the sum

$$y(x,t) = \sum_{n=0}^{\infty} B_n \sin k_n x \cos\omega_n t. \tag{27.26}$$

The Fourier coefficients B_n are determined by using the second initial condition (27.7), which describes how the wave is plucked. We start with

$$y(x,t=0) = \sum_{n}^{\infty} B_n \sin nk_0 x, \tag{27.27}$$

multiply both sides by $\sin mk_0 x$, substitute the value of $y(x,0)$ from (27.7), and integrate from 0 to l to obtain

$$B_m = -0.0125\frac{\sin 0.8\pi}{m^2\pi^2}. \tag{27.28}$$

We will compare (27.26) 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 in real life can be done only approximately.

27.5 IMPLEMENTATION, WAVE EQUATION, EQSTRING.F (.C)

Modify the program given on the diskette or on the Web to solve for the behavior of the plucked string with ends fixed. Assume that the string has length $l = 1$ m, linear density $\rho = 0.01$ kg/m, and tension $\tau = 40$ N. The initial conditions are given by (27.6) and (27.7). You should get a solution that looks like the one in Fig. 27.3.

1. The program uses a two-dimensional array `y(101,3)`. The first index labels the x position along the string and the second, the three time values used in the recurrence relation (27.14). Time 1 is past, time 2 is present, and time 3 is the future.
2. Choose the space step $\Delta x = 0.01$, that is, 1 cm. Choose the time step Δt such that the stability condition (27.15) predicts a stable solution.
3. Use the initial condition (27.6) to assign `y(i,1)`.
4. Before running the program, print out the values of `y(i,1)` to check that you have assigned the initial displacements correctly.
5. Find the string's displacement for time dt by using the algorithm (27.14) to find `y(i,2)` for all i.
6. Check that the ends of the string remain fixed for all times, that is, `y(1,j) = y(101,j)` $\equiv 0$.
7. Program up a loop that steps the time forward using the solutions from two earlier times, (27.11). Keep repeating this iteration, reassigning the last two columns of `y` to the first two:

```
y(i, 1) = y(i, 2)        Present time becomes past.
y(i, 2) = y(i, 3)        Future time becomes present.
```

8. For every five time increments, write out the displacement of the string.

27.6 ASSESSMENT: VISUALIZATION

Compare the analytic and numerical solutions, keeping at least 200 terms in the "analytic" solution.

1. Use your favorite graphics program to make a 3-D plot of displacement y versus position x versus time t. This is what we did in Fig. 27.3 with *gnuplot*.
2. Observe the motion of the peak as a function of time and use your graph to estimate the peak's propagation velocity c. Compare your deduced c to (27.4).

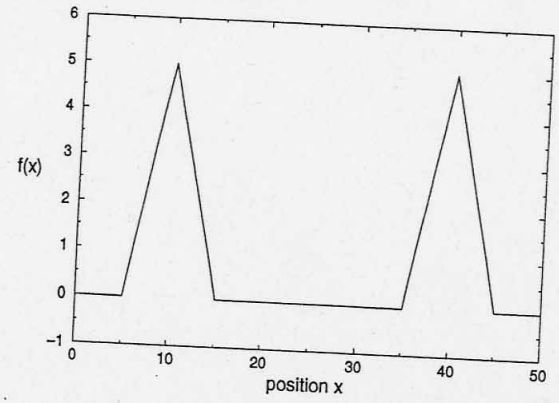


Fig. 27.4 The initial configuration of a string plucked in two places simultaneously. The resulting wavefront as a function of space and time is shown in Fig. 27.5.

27.7 EXPLORATION

Stability: Explore the use of different values for the steps Δx and Δt. Determine at which values the numerical solution becomes unstable. Does your determination agree with the stability condition (27.15)?

Two pulses: Consider a one-meter string with its ends tied down. It is under a tension of 40 N and it has a mass density of 10 g/m. Initially the string is plucked 5 mm at two points, as illustrated in Fig. 27.4. This initial condition is

$$\frac{y(x,t=0)}{0.005} = \begin{cases} 0, & 0.0 \le x \le 0.1, \\ 10x-1, & 0.1 \le x \le 0.2, \\ -10x+3, & 0.2 \le x \le 0.3, \\ 0, & 0.3 \le x \le 0.7, \\ 10x-7, & 0.7 \le x \le 0.8, \\ -10x+9, & 0.8 \le x \le 0.9, \\ 0, & 0.9 \le x \le 1.0. \end{cases} \tag{27.29}$$

Solve for the ensuing motion. In particular, observe whether the pulses move or just oscillate up and down. Our solution is given in Fig. 27.5.

Symmetric vibrating string: You may have seen that when a string is plucked near its end, a pulse reflects off the ends and bounces back and forth. We now want to see what happens if the string is plucked in its middle. Change the initial conditions of the model program to

$$\frac{y(x,t=0)}{0.01} = \begin{cases} x/l, & 0 \le x \le 0.5, \\ -x/l, & 0.5 \le x \le 1. \end{cases} \tag{27.30}$$

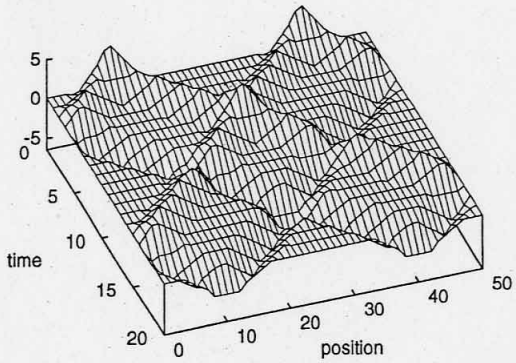


Fig. 27.5 The vertical displacement as a function of position and time of a string initially plucked simultaneously at two points, as shown in Fig. 27.4. Note that each initial peak breaks up into waves traveling to the right and to the left. The traveling waves invert on reflection from the fixed end. As a consequence of these inversions, the $t = 15$ wave is an inverted $t = 0$ wave.

Including friction: We have so far assumed that the string feels no resistance, clearly an idealization because we know that the notes on a guitar fade away rather quickly. The effect of friction on the motion of an element of string between x and $x + dx$ is to oppose the motion of that element. As a model, we assume that the force of friction is proportional to the vertical velocity $\partial y/\partial t$ of the string's element. This changes the wave equation to

$$\frac{\partial^2 y}{\partial t^2} + 2\kappa \frac{\partial y}{\partial t} = c^2 \frac{\partial^2 y}{\partial x^2}. \tag{27.31}$$

The constant κ is proportional to the viscosity of the medium in which the string is vibrating, and is inversely proportional to the density of the string.

Generalize the algorithm for the wave equation to include friction and observe the change in wave behavior. Start off with $T = 40$ N, $\rho = 10$ g/m, and $\kappa = 5$. A solution might look something like that in Fig. 27.6, where damping of the wave is evident. As a check, reverse the sign of κ and see if the wave grows in time (this would eventually violate our assumption of small oscillations).

Normal modes: We know from the analytic solution that there are *normal-mode* solutions to the wave equation that vibrate with one frequency. An example is given in Fig. 27.7. Explore what happens if a frictionless

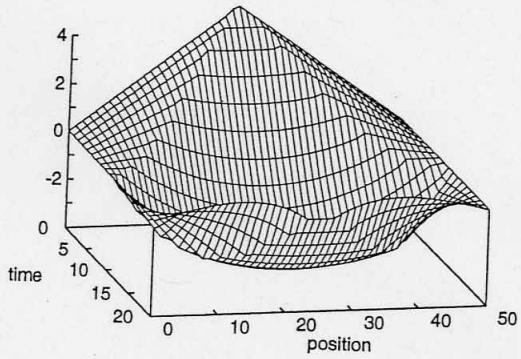


Fig. 27.6 The vertical displacement as a function of position and time of a string with friction nitially plucked at its middle. Here, too, the initial pluck breaks up into waves traveling to the ight and left that get reflected and inverted by the fixed ends. Those parts of the wave with he greatest transverse (y) velocity experience the greatest friction; this effect distorts the wave nd tends to smooth and dampen it.

string is initially placed in a normal mode, for example

$$y(x, t = 0) = 0.001 \sin 2\pi x. \tag{27.32}$$

Try other modes. See if the sum of two modes gives *beating*.

ariable density and tension: If the string has a variable density, then its tension will no longer be a constant, and waves propagate as shown in Fig. 27.8. In this case, the wave equation becomes [F&W 80]:

$$\frac{\partial}{\partial x}\left[T(x)\frac{\partial y(x,t)}{\partial x}\right] = \rho(x)\frac{\partial^2 y(x,t)}{\partial t^2}, \tag{27.33}$$

$$\frac{\partial T(x)}{\partial x}\frac{\partial y(x,t)}{\partial x} + T(x)\frac{\partial^2 y(x,t)}{\partial x^2} = \rho(x)\frac{\partial^2 y(x,t)}{\partial t^2}. \tag{27.34}$$

For constant tension and density, the propagation velocity $c = \sqrt{T/\rho}$. This no longer will be valid for variable densities, and you should observe whether the wave moves faster or slower in regions of high density. To be specific, assume that the density and tension are proportional:

$$\rho(x) = \rho_0 e^{\alpha x}, \tag{27.35}$$

$$T(x) = T_0 e^{\alpha x}. \tag{27.36}$$

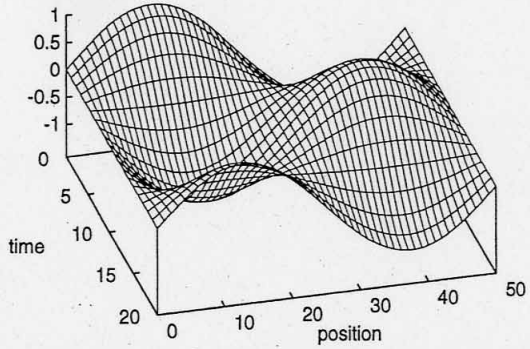


Fig. 27.7 The vertical displacement as a function of position and time of a string initially placed in a normal mode. Notice how the standing wave moves up and down with time.

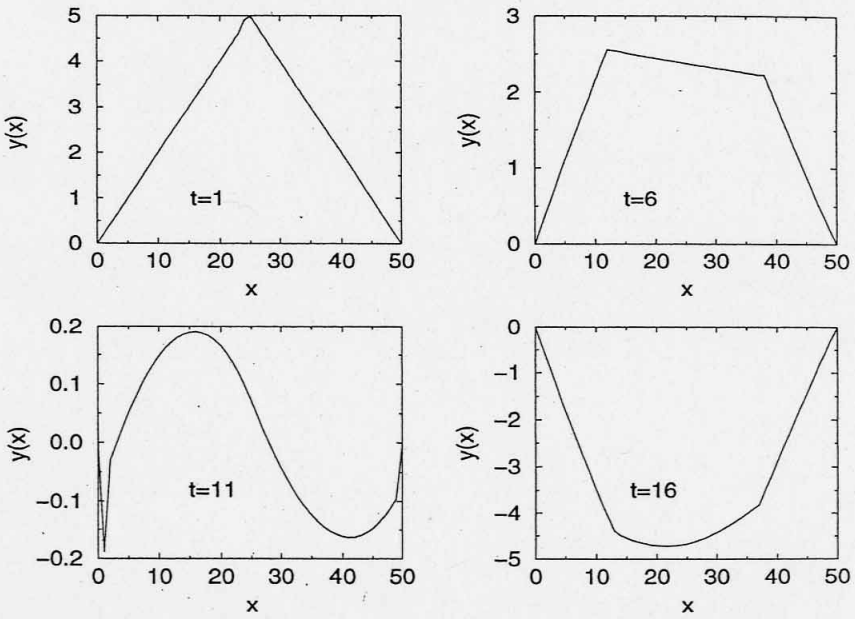


Fig. 27.8 Disturbance versus position for a string with variable density that is initially plucked at its center. The disturbances at four times are given. At $t = 6$ we see that the wave moves faster in the denser region to the right, but that its amplitude decreases because the string is heavier there.

Substitution of these relations into (27.34) yields the new wave equation:

$$\frac{\partial^2 y(x,t)}{\partial x^2} + \alpha \frac{T_0}{\rho_0} \frac{\partial y(x,t)}{\partial x} = \frac{T_0}{\rho_0} \frac{\partial^2 y(x,t)}{\partial t^2}. \tag{27.37}$$

This equation is similar to the wave equation with friction, only now the first derivative is with respect to x. The corresponding difference equations are

$$\begin{aligned} y(i,2) &= y(i,1) + \left(\frac{\Delta t}{\Delta x}\right)^2 \frac{T_0}{2\rho_0}[y(i+1,1) + y(i-1,1) - 2y(i,1)] \\ &\quad + \frac{1}{2}\frac{\alpha(\Delta t)^2 T_0}{\rho_0 \Delta x}[y(i+1,1) - y(i,1)], \end{aligned} \tag{27.38}$$

$$\begin{aligned} y(i,j+1) &= 2y(i,j) - y(i,j-1) + \frac{\alpha(\Delta t)^2 T_0}{\rho_0 \Delta x}[y(i+1,j) - y(i,j)] \\ &\quad + \left(\frac{\Delta t}{\Delta x}\right)^2 \frac{T_0}{\rho_0}[y(i+1,j) + y(i-1,j) - 2y(i,j)]. \end{aligned} \tag{27.39}$$

Modify your program to handle this algorithm with $\alpha = 0.5$, $T_0 = 40$ N, and $\rho_0 = 0.01$ kg/m. Explain in words how the wave motion dampens. The behavior you obtain may look something like that shown in Fig. 27.8.

Part

NONLIN PAR DIFFEREN EQUAT