8

Partial-Differential Equations

The subject of Chapter 6 was ordinary differential equations (ODEs), so called because they involved ordinary derivatives. Some these equations were boundary-value problems where conditions on the problem were specified at the boundaries of some region.

If the region is on a plane or in three-dimensional space, a point in the region has coordinates (x, y) or (x, y, z) and the variation of the dependent function $u = f(x, y, z)$ will be in terms of the space derivatives, $\partial u/dx$, $\partial u/\partial y$, and $\partial u/\partial z$ and/or the corresponding second order derivatives. When a boundary-value problem is defined in terms of these partial derivatives, it is a *partial-differential equation* (PDE). We study PDEs in this chapter.

Types of Partial-Differential Equations

Partial-differential equations (PDEs) are classified as one of three types, with terminology borrowed from the conic sections.

For the second-degree polynomial in x and y,

$$Ax^2 + Bxy + Cy^2 + F = 0,$$

the graph is a quadratic curve, and when

$$B^2 - 4AC < 0, \text{ the curve is an ellipse,}$$
$$B^2 - 4AC = 0, \text{ the curve is a parabola,}$$
$$B^2 - 4AC > 0, \text{ the curve is a hyperbola.}$$

For the general partial-differential equation,

$$A\partial^2 u/\partial x^2 + B\partial^2 u/\partial x \partial y + C\partial^2 u/\partial y^2 + f(x, y, u) = 0,$$

the same terminology is used. If

$$B^2 - 4AC < 0, \quad \text{the equation is elliptic,}$$
$$B^2 - 4AC = 0, \quad \text{the equation is parabolic,}$$
$$B^2 - 4AC > 0, \quad \text{the equation is hyperbolic.}$$

As with the 1-D problems of Chapter 6, the partial-differential equation may have different types of boundary conditions. If the value for u is fixed on some parts of the boundary, it has a *Dirichlet condition* there. If the derivative of u, the *gradient,* is known, it is a *Neumann condition.* (The gradient is always measured along the outward normal.) The condition may be *mixed,* a condition where both the value for u and the gradient is involved. A mixed condition results when heat is lost by conduction or convection to the surroundings.

Elliptic equations describe how a quantity called the *potential* varies within a region. The potential measures the intensity of some quantity (temperature and concentration are "potentials"). The dependent variable, u, that measures the potential at points in the region takes on its equilibrium or *steady-state* value due to values of the potential on the edges or surface of the region. So, elliptic equations are also called potential equations. The general form of an elliptic equation in 2-D is

$$\partial^2 u/\partial x^2 + \partial^2 u/\partial y^2 + f(x, y, u, \partial u/\partial x, \partial u/\partial y) = 0,$$

and we see in comparing with the equations for conic sections that $A = 1$, $B = 0$, and $C = 1$, the values for an ellipse.

How the steady state of the potential is attained from some different starting state is described by a parabolic equation. So, these equations involve time, t, as one of its variables. In effect, we march from the initial state toward the final equilibrium state as time progresses. An important parabolic equation is

$$\partial^2 u/\partial x^2 - (c\rho/k)\, \partial u/\partial t = 0,$$

which tells how temperatures vary with time along a rod subject to certain conditions at its ends. The quantities in $c\rho/k$ are parameters (k = thermal conductivity, ρ = density, c = heat capacity).

Observe that, for this example, $A = 1$, $B = 0$, and $C = 0$, so that $B^2 - 4AC = 0$, the same as for a parabola. This equation and the corresponding ones for 2-D and 3-D regions is then called the *heat equation.* Exactly the same equation but with $c\rho/k$ replaced by 1/D describes the molecular diffusion of matter (D is the *diffusion coefficient*), so the equation in this form is called the *diffusion equation.* The ratio $(k/c\rho)$ is sometimes called the *thermal diffusivity.*

The third type of partial-differential equation, hyperbolic equations, is also time-dependent. It tells how waves are propagated; thus it is called the *wave equation.* In 1-D, it shows how a string vibrates. The partial-differential equation for a vibrating string is

$$\partial^2 u/\partial x^2 - (Tg/w)\, \partial^2 u/\partial t^2 = 0,$$

in which T is the tension in the string, g is acceleration of gravity, and w is the weight per unit length. All of these parameters are positive quantities, so we see that, in comparison to

the conic-section equation, $A = 1$, $B = 0$, and C is a negative quantity. Therefore, $B^2 - 4AC > 0$, the requirement for a hyperbola. In 2-D, the wave equation describes the propagation of waves.

In this chapter, we discuss the usual techniques for solving partial-differential equations numerically. These methods replace the derivatives with finite-difference quotients. You will see that there are limitations to solving these equations in this way because some regions over which we want to solve the problem do not lend themselves to placing the nodes uniformly. There are ways to overcome this but they are awkward and it is not easy to achieve good accuracy in the solution. To some extent, this chapter is preparation for the next where you will find a more recent way to solve PDEs.

Contents of This Chapter

8.1 Elliptic Equations

Extends the derivation of the equation for heat flow in 1-D, along a rod, that was done in Chapter 6 to 2-D (a slab of uniform thickness) and to 3-D objects. Finite-difference quotients are used to approximate the derivatives, allowing one to set up a system of equations whose solution is the steady-state temperatures within the object. Ways to solve the equations more economically are described.

Another form of elliptic equation, called *Poisson's equation,* is employed to find a quantity related to the torsion within a rod when subjected to a twisting force.

8.2 Parabolic Equations

Discusses how temperatures vary with time when heat flows along a rod (1-D) or within a slab (2-D) after deriving the equations for these cases. Beginning with a method that is not very accurate, it progresses to a better technique and then generalizes the procedure to show how these are related.

8.3 Hyperbolic Equations

Begins with the derivation of the equation for determining the lateral displacements of a vibrating string. The equation is solved through finite-difference approximations for the derivatives. Remarkably, the solution is found to match exactly to the analytical solution. Unfortunately, this is found to be not true for a vibrating drum head.

8.1 Elliptic Equations

In Chapter 6, we described how a boundary problem for an ordinary-differential equation could be solved. We now discuss boundary-value problems where the region of interest is two- or three-dimensional. This makes it a partial-differential equation.

There are two standard forms of elliptic partial-differential equations when the object is two-dimensional:

$$\text{Laplace's equation:} \quad -\partial/\partial x\,(c_x \partial u/\partial x + c_y \partial u/\partial y) + au = 0.$$

$$\text{Poisson's equation:} \quad -\partial/\partial x\,(c_x \partial u/\partial x + c_y \partial u/\partial y) + au = f(x, y),$$

where c_x, c_y, and a are parameters of the system that may depend on u and on the values of x and y. u is the variable whose values within the region we desire, the *potential,* at points (x, y) within the 2-D region. Laplace's equation is often called the *potential equation.*

We will deal with a simplified version where $a = 0$. If $c_x = c_y = c$, a constant, the equations can be rewritten as

$$c(\partial^2 u/\partial x^2 + \partial^2 u/\partial y^2) = 0, \quad \text{or} \quad c(\partial^2 u/\partial x^2 + \partial^2 u/\partial y^2) = f(x, y).$$

There is a special symbol that is often used to represent the sum of the second-order partial derivatives:

$$\nabla^2 u = \partial^2 u/\partial x^2 + \partial^2 y/\partial y^2,$$

and the operator ∇^2 is called the *Laplacian.*

Laplace's equation has many applications besides the steady-state distribution of temperature within an object that we use as our model. We chose this because that situation is easier for most people to visualize.

We derived the equation for temperature distribution within a rod, a one dimensional problem, in Chapter 6. We do this now for a two-dimensional region, a flat plate. Figure 8.1 shows a rectangular slab of uniform thickness τ with an element of size $dx \times dy$. u, the dependent variable, is the temperature within the element. We measure to the location of the element from the lower-left corner of the slab. We consider heat to flow through the element in the direction of positive x and positive y.

The rate at which heat flows into the element in the x-direction is

$$\begin{aligned} -(\text{conductivity})\,(\text{area})\,(\text{temperature gradient}) &= -kA\ \partial u/\partial x, \\ &= -k(\tau dy)\ \partial u/\partial x, \end{aligned}$$

where the derivative is a partial derivative because there are two space dimensions.

Similarly, the rate of heat flow into the element in the y-direction is

$$-k(\tau dx)\ \partial u/\partial y.$$

We equate the rate of heat flow into the element to that leaving plus the rate of flow out of the element from the surface of the slab, Q cal/cm^2 (the system is at steady state). For the rate of heat leaving, we must use the gradients at $x + dx$ and $y + dy$:

$$\text{rate of flow out in } x\text{-direction} = -k(\tau dy)\left[\frac{\partial u}{\partial x} + \frac{\partial^2 u}{\partial x^2}\,dx\right],$$

$$\text{rate of flow out in } y\text{-direction} = -k(\tau dx)\left[\frac{\partial u}{\partial y} + \frac{\partial^2 u}{\partial y^2}\,dy\right],$$

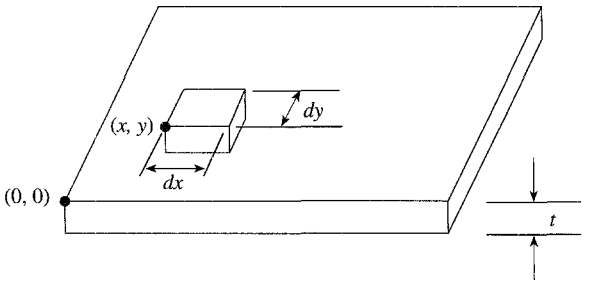


Figure 8.1

so the total flow of heat from the element is

$$-k(\tau\,dy)\left[\frac{\partial u}{\partial x}+\frac{\partial^2 u}{\partial x^2}\,dx\right]-k(\tau\,dx)\left[\frac{\partial u}{\partial y}+\frac{\partial^2 u}{\partial y^2}\,dy\right]+Q(dx\,dy).$$

The sum of the flows into the element must equal the rate at which heat flows from the element plus the heat loss from the surface of the element if the temperature of the element is to remain constant (and we are here considering only the steady-state), so that we have, after some rearrangement:

$$k\tau\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)(dx\,dy)=Q(dx\,dy),$$

or

$$\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)=\frac{Q}{k\tau} \tag{8.1}$$

If the object under consideration is three-dimensional, a similar development leads to

$$\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}+\frac{\partial^2 u}{\partial z^2}\right)=\frac{Q}{k},$$

where now Q is the rate of heat loss per unit volume.

(The loss of heat in the three-dimensional case would have to be through an imbedded "heat-sink," perhaps a cooling coil. It is easier to visualize heat generation within the object, perhaps because there is an electrical current passing through it.)

As we have said, the Laplacian, the sum of the second partial derivatives, is often represented by $\nabla^2 u$, so Eq. (8.1) is frequently seen as

$$\nabla^2 u=\frac{Q}{k\tau}.$$

If the thickness of the plate varies with x and y, a development that parallels that of Section 6.7 gives

$$\tau\nabla^2 u+\frac{\partial\tau}{\partial x}\left(\frac{\partial u}{\partial x}\right)+\frac{\partial\tau}{\partial y}\left(\frac{\partial u}{\partial y}\right)=\frac{Q}{k}. \tag{8.2}$$

If both the thickness and the thermal conductivity are variable:

$$k\tau\nabla^2 u + \left(k\frac{\partial\tau}{\partial x} + \tau\frac{\partial k}{\partial x}\right)\left(\frac{\partial u}{\partial x}\right) + \left(k\frac{\partial\tau}{\partial y} + \tau\frac{\partial k}{\partial y}\right)\left(\frac{\partial u}{\partial y}\right) = Q. \qquad (8.3)$$

Solving for the Temperature Within the Slab

The standard way to obtain a solution to Eqs. (8.1), (8.2), and (8.3) is to approximate the derivatives with finite differences. We will use central differences and assume that the elements are all square and of equal size so that nodes are placed uniformly within the slab. This is relatively easy to do if the slab is rectangular and the height and width are in an appropriate ratio. (If this is not true, another technique, the *finite element method,* which we describe in the next chapter, is most often used.) When the nodes are uniformly spaced so that $\Delta x = \Delta y$, we will use the symbol h for that spacing.

A convenient way to write the central difference approximations to the second partial with respect to x is

$$\frac{\partial^2 u}{\partial x^2} = \frac{(uL - 2uO + uR)}{(\Delta x)^2},$$

where uL and uR are temperatures at nodes to the left and to the right, respectively, of a central node whose temperature is uO. The nodes are Δx apart. A similar formula approximates $\partial^2 u/\partial y^2$:

$$\frac{\partial^2 u}{\partial y^2} = \frac{(uA - 2uO + uB)}{(\Delta y)^2},$$

in which uA and uB are at nodes above and below the central node. It is customary to make $\Delta x = \Delta y = h$. So, if we combine these, we get

$$\nabla^2 u = \frac{(uL + uR + uA + uB - 4uO)}{h^2}.$$

Here is an example.

EXAMPLE 8.1 Solve for the steady-state temperatures in a rectangular slab that is 20 cm wide and 10 cm high. All edges are kept at 0° except the right edge, which is at 100°. There is no heat gained or lost from the surface of the slab. Place nodes in the interior spaced 2.5 cm apart (giving an array of nodes in three rows and seven columns) so that there are a total of 21 internal nodes.

Figure 8.2 is a sketch of the slab with the nodes numbered in succession by rows. We could also number them according to their row and column, with node (1, 1) at the upper left and node (3, 7) at the lower right. However, it is better to number them with a single subscript by rows when we are setting up the equations, as we have done in the figure. (In a second example, the alternative numbering system will be preferred.) Let u_i be the temperature at node (i).

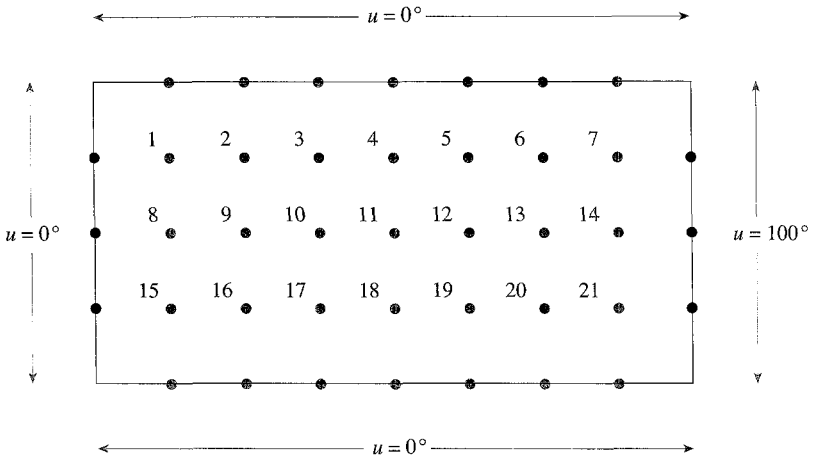


Figure 8.2

The equation that governs this situation is Eq. (8.1) with $Q = 0$:

$$\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) = 0. \tag{8.4}$$

We use these approximations for the second-order derivatives at a central node, where the temperature is uO:

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{(uL - 2uO + uR)}{2.5^2};$$
$$\frac{\partial^2 u}{\partial y^2} \approx \frac{(uA - 2uO + uB)}{2.5^2},$$

where uL and uR are nodes to the left and right of the central node. Similarly, nodes uA and uB are nodes above and below the central node. Substituting these into Eq. (8.4) gives

$$\frac{(uL + uR + uA + uB - 4uO)}{6.25} = 0.$$

There is a simple device we can use to remember this approximation to the Laplacian. We call it a "pictorial operator":

$$\nabla^2 u \approx \frac{1}{h^2}\begin{Bmatrix} & 1 & \\ 1 & -4 & 1 \\ & 1 & \end{Bmatrix} uO. \tag{8.5}$$

This pictorial operator says: Add the temperatures at the four neighbors to uO, subtract 4 times uO, then divide by h^2, and you have an approximation to the Laplacian.

We can now write the 21 equations for the problem. Because in this example we set the Laplacian for every node equal to zero, we can drop the h^2 term. A node that is adjacent to a boundary will have the boundary value(s) in its equation; this will be subtracted from the right-hand side of that equation before we solve the system. Rather than write out all the equations, we will only show a few of them:

For node 1: $0 + u_2 + 0 + u_8 - 4u_1 = 0$, which, when the nodes are put in order, becomes:

$$-4u_1 + u_2 + u_8 = 0.$$

For node 7: $u_6 - 4u_7 + u_{14} = -100.$

For node 9: $u_2 + u_8 - 4u_9 + u_{10} + u_{16} = 0.$

For node 14: $u_7 + u_{13} - 4u_{14} + u_{21} = -100.$

For node 18: $u_{11} + u_{17} - 4u_{18} + u_{19} = 0.$

If we write out all 21 equations in matrix form, we get

$$\begin{bmatrix}
-4 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & -4 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & -4 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & -4 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & -4 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & -4 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & -4 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & -4 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & -4 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & -4 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & -4 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & -4 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & -4 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & -4 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -4 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & -4 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & -4 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & -4 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & -4 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & -4 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & -4
\end{bmatrix}
u =
\begin{bmatrix}
0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -100 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -100 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -100
\end{bmatrix},$$

and we see that the coefficient matrix is symmetric and banded with a band width of 15. There are modifications of Gaussian elimination that can take advantage of the symmetry and bandedness, and we can use less memory to store the coefficients. You will find that numbering the nodes in a different order can reduce the band width to seven. (An exercise at the end of the chapter asks you to find this preferred ordering.)

When the system of equations is solved by Gaussian elmination, we get these results:

Column	Row 1	Row 2	Row 3
1	0.3530	0.4988	0.3530
2	0.9132	1.2894	0.9132
3	2.0103	2.8323	2.0103
4	4.2957	7.0193	4.2931
5	9.1531	12.6537	9.1531
6	19.6631	27.2893	19.6631
7	43.2101	53.1774	43.2101

Rows 1 and 3 are the same; this is to be expected from the symmetry of boundary conditions at the top and bottom of the region. Nodes near the hot edge are warmer than those farther away.

The accuracy of the solution would be improved if the nodes are closer together; the errors decrease about proportional to h^2, which we anticipate because the central difference approximation to the derivative is of $O(h^2)$. Another way to improve the accuracy is to use a nine-point approximation to the Laplacian. This uses the eight nodes that are adjacent to the central node and has an error of $O(h^6)$. A pictorial operator for this is

$$\nabla^2 u \approx \frac{1}{(6h^2)} \begin{Bmatrix} 1 & 4 & 1 \\ 4 & -20 & 4 \\ 1 & 4 & 1 \end{Bmatrix}. \tag{8.6}$$

If Example 8.1 is solved using this nine-point formula and with $h = 2.5$ cm, the answers will be within ± 0.0032 of the "analytical" solution (from a series solution given by classical methods for partial differential equations).

Iterative Methods

The difficulty with getting the solution to a problem in the way that was done in the last example is that a very large matrix is needed when the nodal spacing is close. In that example, if $h = 1.25$, the number of equations increases from 21 to 105; if h were 0.625, there would be 465 equations. The coefficient matrix for 465 equations has $465^2 = 216{,}225$ elements! Not only is this an extravagant use of computer memory to store the values but also the solution time may be excessive. However, the matrix is *sparse*, meaning that most of the elements are zero. (Only about 1% of the elements in the last case are nonzero.)

Iterative methods that were discussed in Chapter 2 are an ideal technique for solving a sparse matrix. We do need to arrange the equations so that there is diagonal dominance (and this is readily possible for the problems of this section). We can write the equations in a form useful for iteration from this pictorial operator:

$$uO = \frac{1}{4} \begin{Bmatrix} & 1 & \\ 1 & 0 & 1 \\ & 1 & \end{Bmatrix}, \tag{8.7}$$

which is, when nodes are specified using row and column subscripts:

$$u_{i,j} = \frac{(u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1})}{4}.$$

We can enter the Dirichlet boundary conditions into the equations by substituting these specified values for the boundary nodes that are adjacent to interior nodes.

The name given to this method of solving boundary-value problems is *Liebmann's method.* We illustrate with the same example problem as Example 8.1.

EXAMPLE 8.2 Solve Example 8.1, but now use Liebmann's method. Use $h = 2.5$ cm.

We will designate the temperatures at the nodes by $u_{i,j}$, where i and j are the row and column for the node. Row 1 is at the top; column 1 is at the left and there are three rows and seven columns for interior nodes. The boundary conditions will be stored in row 0 and row 4, and in column 0 and column 8.

Figure 8.3 shows how nodes are numbered for this problem—we use double subscripts to indicate the row and column.

Here is the typical equation for node (i, j):

$$u_{i,j} = \frac{(u_{i,j-1} + u_{i,j+1} + u_{i-1,j} + u_{i+1,j})}{4}, \quad \text{with } i = 1 \ldots 3, \quad j = 1 \ldots 7.$$

It is best to begin the iterations with approximate values for the $u_{i,j}$, but beginning with all values set to zero will also work. Another way to begin the iterations is with all interior node values set to the average of the boundary values. If this is done, 26 iterations give answers that change by less than 0.0001 and that essentially duplicate those of Example 8.1. (If the starting values are all equal to zero, it takes 30 iterations.)

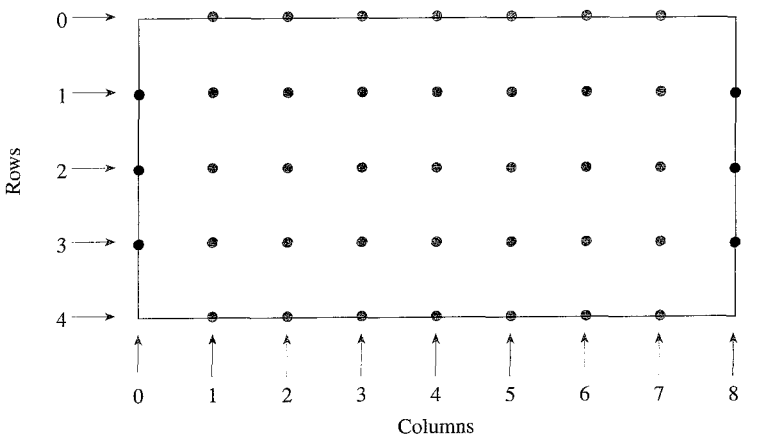


Figure 8.3

Accelerating Convergence in Liebmann's Method

In Chapter 2, it was observed that solving a linear system by iteration can be speeded by applying an *overrelaxation factor* to the process. In the present context, this is called *successive overrelaxation,* abbreviated S.O.R.

To use the S.O.R. techniques, the calculations are made with this formula:

$$u_{i,j} = u_{i,j} + \omega * \frac{(u_{i,j-1} + u_{i,j+1} + u_{i-1,j} + u_{i+1,j} - 4 * u_{i,j})}{4},$$

$$\text{with } i = 1 \ldots 3, \qquad j = 1 \ldots 7,$$

where the $u_{i,j}$ terms on the right are the current values of that variable and the one on the left becomes the new value. The ω-term is called the *overrelaxation factor:*

Solving Example 8.2 with various values for the overrelaxation factor gives these results:

Overrelaxation factor	Number of iterations
1.0	26
1.1	22
1.2	18
1.3	15
1.4	18
1.5	21

From this we see that overrelaxation can decrease the number of iterations required by almost one-half.

The optimal value to use for ω, the overrelaxation factor, is not always predictable. There are methods that use the results of the first few iterations to find a good value. For a rectangular region with Dirichlet boundary conditions, there is a formula:

Optimal ω = smaller root of this quadratic equation = 0:

$$\left[\cos\left(\frac{\pi}{p}\right) + \cos\left(\frac{\pi}{q}\right)\right]^2 \omega^2 - 16\omega + 16, \tag{8.8}$$

where p and q are the number of subdivisions of each of the sides. This formula suggests using $\omega = 1.267$ for the previous example. This is about the same as the value $\omega_{\text{opt}} = 1.3$ that was found by trial and error.

Why Does S.O.R. Accelerate Convergence?

We can find the basis for S.O.R. by examining the rate of convergence of iterative methods, both Gauss–Seidel, which we have used on Example 8.2, and the Jacobi method. Both of these techniques can be expressed in the form

$$x^{(n+1)} = Gx^{(n)} = -Bx^{(n)} + b'. \tag{8.9}$$

(Of course, both methods require that matrix A be diagonally dominant, or nearly so.) The two methods differ, and the difference can be expressed through these matrix equations. where A is written as $L + D + U$:

$$\text{Jacobi:} \qquad x^{(n+1)} = -D^{-1}(L + U)x^{(n)} + D^{-1}b, \tag{8.10}$$

$$\text{Gauss–Seidel:} \quad x^{(n+1)} = -(L + D)^{-1}Ux^{(n)} + (L + D)^{-1}b. \tag{8.11}$$

As Eq. (8.9) makes clear, the rate of convergence depends on how matrix B affects the iterations.

We now discuss how matrix B operates in these two methods. If an iterative method converges, $x^{(n+1)}$ will converge to x, where this last is the solution vector. Because it is the solution, it follows that $Ax = b$. Equation (8.9) becomes, for $x^{n+1} = x^n = x$,

$$x = -Bx + b'.$$

Let $e^{(n)}$ be the error in the nth iteration

$$e^{(n)} = x^{(n)} - x.$$

When there is convergence, $e^{(n)} \to 0$, the zero vector, as n gets large. Using Eq. (8.9) it follows that

$$e^{(n+1)} = -Be^{(n)} = B^2e^{(n-1)} = -B^3e^{(n-2)} = \ldots = (-B)^{n+1}e^{(0)}.$$

Now, if $B^n \to 0$, the zero matrix, it is clear that $e^{(n)} \to 0$. To show when this occurs, we need a principle from linear algebra:

> Any square matrix B can be written as UDU^{-1}. If the eigenvalues of B are distinct, then D is a diagonal matrix with the eigenvalues of B on its diagonal. (If some of the eigenvalues of B are repeated, then D may be triangular, but the argument holds in either case.)

From this we write

$$B = UDU^{-1}, \qquad B^2 = UD^2U^{-1}, \qquad B^3 = UD^3U^{-1}, \ldots, \qquad B^n = UD^nU^{-1}.$$

Now, if all the eigenvalues of B (these are on the diagonal of D) have magnitudes less than one, it is clear that D^n will approach the zero matrix and that means that B^n will also. We then see that iterations converge depending on the eigenvalues of matrix B: They must all be less than one in magnitude. Further, the rate of convergence is more rapid if the largest eigenvalue is small. We also see that even if matrix A is not diagonally dominant, there may still be convergence if the eigenvalues of B are less than unity.

This example will clarify the argument.

EXAMPLE 8.3 Compare the rates of convergence for the Jacobi and Gauss–Seidel methods for $Ax = b$, where

$$A = \begin{bmatrix} 6 & -2 & 1 \\ -2 & 7 & 2 \\ 1 & 2 & -5 \end{bmatrix} \qquad b = \begin{bmatrix} 11 \\ 5 \\ -1 \end{bmatrix}.$$

For this example, we have

$$D\begin{bmatrix}6 & 0 & 0\\ 0 & 7 & 0\\ 0 & 0 & -5\end{bmatrix} \qquad L=\begin{bmatrix}0 & 0 & 0\\ -2 & 0 & 0\\ 1 & 2 & 0\end{bmatrix} \qquad U=\begin{bmatrix}0 & -2 & 1\\ 0 & 0 & 2\\ 0 & 0 & 0\end{bmatrix}.$$

and

$$D^{-1}=\begin{bmatrix}1/6 & 0 & 0\\ 0 & 1/7 & 0\\ 0 & 0 & -1/5\end{bmatrix}.$$

For the Jacobi method, we need to compute the eigenvalues of this B matrix:

$$B=D^{-1}(L+U)=\begin{bmatrix}1/6 & 0 & 0\\ 0 & 1/7 & 0\\ 0 & 0 & 1/5\end{bmatrix}*\begin{bmatrix}0 & -2 & 1\\ -2 & 0 & 2\\ 1 & 2 & 0\end{bmatrix}=\begin{bmatrix}0 & -1/3 & 1/6\\ -2/7 & 0 & 2/7\\ -1/5 & -2/5 & 0\end{bmatrix}.$$

The eigenvalues are $-0.1425 + 0.3366i$, $-0.1425 - 0.3366i$, and 0.2851. The largest in magnitude is 0.3655.

For the Gauss–Seidel method, we need the eigenvalues of this B matrix:

$$B=(L+D)^{-1}U=\begin{bmatrix}1/6 & 0 & 0\\ 1/21 & 1/7 & 0\\ 11/210 & 2/35 & -1/5\end{bmatrix}*\begin{bmatrix}0 & -2 & 1\\ 0 & 0 & 2\\ 0 & 0 & 0\end{bmatrix}=\begin{bmatrix}0 & -1/3 & 1/6\\ 0 & -2/21 & 1/3\\ 0 & -11/105 & 1/6\end{bmatrix},$$

which has these eigenvalues: 0, $0.0357 + 0.1333i$, and $0.0357 - 0.1333i$. The largest in magnitude for the Gauss–Seidel method is 0.1380. We then see that (as expected) the Gauss–Seidel method will converge faster. If we solve this example problem with both methods, starting with [0 0 0] and ending the iterations when the largest change in any element of the solution is less than 0.00001, we find that Gauss–Seidel takes only seven iterations, whereas the Jacobi method takes 12.

We have used overrelaxation (the S.O.R. method) to speed the convergence of the iterations in solving a set of equations by the Gauss–Seidel technique. In view of the last discussion, this must be to reduce the eigenvalue of largest magnitude in the iteration equation. We have used S.O.R. in the following form:

$$x_i^{(n+1)}=x_i^{(n)}+\omega/a_{ii}(b_i-\Sigma a_{ij}x_j^{(n+1)}-\Sigma a_{ij}x_j^{(n)}), \qquad i=1,2,\ldots,N,$$

with the first summation from $j = 1$ to $j = i - 1$ and the second from $j = i$ to $j = N$. As shown before, the standard Gauss–Seidel iteration can be expressed in matrix form:

$$x^{(n+1)}=-(L+D)^{-1}Ux^{(n)}+(L+D)^{-1}b, \tag{8.12}$$

which is more convenient for the present purpose. We want the overrelaxation equation to be in a similar form. From $A = L + D + U$, we can write

$$\omega(b-Ax)=\omega(b-(L+D+U)x)=0.$$

Now, if we add Dx to both sides of this, we get

$$Dx - \omega Lx - \omega Dx - \omega Ux + \omega b = Dx,$$

which can be rearranged into

$$x^{(n+1)} = (D + \omega L)^{-1}[(1 - \omega)D - \omega U]x^{(n)} + \omega(D + \omega L)^{-1}b, \tag{8.13}$$

and this is the S.O.R. form with ω equal to the overrelaxation factor. It is not easy to show in the general case that the eigenvalue of largest magnitude in Eq. (8.13) is smaller than that in Eq. (8.12), but we can do it for a simple example.

EXAMPLE 8.4 Show that overrelaxation will speed the convergence of iterations in solving

$$\begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix} x = \begin{bmatrix} 6 \\ -2 \end{bmatrix}.$$

For this, the Gauss–Seidel iteration matrix is

$$-(L + D)^{-1}U = \begin{bmatrix} 0 & -1/2 \\ 0 & 1/6 \end{bmatrix},$$

whose eigenvalues are 0 and 1/6.

For the overrelaxation equation, the iteration matrix is

$$(D + \omega L)^{-1}[(1 - \omega)D - \omega U] = \begin{bmatrix} 1 - \omega & -\omega/2 \\ -\omega(1 - \omega)/3 & (\omega^2/6 - \omega + 1) \end{bmatrix}. \tag{8.14}$$

We want the eigenvalues of this, which are, of course, functions of ω. We know that, for any matrix, the product of its eigenvalues equals its determinant (why?), so we set

$$\lambda_1 * \lambda_2 = \det(\text{iteration matrix}) = (\omega - 1)^2.$$

To get the smallest possible value for λ_1 and λ_2, we set them equal, so $\lambda_1 = \lambda_2 = (\omega - 1)$. We also know that, for any matrix, the sum of its eigenvalues equals its trace, so

$$\lambda_1 + \lambda_2 = 2(\omega - 1) = \frac{\omega^2}{6} - 2\omega + 2,$$

which has a solution $\omega = 1.045549$. Substituting this value of ω into Eq. (8.14) gives

$$\begin{bmatrix} -0.0455 & -0.5228 \\ 0.0159 & 0.1366 \end{bmatrix},$$

whose eigenvalues are $0.0456 \pm 0.0047i$, whose magnitudes are smaller than the largest for the Gauss–Seidel matrix, which is $1/6 = 0.16667$.

Poisson's Equation

The previous examples were for an equation known as *Laplace's equation:*

$$\nabla^2 u = 0.$$

If the right-hand side is nonzero, we have *Poisson's equation:*

$$\nabla^2 u = R,$$

where R can be a function of position in the region (x, y). To solve a Poisson equation, we need to make only a minor modification to the methods described for Laplace's equation.

EXAMPLE 8.5 Solve for the *torsion function,* ϕ, in a bar of rectangular cross section, whose dimensions are 6 in. × 8 in. (The tangential stresses are proportional to the partial derivatives of the torsion function when the bar is twisted.) The equation for ϕ is

$$\nabla^2 \phi = -2, \qquad \text{with } \phi = 0 \text{ on the outer boundary of the bar's cross section.}$$

If we subdivide the cross section of the bar into 1-in. squares, there will be 35 interior nodes at the corners of these squares ($h = 1$). If we use the iterative technique, the equation for ϕ is

$$\phi_{i,j} = \frac{(\phi_{i,j-1} + \phi_{i,j+1} + \phi_{i-1,j} + \phi_{i+1,j} + 2)}{4}, \qquad i = 1 \ldots 7, \qquad j = 1 \ldots 5.$$

Convergence will be hastened if we employ overrelaxation. Equation (8.8) predicts ω_{opt} to be 1.383. Using overrelaxation with this value for ω converges in 13 iterations to the values in Table 8.1.

If overrelaxation is not employed, it takes 25 iterations to get the values of Table 8.1. Again, overrelaxation cuts the number of iterations about in half.

Derivative Boundary Conditions

Just as we saw in Section 6.7 for a one-dimensional problem, two-dimensional problems may have derivative boundary conditions. These may be of either Neumann or mixed type. We can define a more universal type of boundary conditions by the relation:

$$Au + B = Cu', \qquad \text{where } A, B, \text{ and } C \text{ are constants.}$$

If $C = 0$, we have a Dirichlet condition: $u = -B/A$. If $A = 0$, the condition is Neumann: $u' = B/C$. If none of the constants is zero, it is mixed condition. This relation can match a boundary condition for heat loss from the surface:

$$-ku' = H(u - u_s)$$

Table 8.1 Torsion function at interior nodes for Example 8.5

2.042	3.047	3.353	3.047	2.043
3.123	4.794	5.319	4.794	3.123
3.657	5.686	7.335	5.686	3.657
3.818	5.960	7.647	5.960	3.818
3.657	5.686	7.335	5.686	3.657
3.123	4.794	5.319	4.794	3.123
2.042	3.048	3.354	3.048	2.043

by taking $A = H, B = -H * u_s, C = -k$.

Here is an example that shows how this universal type of boundary conditions can be handled.

EXAMPLE 8.6 Find the steady-state temperatures in a slab that is 5 cm × 9 cm and is 0.5 cm thick. Everywhere within the slab, heat is being generated at the rate of 0.6 cal/sec/cm^3. The two 5-cm edges are held at 20° while heat is lost from the bottom 9-cm edge at a rate such that $\partial u/\partial y = 15$. The top edge exchanges heat with the surroundings according to $-k\,\partial u/\partial y = H * (uO - u_s)$, where k, the thermal conductivity, is 0.16; H, the heat transfer coefficient, is 0.073; and u_s, the temperature of the surroundings, is 25°. (uO in this case is the temperature of a node on the top edge.) No heat is gained or lost from the surfaces of the slab. Place nodes within the slab (and on the edges) at a distance 1 cm apart so that there are a total of 60 nodes.

Figure 8.4 illustrates the problem. In Figure 8.4, rows of fictitious nodes are shown above and below the top and bottom nodes in the slab. These are needed because there are derivative boundary conditions on the top and bottom edges.

The Dirichlet conditions on the left and the right will be handled by initializing the entire array of nodal temperatures to 20°, and omitting these left- and right-edge nodes from the iterations that find new values for the nodal temperatures.

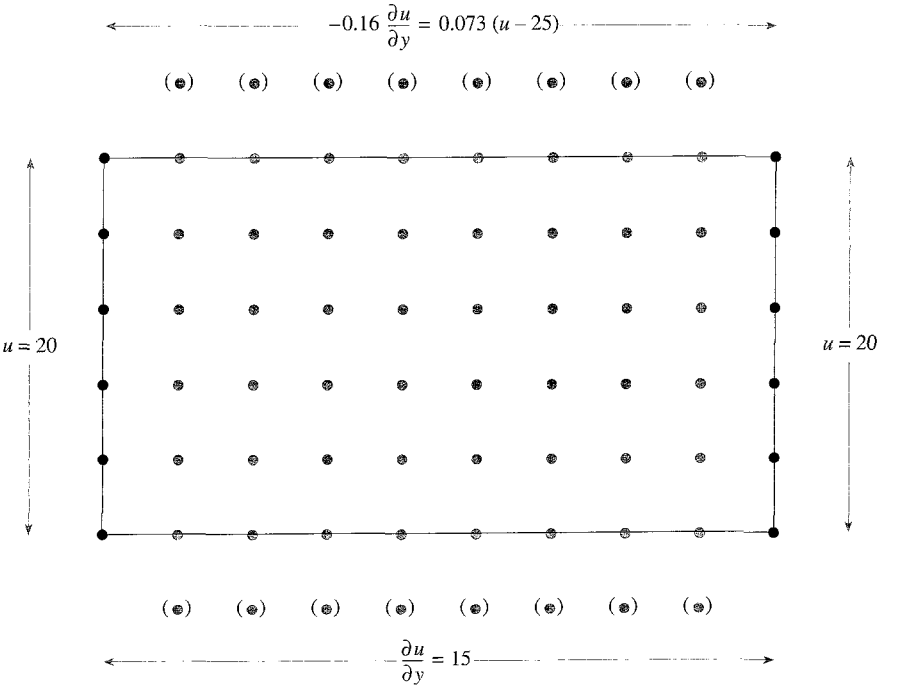


Figure 8.4

(These edge nodes are the uL or uR in the formula:

$$uO = \frac{(uL + uR + uA + uB)}{4} - \frac{Q * h^2}{kt}$$

For computations along the bottom edge (row 6), where $\partial u/\partial y = 15$, the gradient, $\partial u/\partial y$, will be computed by

$$\frac{\partial u}{\partial y} = \frac{(uA - uF)}{2h} = 15, \qquad uF = uA - 2 * h * 15,$$

where uA is at a node in the fifth row and uF is a fictitious node. (Take note of the fact that, if the gradient here is positive, heat flows in the negative y-direction, so heat is being lost as specified.) The equation for computing temperatures along the bottom edge is then

$$uO = \frac{(uL + uR + uA + uB)}{4} - \frac{Q * h^2}{kt}, \qquad \text{with } uB = uF.$$

For computations along the top edge where the relation is

$$-k\frac{\partial u}{\partial y} = H * (uO - uS),$$

temperatures will be computed using a fictitious node above uO, uF, from

$$uO = \frac{(uL + uR + uA + uB)}{4} - \frac{Q * h^2}{kt},$$

where $uA = uF$, and, because

$$\partial u/\partial y = \frac{(uF - uB)}{2h},$$

where uB is a node in the second row, we have

$$-k\frac{\partial u}{\partial y} = \frac{-k * (uF - uB)}{2h} = H * (uO - uS),$$

which gives

$$uA = uF = uB - \left(\frac{2 * H * h}{k}\right) * uO + \left(\frac{2 * H * h}{k}\right) * uS.$$

When these replacements are included in a program and overrelaxation is employed ($\omega = 1.57$), the results after 28 iterations are as shown in Table 8.2.

Table 8.2 Temperatures after 28 iterations for Example 8.6

20.000	73.510	107.915	128.859	138.826	138.826	128.859	107.915	73.510	20.000
20.000	90.195	137.476	167.733	180.743	180.743	167.733	137.476	90.195	20.000
20.000	99.793	155.061	189.855	207.669	207.669	189.855	155.061	99.794	20.000
20.000	103.918	163.119	200.956	219.410	219.410	200.956	163.119	103.919	20.000
20.000	102.762	162.539	201.442	220.604	220.604	201.442	162.539	102.762	20.000
20.000	94.589	152.834	191.669	210.959	210.959	191.669	152.834	94.589	20.000

The Alternating Direction Implicit (A.D.I.) Method

When the partial-differential equations of this chapter are solved (using the finite-difference method), the resulting coefficient matrix is sparse. The sparseness increases as the number of nodes increases: If there are 21 nodes, 81% of the values are zeros; if there are 105 nodes, 96% are zeros; for a $30 \times 30 \times 30$ three-dimensional system, only 0.012% of the $729 * 10^6$ values are nonzero!

The coefficient matrices are not only sparse in two- and three-dimensional problems. They are also *banded,* meaning that the nonzero values fall along diagonal bands within the matrix. There are solution methods that take advantage of this banding, but, because the location of the bands depends strongly on the number of nodes in rows and columns, it is not simple to accomplish. Only for a tridiagonal coefficient matrix is getting the solution straightforward.

One way around the difficulty, as we have shown, is iteration. This is an effective way to decrease the amount of memory needed to store the nonzero coefficients and to (usually) speed up the solution process. However, as we saw in Section 6.7, the system of equations for the one-dimensional case always has a tridiagonal coefficient matrix, and, for this, neither the computational time nor the storage requirements is excessive. We ask "Is there a way to get a tridiagonal coefficient matrix when the region has two or three dimensions?" The answer to this question is yes, and the technique to achieve this is called the *alternating direction implicit method,* usually abbreviated to the *A.D.I. method.*

The trick to get a tridiagonal coefficient matrix for computing the temperatures in a slab is this: First make a traverse of the nodes across the rows and consider the values above and below each node to be constants. These "constants" go on the right-hand sides of the equations, of course. (We know that these "constant" values really do vary, but we will handle that variation in the next step.) After all the nodes have been given new values with the horizontal traverse, we now make a traverse of the nodes by columns, assuming for this step that the values at nodes to the right and left are constants. There is an obvious bias in these computations, but the bias in the horizontal traverse is balanced by the opposing bias of the second step. If the object is three-dimensional, three passes are used: first in the x-direction, then in the y-direction, and finally in the z-direction.

A.D.I. is particularly useful in three-dimensional problems but it is easier to explain with a two-dimensional example. When we attack Laplace's equation in two dimensions, we write the equations as

$$\nabla^2 u = \frac{(uL - 2uO + uR)}{(\Delta x)^2} + \frac{(uA - 2uO + uB)}{(\Delta y)^2} = 0,$$

where, as before, uL, uR, uA, and uB stand for temperatures at the left, right, above, and below the central node, respectively, where it is uO. When, as is customary, $\Delta x = \Delta y$, the denominators can be canceled. The row-wise equations for the $(k + 1)$ iteration are

$$(uL - 2uO + uR)^{(k+1)} = -(uA - 2uO + uB)^{(k)}, \tag{8.15}$$

where the right-hand nodal values are the constants for the equations. When we work column-wise, the equations are for the $(k + 2)$ iteration

$$(uA - 2uO + uB)^{(k+2)} = -(uL - 2uO + uR)^{(k+1)}, \tag{8.16}$$

where, again, the right-hand nodal values are the constants.

We can speed up the convergence of the iterations by introducing an acceleration factor, ρ, to make Eq. (8.15) become

$$uO^{(k+1)} = uO^{(k)} + \rho(uL - 2uO + uR)^{(k+1)} + \rho(uA - 2uO + uB)^{(k)},$$

and Eq. (8.16) becomes

$$uO^{(k+2)} = uO^{(k+1)} + \rho(uA - 2uO + uB)^{(k+2)} + \rho(uL - 2uO + uR)^{(k+1)},$$

where the last terms in both use the values from the previous traverse.

Rearranging further, we get the tridiagonal systems

$$-uL^{(k+1)} + \left(\frac{1}{\rho} + 2\right) uO^{(k+1)} - uR^{(k+1)} = \left[uA + \left(\frac{1}{\rho} - 2\right) uO + uB\right]^{(k)}, \quad (8.17)$$

and

$$-uA^{(k+2)} + \left(\frac{1}{\rho} + 2\right) uO^{(k+2)} - uB^{(k+2)} = \left[uL + \left(\frac{1}{\rho} - 2\right) uO + uR\right]^{(k+1)}, \quad (8.18)$$

for the horizontal and vertical traverses, respectively.

In writing a program for the A.D.I. method, we must take note of the fact that the coefficient matrices for the two traverses are not identical because the boundary values enter differently. Here is a deliberately simple example that illustrates the procedure.

EXAMPLE 8.7 A rectangular plate is 6 in. × 8 in. The top edge (an 8-in. edge) is held at 100°, the right edge at 50°, and the other two edges at 0°. Use the A.D.I. method to find the steady-state temperatures at nodes spaced 1 in. apart within the plate.

There are 5 * 7 = 35 interior nodes, so there are 35 equations in each set (the horizontal and vertical traverses). With $\rho = 0.9$, and starting with all interior values set to 0°, the values of Table 8.3 result after 28 iterations, which is when the maximum change in any of the values is less than 0.001. (If we begin with the interior nodes set to the average of the boundary values, these values are reached in 24 iterations with $\rho = 1.1$.)

For this particular example, the number of nodes is small enough that Liebmann's method with overrelaxation could be used. That method is somewhat more efficient because it requires only 15 iterations to attain the same accuracy.

Table 8.3 Temperatures at interior nodes for Example 8.7

48.523	67.828	74.203	78.669	79.341	77.984	71.464
27.262	44.122	53.644	58.803	61.178	61.132	57.873
17.404	28.754	37.982	42.188	45.434	47.495	48.804
9.599	17.508	23.344	27.534	30.878	34.518	40.209
4.484	8.336	11.352	13.728	17.024	19.492	27.425

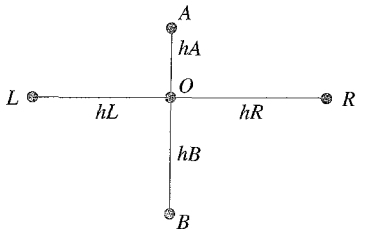


Figure 8.5

Nodes Spaced Nonuniformly

All of the examples that we have used so far have had regions where the nodes can be spaced uniformly. That is not always the case. There are three reasons why we may need a nonuniform spacing:

1. A rectangular region may have width and length incompatible with a uniform spacing.
2. The region may be nonrectangular.
3. We may want nodes closer together in some areas to improve the accuracy where the dependent variable is changing rapidly.

(If the region is three-dimensional, analogous cases apply.)

For case 2, we may be able to change the coordinate system and use an appropriate redefinition of the Laplacian. In any case, we can approximate it for a set of nodes not uniformly spaced. Consider Figure 8.5.

Figure 8.5 illustrates a situation where the four nodes around the central node have different spacing. As shown in the figure, the distances to points L, R, A, and B from point O, the central node, are hL, hR, hA, and hB. These points are nodes to the left, right, above, and below the central node, respectively. The u-values at these points are uL, uR, uA, and uB. Approximate the first derivatives between points L and O and between points O and R with:

$$\left(\frac{\partial u}{\partial x}\right)_{L,O} = \frac{(uO - uL)}{hL}, \qquad \left(\frac{\partial u}{\partial x}\right)_{O,R} = \frac{(uR - uO)}{hR}.$$

These can be interpreted as central difference approximations at points halfway between points L and O and halfway between O and R. We then approximate the second derivative with:

$$\begin{aligned}\frac{\partial^2 u}{\partial x^2} &= \frac{\left[\left(\dfrac{\partial u}{\partial x}\right)_{O,R} - \left(\dfrac{\partial u}{\partial x}\right)_{L,O}\right]}{\left[\dfrac{(hL + hR)}{2}\right]} \\ &= \frac{2}{(hL + hR)}\left[\frac{uL}{hL} - \frac{(hL + hR)}{(hL * hR)/uO} + \frac{uR}{hR}\right],\end{aligned} \tag{8.19}$$

but this is not a central difference approximation at exactly point O. We can use it to approximate the second derivative there but doing so incurs an error of $O(h)$. We can do the same to approximate $\partial^2 u/\partial y^2$ by using the points in a vertical line.

Using Eq. (8.19) is not the best way to handle the problem, however. The *finite-element method (FEM)** is much to be preferred and we describe this in the next chapter. In FEM, the region is divided into subregions and these can be other than squares, usually triangles in 2-D. The subregions, which have common vertices, can be of varying sizes. A boundary that is not straight is approximated by a sequence of straight lines that can be very short where the boundary is sharply curved.

8.2 Parabolic Equations

The second class of partial-differential equations is usually called the *diffusion equation* or the *heat equation* because the typical examples are the molecular diffusion of matter and the flow of heat within regions. We will use heat flow as our example, similarly to Section 8.1. In contrast to that for an elliptic PDE, the situation is not the steady state but is time dependent; temperatures vary with time.

We begin with the 1-D case, but we will extend the treatment to 2-D and 3-D. For l-D, we think of heat flowing along a rod. (If the temperatures do reach a steady state, these will be the same as those found by the method of Section 8.1.)

Figure 8.6 shows a rod of length L with an element of length dx in the interior. No heat leaves or enters the rod through its circumference (it may be insulated) but flows only along the rod. As described in Chapter 6, heat flows into the element from the left at a rate, measured in cal/sec, of

$$-kA\left(\frac{du}{dx}\right).$$

The minus sign is required because du/dx expresses how rapidly temperatures increase with x, whereas the heat always flows from high temperature to low.

The rate at which heat leaves the element is given by a similar equation, but now the temperature gradient must be at the point $x + dx$:

$$-kA\left[\frac{du}{dx} + \frac{d}{dx}\left(\frac{du}{dx}\right)dx\right],$$

in which the gradient term is the gradient at x plus the change in the gradient between x and $x + dx$.

These two relations are precisely those of Section 6.7. Now, however, we do not assume that these two rates are equal, but that their difference is the rate at which heat is stored

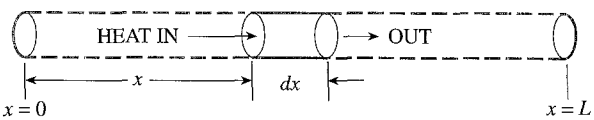


Figure 8.6

* The abbreviation FEA is sometimes used, from *finite-element analysis.*

within the element. This heat that is stored within the element raises its temperature. The rate of increase in the amount of heat that is stored is related to the rate of change in temperature of the element by an equation that involves the volume of the element ($A * dx$, measured in cm^3), the density of the material (ρ, measured in cal/gm), and a property of the material called the heat capacity, [c, measured in cal/(gm * °C)]:

$$\text{rate of increase of heat stored} = c\rho(A\,dx)\frac{du}{dt}.$$

We equate this increase in the rate of heat storage to the difference between the rates at which heat enters and leaves:

$$-kA\left(\frac{\partial u}{\partial x}\right) - \left(-kA\left[\frac{\partial u}{\partial x} + \frac{\partial}{\partial x}\left(\frac{\partial u}{\partial x}\right)dx\right]\right) = c\rho\,(A\,dx)\frac{\partial u}{\partial t}, \tag{8.20}$$

where the derivatives are now partial derivatives because there are two independent variables, x and t. We can simplify Eq. (8.20) to

$$k\left(\frac{\partial^2 u}{\partial x^2}\right) = c\rho\frac{\partial u}{\partial t}. \tag{8.21}$$

If the region is a slab or a three-dimensional object, we have the analogous equation

$$k\,\nabla^2 u = c\rho\frac{\partial u}{\partial t}, \tag{8.22}$$

in which the Laplacian appears.

It may be that the material is not homogeneous and its thermal properties may vary with position. Also, there could be heat generation within the element equal to Q cal/(sec * cm^3). In this more general case we have, in three dimensions,

$$\frac{\partial}{\partial x}\left[k(x,y,z)\frac{\partial u}{\partial x}\right] + \frac{\partial}{\partial y}\left[k(x,y,z)\frac{\partial u}{\partial y}\right] + \frac{\partial}{\partial z}\left[k(x,y,z)\frac{\partial u}{\partial z}\right] + Q(x,y,z)$$
$$= c(x,y,z) * \rho(x,y,z)\frac{\partial u}{\partial t}.$$

Our illustrations will stay with the simpler cases represented by Eqs. (8.21) and (8.22).

In order to solve these equations for unsteady-state heat flow (and they apply as well to diffusion or to any problem where the potential is proportional to the gradient), we need to make the solution agree with specified conditions along the boundary of the region of interest. In addition, because the problems are time dependent, we must begin with specified initial conditions (at $t = 0$) at all points within the region. We might think of these problems as both boundary-value problems with respect to the space variables and as initial-value problems with respect to time.

Solving the Heat Equation

We describe three different ways to solve for temperatures as they vary with time along a rod, the one-dimensional case. All three techniques are similar in that they replace the

space derivative with a central difference. They differ in that different finite-difference quotients are used for the time derivative. We begin with what is called the *explicit method.* We use this forward approximation for the time derivative:

$$\frac{\partial u}{\partial t} \approx \frac{u_i^{j+1} - u_i^j}{\Delta t} \qquad \text{(at point } x_i \text{ and time } t_j\text{)}, \tag{8.23}$$

where we use subscripts to indicate the location and superscripts to indicate the time.* For the derivative with respect to x, we use (at point x_i and time t_j):

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{u_{i+1}^j - 2u_i^j + u_{i-1}^j}{(\Delta x)^2}. \tag{8.24}$$

Observe that we are using a forward difference in Eq. (8.23) but a central difference in Eq. (8.24). From the discussion in Chapter 3, we know that the first has an error of order $O(\Delta t)$, whereas the second has an error of order $O(\Delta x)^2$. This difference in orders has an important consequence, as will be seen.

Substituting these approximations into Eq. (8.21) and solving for u_i^{j+1}, we get

$$u_i^{j+1} = r * (u_{i+1}^j + u_{i-1}^j) + (1 - 2r) * u_i^j, \tag{8.25}$$

where

$$r = \frac{k\,\Delta t}{c\rho(\Delta x)^2}.$$

Equation (8.25) is a way that we can march through time one Δt at a time. For $t = t_1$, we have the u's at t_0 from the initial conditions. At each subsequent time interval, we have the values for the previous time from the last computations. We apply the equation at each point along the rod where the temperature is unknown. (If an end condition involves a temperature gradient, that endpoint is included.)

The use of Eq. (8.25) to compute temperatures as a function of position and time is called the *explicit method* because each subsequent computation is explicitly given from the previous u-values.

An example will clarify the procedure.

EXAMPLE 8.8 Solve for the temperatures as a function of time within a large steel plate that is 2 cm thick. For steel, $k = 0.13$ cal/(sec * cm * °C), $c = 0.11$ cal/(g * °C), and $\rho = 7.8$ g/cm^3. Because the plate is large, neglect lateral flow of heat and consider only the flow perpendicular to the faces of the plate.

Initially, the temperatures within the plate, measured from the top face (where $x = 0$) to the bottom (where $x = 2$) are given by this relation:

$$u(x) = 100x, \quad 0 \le x \le 1; \qquad u(x) = 200 - 100x, \quad 1 \le x \le 2.$$

---

* The x_i are locations of evenly spaced nodes. The t_j are times spaced apart by Δt.

The boundary conditions, both at $x = 0$ and at $x = 2$, are $u = 0°$. Use $\Delta x = 0.25$ so there are eight subdivisions. Number the interior nodes from 1 to 7 so that node 0 is on the top face and node 8 is at the bottom.

The value that we use for Δt depends on the value that we choose for r, the ratio $(k\Delta t)/[c\rho(\Delta x)^2]$. Let us use $r = 0.5$ for a first trial. Doing so greatly simplifies Eq. (8.25). It becomes

$$u_i^{j+1} = 0.5(u_{i+1}^j + u_{i-1}^j). \tag{8.26}$$

(We shall compare the results of this first trial to other trials with different values for r.) With $r = 0.5$, the value of Δt is $rc\rho(\Delta x)^2/k = 0.5(0.11)(7.8)(0.25)^2/0.13 = 0.206$ sec.

We use Eq. (8.26) to compute temperatures at each node for several time steps. When this is done, the results shown in Table 8.4 are obtained. Because the values are symmetrical about the center of the rod, only those for the top half are tabulated, and the values for $x = 0$, which are all $u = 0$, are omitted. Table 8.4 also shows values from the "analytical" solution at $x = 0.5$ and at $x = 1$ from the series solution given by a classical method for solving the problem.

It is apparent from the conditions for this example that the temperatures will eventually reach the steady-state temperatures; at $t = \infty$, u will be 0° everywhere. The values in Table 8.4 are certainly approaching this equilibrium temperature. (All temperatures are within 0.1 of 0.0 after 85 time steps.)

Table 8.4 Computed and analytical temperatures for Example 8.8

		***x* value**					
		0.25	**0.50**		**0.75**	**1.00**	
Time steps	t	**(computed)**	**(comp)**	**(anal)**	**(computed)**	**(comp)**	**(anal)**
0	0	25.00	50.00	50.00	75.00	100.00	100.00
1	0.206	25.00	50.00	49.58	75.00	75.00	80.06
2	0.413	25.00	50.00	47.49	62.50	75.00	71.80
3	0.619	25.00	43.75	44.68	62.50	62.50	65.46
4	0.825	21.88	43.75	41.71	53.13	62.50	60.11
5	1.031	21.88	37.50	38.79	53.13	53.13	55.42
6	1.237	18.75	37.50	35.99	45.31	53.13	51.18
7	1.444	18.75	32.03	33.37	45.31	45.31	47.33
8	1.650	16.02	32.03	30.91	38.67	45.31	43.79
9	1.856	16.02	27.34	28.63	38.67	38.67	40.52
10	2.062	13.67	27.34	26.51	33.01	38.67	37.51
11	2.269	13.67	23.34	24.55	33.01	33.01	34.72
12	2.475	11.67	23.34	22.73	28.17	33.01	32.15
13	2.681	11.67	19.92	21.04	28.17	28.17	29.76
14	2.887	9.96	19.92	19.48	24.05	28.17	27.55

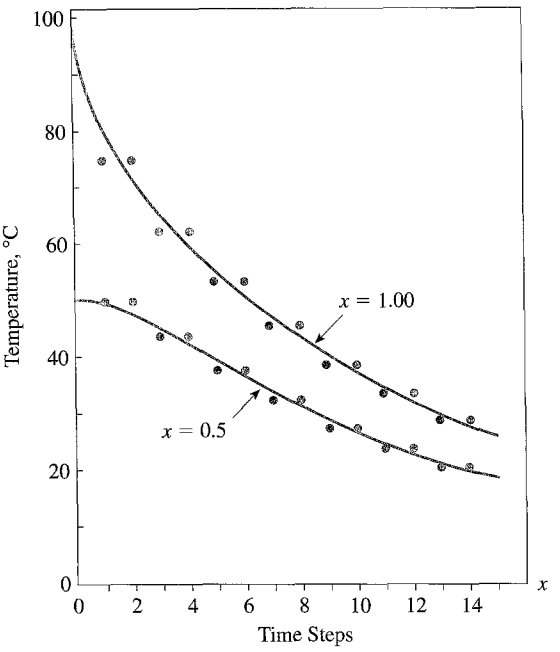


Figure 8.7

The computed values generally follow the analytical but oscillate above and below successive values. This is shown more clearly in Figure 8.7, where the computed temperatures at the center node and at $x = 0.5$ cm are plotted. The curves represent the analytical solution. If the computations are repeated but with two other values of r ($r = 0.4$ and $r = 0.6$), we find an interesting phenomenon. Of course, the values of Δt will change as well. With the smaller value for r, 0.4, the computed results are much more accurate, and the differences from the analytical values are about half as great during the early time steps and become only one-tenth as great after ten time steps. We would expect somewhat better agreement because the time steps are smaller, but the improvement is much greater than this change would cause.

On the other hand, using a value of 0.6 for r results in extremely large errors. In fact, after only eight time steps, some of the calculated values for u are negative, a patently impossible result. Figure 8.8 illustrates this quite vividly. The open circles in the figure are results with $r = 0.6$; the solid points are for $r = 0.4$. The explanation for this behavior is *instability*. The maximum value for r to avoid instability (which is particularly evident for $r = 0.6$) is $r = 0.5$. The oscillation of points about the analytical curve in Figure 8.7 shows incipient instability. Even this value for r is too large if the boundary conditions involve a gradient.

The Crank–Nicolson Method

The reason why there as instability when r is greater than 0.5 in the explicit method is the difference in orders of the finite-difference approximations for the spatial derivative and

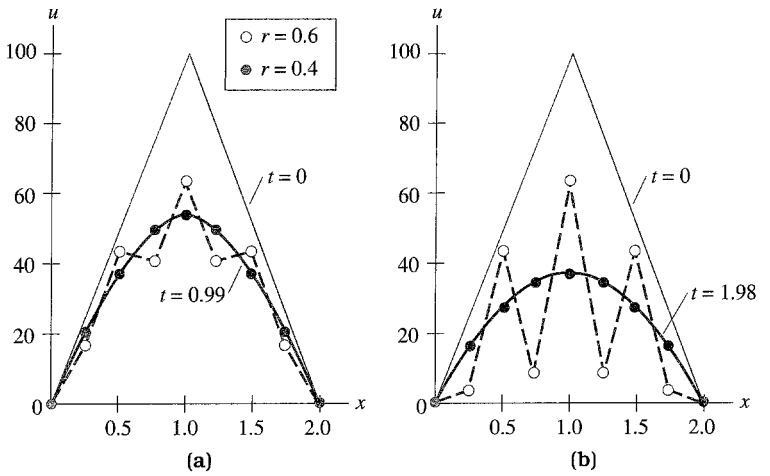


Figure 8.8

the time derivative. The *Crank–Nicolson method* is a technique that makes these finite-difference approximations of the same order.

The difference quotient for the time derivative, $(u_i^{j+1} - u_i^j)/\Delta t$, can be considered a central-difference approximation at the midpoint of the time step. If we do take this as a central-difference approximation, we will need to equate it to a central-difference approximation of the spatial derivative at the same halfway point in the time step, and this we can hope to obtain by averaging two approximations for $\partial^2 u/\partial x^2$, one computed at the start and the other at the end of the time step. So, we write, for

$$\frac{\partial u}{\partial t} = \frac{k}{c\rho}\frac{\partial^2 u}{\partial x^2},$$

this approximation:

$$\frac{u_i^{j+1} - u_i^j}{\Delta t} = \frac{1}{2}\frac{k}{c\rho}\left[\frac{u_{i+1}^j - 2u_i^j + u_{i-1}^j}{(\Delta x)^2} + \frac{u_{i+1}^{j+1} - 2u_i^{j+1} + u_{i-1}^{j+1}}{(\Delta x)^2}\right],$$

which we solve for the u-values at the end of the time step to give

$$\begin{aligned} -ru_{i+1}^{j+1} + (2 + r)u_i^{j+1} - ru_{i-1}^{j+1} &= ru_{i+1}^j + (2 - r)u_i^j + ru_{i-1}^j, \\ r &= \frac{k\Delta t}{c\rho(\Delta x)^2}. \end{aligned} \tag{8.27}$$

Equation (8.27) is the Crank–Nicolson formula, and using it involves solving a set of simultaneous equations, because the equation for u_i^{j+1} includes two adjacent u-values at $t = t^{j+1}$. Hence, this is an *implicit method*. Fortunately, the coefficient matrix is tridiagonal. A most important advantage of the method is that it is stable for any value for r, although smaller values usually give better accuracy. This next example illustrates the method.

EXAMPLE 8.9 Solve Example 8.8, but now use the Crank–Nicolson method. Compare the results with $r = 0.5$ and with $r = 1.0$ to the analytical values.

Employing Eq. (8.27) gives the results shown in Table 8.5 for the centerline temperatures with $r = 0.5$ and in Table 8.6 for the centerline temperatures with $r = 1.0$. The error columns are the differences between the computed temperatures and those from the series solution. In Table 8.5, these range from 2.0% to 2.7% of the analytical values, whereas in Table 8.6, they range from 1.0% to 2.5%. One would expect the errors with $r = 0.5$ to be smaller, but this is not the case. Both sets of computations are more accurate than those in Table 8.4, where the explicit method was used with $r = 0.5$.

The Theta Method – A Generalization

In the Crank–Nicolson method, we interpret the finite-difference approximation to the time derivative as a central difference at the midpoint of the time interval. In the *theta method,* we make a more general statement by interpreting this approximation to apply at some other point within the time interval. If we interpret it to apply at a fraction θ of Δt, we then equate the time-derivative approximation to a weighted average of the spatial derivatives at the beginning and end of the time interval, giving this relation:

$$\frac{u_i^{j+1} - u_i^j}{\Delta t} = \left(\frac{k}{c\rho}\right)\left[\frac{(1-\theta)(u_{i+1}^j - 2u_i^j + u_{i-1}^j)}{(\Delta x)^2} + \frac{\theta\,(u_{i+1}^{j+1} - 2u_i^{j+1} + u_{i-1}^{j+1})}{(\Delta x)^2}\right].$$

Observe that using $\theta = 0.5$ gives the Crank–Nicolson method, whereas using $\theta = 0$ gives the explicit method. If we use $\theta = 1$, the theta method is often called the *implicit method.* For $\theta = 1$, the analog of Eq. (8.27) is

$$-ru_{i+1}^{j+1} + (1 + 2r)u_i^{j+1} - ru_{i-1}^{j+1} = u_i^j, \qquad r = \frac{k\Delta t}{c\rho(\Delta x)^2}.$$

Table 8.5 Centerline temperatures with Crank–Nicolson Method, $r = 0.5$

Time steps	t	u-values	Error
0	0	100.00	—
1	0.206	82.32	2.26
2	0.413	73.48	1.68
3	0.619	66.86	1.40
4	0.825	61.34	1.23
5	1.031	56.52	1.10
6	1.237	52.21	1.03
7	1.444	48.30	0.97
8	1.650	44.71	0.92
9	1.856	41.40	0.88
10	2.062	38.36	0.85

Table 8.6 Centerline temperatures with Crank–Nicolson Method, $r = 1.0$

Time steps	t	u-values	Error
0	0	100.00	—
1	0.413	71.13	0.67
2	0.825	61.53	1.42
3	1.237	51.97	0.79
4	1.650	44.67	0.88
5	2.062	38.29	0.78
6	2.475	32.88	0.73
7	2.887	28.23	0.68
8	3.300	24.23	0.61

For any value of θ, the typical equation is

$$-ru_{i+1}^{j+1} + (1 + 2r\theta)u_i^{j+1} - ru_{i-1}^{j+1} = r(1 - \theta)u_{i+1}^j + [1 - 2r(1 - \theta)]u_i^j + r(1 - \theta)u_{i-1}^j. \quad (8.28)$$

What value is best for θ? Burnett (1987) suggests that $\theta = \frac{2}{3}$ is nearly optimal, but he points out that a case can be made for using $\theta = 0.878$. This next example compares the use of these two values.

EXAMPLE 8.10 Solve Example 8.8 by the theta method with $\theta = \frac{2}{3}$, 0.878, and 1.0. Compare these to results from the Crank–Nicolson and explicit methods.

Using Eq. (8.28), computations were made for ten time steps. Table 8.7 shows how the values at the centerline, $x = 1.0$ differ from the analytical values. It is interesting to observe that, for this problem, the Crank–Nicolson results ($\theta = 0.5$) have smaller errors than those with larger values for θ. Even the results from the explicit method ($\theta = 0$) are better than those with $\theta = 1.0$ (although the explicit values oscillate around the analytical). This suggests that there is an optimal value for θ less than $\frac{2}{3}$ and greater than zero. We leave this determination as an exercise, as well as the comparison at other values for x. We also leave as an exercise to find if there is an optimal value in other problems.

Stability Considerations

We have seen in our examples that when the ratio $k\Delta t/c\rho(\Delta x)^2$ is greater than 0.5, the explicit method is unstable. Crank–Nicolson and the implicit methods do not have such a limitation. We now look at this more analytically. We also discuss the *convergence* of the methods.

Table 8.7 Comparisons of results from the theta method, $r = 0.5$

Time steps	Errors in computed centerline temperatures, θ-value				
	2/3	**0.878**	**1.0**	**0.5**	**0.0**
1	3.57	4.88	5.51	2.26	5.06
2	2.48	3.55	4.15	1.68	3.20
3	1.98	2.79	3.28	1.40	2.96
4	1.71	2.37	2.78	1.23	2.39
5	1.53	2.11	2.46	1.10	2.29
6	1.43	1.97	2.29	1.03	1.95
7	1.35	1.86	2.16	0.97	2.02
8	1.30	1.80	2.09	0.92	1.52
9	1.27	1.76	2.04	0.88	1.85
10	1.23	1.72	2.00	0.85	1.16

By *convergence,* we mean that the results of the method approach the analytical values as Δt and Δx both approach zero. By *stability,* we mean that errors made at one stage of the calculations do not cause increasingly large errors as the computations are continued, but rather will eventually damp out.

We will first discuss convergence, limiting ourselves to the simple case of the unsteady-state heat-flow equation in one dimension:*

$$\frac{\partial U}{\partial t} = \frac{k}{c\rho}\frac{\partial^2 U}{\partial x^2}. \tag{8.29}$$

We will use the symbol U to represent the exact solution to Eq. (8.29), and u to represent the numerical solution. At the moment, we assume that u is free of round-off errors, so the only difference between U and u is the error made by replacing Eq. (8.29) by the difference equation. Let $e_i^j = U_i^j - u_i^j$, at the point $x = x_i$, $t = t_j$. By the explicit method, Eq. (8.29) becomes

$$u_i^{j+1} = r(u_{i+1}^j + u_{i-1}^j) + (1 - 2r)u_i^j, \tag{8.30}$$

where $r = k\,\Delta t/c\rho(\Delta x)^2$. Substituting $u = U - e$ into Eq. (8.30), we get

$$e_i^{j+1} = r(e_{i+1}^j + e_{i-1}^j) + (1 - 2r)e_i^j - r(U_{i+1}^j + U_{i-1}^j) - (1 - 2r)U_i^j + U_i^{j+1}. \tag{8.31}$$

By using Taylor-series expansions, we have

$$U_{i+1}^j = U_i^j + \left(\frac{\partial U}{\partial x}\right)_{i,j}\Delta x + \frac{(\Delta x)^2}{2}\frac{\partial^2 U(\xi_1, t_j)}{\partial x^2}, \qquad x_i < \xi_1 < x_{i+1},$$

$$U_{i-1}^j = U_i^j - \left(\frac{\partial U}{\partial x}\right)_{i,j}\Delta x + \frac{(\Delta x)^2}{2}\frac{\partial^2 U(\xi_2, t_j)}{\partial x^2}, \qquad x_{i-1} < \xi_2 < x_i,$$

$$U_i^{j+1} = U_i^j + \Delta t\frac{\partial U(x_i, \eta)}{\partial t}, \qquad t_j < \eta < t_{j+1}.$$

Substituting these into Eq. (8.31) and simplifying, remembering that $r(\Delta x)^2 = k\,\Delta t/c\rho$, we get

$$e_i^{j+1} = r(e_{i+1}^j + e_{i-1}^j) + (1 - 2r)e_i^j + \Delta t\left[\frac{\partial U(x_i, \eta)}{\partial t} - \frac{k}{c\rho}\frac{\partial^2 U(\xi, t_j)}{\partial x^2}\right],$$
$$t_j \le \eta \le t_{j+1}, \qquad x_{i-1} \le \xi \le x_{i+1}. \tag{8.32}$$

* We could have treated the simpler equation $\partial U/\partial T = \partial^2 U/\partial X^2$ without loss of generality, because with the change of variables—$X = \sqrt{c\rho}\,x$, $T = kt$—the two equations are seen to be identical.

Let E^j be the magnitude of the maximum error in the row of calculations for $t = t_j$, and let $M > 0$ be an upper bound for the magnitude of the expression in brackets in Eq. (8.32). If $r \leq \frac{1}{2}$, all the coefficients in Eq. (8.32) are positive (or zero) and we may write the inequality

$$|e_i^{j+1}| \leq 2rE^j + (1 - 2r)E^j + M\Delta t = E^j + M\Delta t.$$

This is true for all the e_i^{j+1} at $t = t_{j+1}$, so

$$E^{j+1} \leq E^j + M\,\Delta t.$$

This is true at each time step,

$$E^{j+1} \leq E^j + M\,\Delta t \leq E^{j-1} + 2M\,\Delta t \leq \cdots \leq E^0 + (j+1)M\,\Delta t = E^0 + Mt_{j+1} = Mt_{j+1},$$

because E^0, the errors at $t = 0$, is zero, as U is given by the initial conditions.

Now, as $\Delta x \to 0$, $\Delta t \to 0$ if $k\,\Delta t/c\rho(\Delta x)^2 \leq \frac{1}{2}$, and $M \to 0$, because, as both Δx and Δt get smaller,

$$\left[\frac{\partial U(x_i, \eta)}{\partial t} - \frac{k}{c\rho} * \frac{\partial^2 U(\xi, t_j)}{\partial x^2}\right] \to \left(\frac{\partial U}{\partial t} - \frac{k}{c\rho} * \frac{\partial^2 U}{\partial x^2}\right)_{i,j} = 0.$$

This last is by virtue of Eq. (8.29), of course. Consequently, we have shown that the explicit method is convergent for $r \leq \frac{1}{2}$, because the errors approach zero as Δt and Δx are made smaller.

For the solution to the heat-flow equation by the Crank–Nicolson method, the analysis of convergence may be made by similar methods. The treatment is more complicated, but it can be shown that each E^{j+1} is no greater than a finite multiple of E^j plus a term that vanishes as both Δx and Δt become small, and this is independent of r. Hence, because the initial errors are zero, the finite-difference solution approaches the analytical solution as $\Delta t \to 0$ and $\Delta x \to 0$, requiring only that r stay finite. This is also true for the θ method whenever $0.5 \leq \theta \leq 1$.

We begin our discussion of stability with a numerical example. Because the heat-flow equation is linear, if two solutions are known, their sum is also a solution. We are interested in what happens to errors made in one line of the computations as the calculations are continued, and because of the additivity feature, the effect of a succession of errors is just the sum of the effects of the individual errors. We follow, then, a single error,* which most likely occurred due to round off. If this single error does not grow in magnitude, we will call the method *stable,* because then the cumulative effect of all errors affects the later calculations no more than a linear combination of the previous errors would. (Because round-off errors are both positive and negative, we can expect some cancellation.)

Table 8.8 illustrates the principle. We have calculated for the simple case where the boundary conditions are fixed, so that the errors at the endpoints are zero. We assume that a single

* A computation made assuming that each of the interior points has an error equal to e at $t = t_1$ demonstrates the effect more rapidly.

Table 8.8 Propagation of errors—explicit method

t	Endpoint x_1	x_2	x_3	x_4	Endpoint x_5
t_0	0	0	0	0	0
t_1	0	e	0	0	0
t_2	0	0	$0.50e$	0	0
t_3	0	$0.25e$	0	$0.25e$	0
t_4	0	0	$0.25e$	0	0
t_5	0	$0.125e$	0	$0.125e$	0
t_6	0	0	$0.125e$	0	0
t_7	0	$0.062e$	0	$0.062e$	0
t_8	0	0	$0.062e$	0	0

error of size e occurs at $t = t_1$ and $x = x_2$. The explicit method, $k\Delta t/c\rho(\Delta x)^2 = \frac{1}{2}$, was used. The original error quite obviously dies out. As an exercise, it is left to the student to show that with $r > 0.5$, errors have an increasingly large effect on later computations. Table 8.9 shows that errors damp out for the Crank–Nicolson method with $r = 1$ even more rapidly than in the explicit method with $r = 0.5$. The errors with the implicit method also die out with $r = 1$, more rapidly than with the explicit method but less rapidly than with Crank–Nicolson.

A More Analytical Argument

To discuss stability in a more analytical sense, we need some material from linear algebra. In Chapter 6, we discussed eigenvalues and eigenvectors of a matrix. We recall that for the matrix A and vector x, if

$$Ax = \lambda x,$$

then the scalar λ is an eigenvalue of A and x is the corresponding eigenvector. If the N eigenvalues of the $N \times N$ matrix A are all different, then the corresponding N eigenvectors

Table 8.9 Propagation of errors—Crank–Nicolson method

t	x_1	x_2	x_3	x_4	x_5
t_0	0	0	0	0	0
t_1	0	e	0	0	0
t_2	0	$0.071e$	$0.286e$	$0.071e$	0
t_3	0	$0.092e$	$0.082e$	$0.092e$	0
t_4	0	$0.036e$	$0.064e$	$0.036e$	0
t_5	0	$0.024e$	$0.030e$	$0.024e$	0
t_6	0	$0.012e$	$0.018e$	$0.012e$	0
t_7	0	$0.007e$	$0.009e$	$0.007e$	0
t_8	0	$0.004e$	$0.005e$	$0.004e$	0

are linearly independent, and any N-component vector can be written uniquely in terms of them.

Consider the unsteady-state heat-flow problem with fixed boundary conditions. Suppose we subdivide into $N + 1$ subintervals so there are N unknown values of the temperature being calculated at each time step. Think of these N values as the components of a vector. Our algorithm for the explicit method (Eq. 8.25) can be written as the matrix equation*

$$\begin{bmatrix} u_1^{j+1} \\ u_2^{j+1} \\ \vdots \\ u_N^{j+1} \end{bmatrix} = \begin{bmatrix} (1-2r) & r & & & \\ r & (1-2r) & r & & \\ & & \ddots & & \\ & & & r & (1-2r) \end{bmatrix} \begin{bmatrix} u_1^j \\ u_2^j \\ \vdots \\ u_N^j \end{bmatrix}, \tag{8.33}$$

or

$$u^{j+1} = Au^j,$$

where A represents the coefficient matrix and u^j and u^{j+1} are the vectors whose N components are the successive calculated values of temperature. The components of u^0 are the initial values from which we begin our solution. The successive rows of our calculations are

$$\begin{aligned} u^1 &= Au^0, \\ u^2 &= Au^1 = A^2u^0, \\ &\vdots \\ u^j &= Au^{j-1} = A^2u^{j-2} = \cdots = A^ju^0. \end{aligned}$$

(Here the superscripts on the A's are exponents; on the vectors they indicate time.)

Suppose that errors are introduced into u^0, so that it becomes $\bar{u}^0$. We will follow the effects of this error through the calculations. The successive lines of calculation are now

$$\bar{u}^j = A\bar{u}^{j-1} = \cdots = A^j\bar{u}^0.$$

Let us define the vector e^j as $u^j - \bar{u}^j$ so that e^j represents the errors in u^j caused by the errors in $\bar{u}^0$. We have

$$e^j = u^j - \bar{u}^j = A^ju^0 - A^j\bar{u}^0 = A^je^0. \tag{8.34}$$

This shows that errors are propagated by using the same algorithm as that by which the temperatures are calculated, as was implicitly assumed earlier in this section.

---

* A change of variable is required to give boundary conditions of $u = 0$ at each end. This can always be done for fixed end conditions.

Now the N eigenvalues of A are distinct (see below) so that its N eigenvectors x_1, $x_2, \ldots, x_N$ are independent, and

$$\begin{aligned} Ax_1 &= \lambda_1 x_1, \\ Ax_2 &= \lambda_2 x_2, \\ &\vdots \\ Ax_N &= \lambda_N x_N. \end{aligned}$$

We now write the error vector e^0 as a linear combination of the x_i:

$$e^0 = c_1 x_1 + c_2 x_2 + \cdots + c_N x_N,$$

where the c's are constants. Then e^1 is, in terms of the x_i,

$$e^1 = Ae^0 = \sum_{i=1}^{N} Ac_i x_i = \sum_{i=1}^{N} c_i A x_i = \sum_{i=1}^{N} c_i \lambda_i x_i,$$

and for e^2,

$$e^2 = Ae^1 = \sum_{i=1}^{N} Ac_i \lambda_i x_i = \sum_{i=1}^{N} c_i \lambda_i^2 x_i.$$

(Again, the superscripts on vectors indicate time; on λ they are exponents.) After j steps, Eq. (8.34) can be written

$$e^j = \sum_{i=1}^{N} c_i \lambda_i^j x_i.$$

If the magnitudes of all of the eigenvalues are less than or equal to unity, errors will not grow as the computations proceed; that is, the computational scheme is stable. This then is the analytical condition for stability: that the largest eigenvalue of the coefficient matrix for the algorithm be one or less in magnitude.

The eigenvalues of matrix A (Eq. 8.33) can be shown to be

$$1 - 4r \sin^2 \frac{n\pi}{2(N+1)}, \qquad n = 1, 2, \ldots, N$$

(note that they are all distinct). We will have stability for the explicit scheme if

$$-1 \leq 1 - 4r \sin^2 \frac{n\pi}{2(N+1)} \leq 1.$$

The limiting value of r is given by

$$-1 \leq 1 - 4r \sin^2 \frac{n\pi}{2(N+1)}$$

$$r \leq \frac{\frac{1}{2}}{\sin^2\left(\dfrac{n\pi}{2(N+1)}\right)}.$$

Hence, if $r \leq \frac{1}{2}$, the explicit scheme is stable.

The Crank–Nicolson scheme, in matrix form, is

$$\begin{bmatrix} (2+2r) & -r & & & \\ -r & (2+2r) & -r & & \\ & & \vdots & & \\ & & & -r & (2+2r) \end{bmatrix} \begin{bmatrix} u_1^{j+1} \\ u_2^{j+1} \\ \vdots \\ u_N^{j+1} \end{bmatrix} = \begin{bmatrix} (2-2r) & r & & & \\ r & (2-2r) & r & & \\ & & \vdots & & \\ & & & r & (2-2r) \end{bmatrix} \begin{bmatrix} u_1^{j} \\ u_2^{j} \\ \vdots \\ u_N^{j} \end{bmatrix},$$

or

$$Au^{j+1} = Bu^j.$$

We can write

$$u^{j+1} = (A^{-1}B)u^j,$$

so that stability is given by the magnitudes of the eigenvalues of $A^{-1}B$. These are

$$\frac{2 - 4r\sin^2\left(\dfrac{n\pi}{2(N-1)}\right)}{2 + 4r\sin^2\left(\dfrac{n\pi}{2(N-1)}\right)}, \qquad n = 1, 2, \ldots, N.$$

Clearly, all the eigenvalues are no greater than one in magnitude for any positive value of r. A similar argument shows that the implicit method is also unconditionally stable.

The Heat Equation in Two or Three Dimensions

In dimensions greater than one, the equation that we are to solve is

$$\frac{\partial u}{\partial t} = \frac{k}{c\rho}\nabla^2 u. \tag{8.35}$$

We will apply finite-difference approximations to the derivatives as we did in 1-D. We show how a typical example is solved.

Suppose we have a rectangular region whose edges fit to evenly spaced nodes. If we replace the right-hand side of Eq. (8.35) with central-difference approximations, where $\Delta x = \Delta y = h$, and $r = k\,\Delta t/(c\rho h^2)$, the explicit scheme becomes

$$u_{i,j}^{k+1} - u_{i,j}^k = r(u_{i+1,j}^k - 2u_{i,j}^k + u_{i-1,j}^k + u_{i,j+1}^k - 2u_{i,j}^k + u_{i,j-1}^k)$$

or

$$u_{i,j}^{k+1} = r(u_{i+1,j}^k + u_{i-1,j}^k + u_{i,j+1}^k + u_{i,j-1}^k) + (1-4r)u_{i,j}^k.$$

In this scheme, stability requires that the value of r be $\frac{1}{4}$ or less in the simple case of Dirichlet boundary conditions. (Note that this corresponds again to the numerical value that gives a particularly simple formula.) In the more general case with $\Delta x \neq \Delta y$, the criterion is

$$\frac{k\,\Delta t}{c\rho[(\Delta x)^2 + (\Delta y)^2]} \leq \frac{1}{8}.$$

The analogous equation in three dimensions, with equal grid spacing each way, has the coefficient $(1 - 6r)$, and $r \leq \frac{1}{6}$ is required for convergence and stability.

The difficulty with the use of the explicit scheme is that the restrictions on Δt require inordinately many rows of calculations. One then looks for a method in which Δt can be made larger without loss of stability. In one dimension, the Crank–Nicolson method was such a method. In the 2-D case, using averages of central-difference approximations to give $\partial^2 u/\partial x^2$ and $\partial^2 u/\partial y^2$ at the midvalue of time, we get

$$u_{i,j}^{k+1} - u_{i,j}^{k} = \frac{r}{2}[u_{i+1,j}^{k+1} - 2u_{i,j}^{k+1} + u_{i-1,j}^{k+1} + u_{i+1,j}^{k} - 2u_{i,j}^{k} + u_{i-1,j}^{k} + u_{i,j+1}^{k+1} - 2u_{i,j}^{k+1} + u_{i,j-1}^{k+1} + u_{i,j+1}^{k} - 2u_{i,j}^{k} + u_{i,j-1}^{k}].$$

The problem now is that a set of $(M)(N)$ simultaneous equations must be solved at each time step, where M is the number of unknown values in the x-direction and N in the y-direction. Furthermore, the coefficient matrix is no longer tridiagonal, so the solution to each set of equations is slower and memory space to store the elements of the matrix may be exorbitant.

The advantage of a tridiagonal matrix is retained in the alternating direction implicit scheme (A.D.I.) proposed by Peaceman and Rachford (1955). It is widely used in modern computer programs for the solution of parabolic partial-differential equations. We discussed the A.D.I. method in Section 8.1 applied to elliptic equations. For parabolic equations, we approximate $\nabla^2 u$ by adding a central-difference approximation to $\partial^2 u/\partial x^2$ written at the beginning of the time interval to a similar expression for $\partial^2 u/\partial y^2$ written at the end of the time interval. We will use subscripts L, R, A, and B to indicate nodes to the left, right, above, and below the central node, respectively, where $u = u_0$. We then have

$$u_0^{j+1} - u_0^{j} = r[u_L^{j} - 2u_0^{j} + u_R^{j}] + r[u_A^{j+1} - 2u_0^{j+1} + u_B^{j+1}], \tag{8.36}$$

where $r = k\,\Delta t/c\rho\Delta^2$ and $\Delta = \Delta x = \Delta y$. The obvious bias in this formula is balanced by reversing the order of the second derivative approximations in the next time span:

$$u_0^{j+2} - u_0^{j+1} = r[u_L^{j+2} - 2u_0^{j+2} + u_R^{j+2}] + r[u_A^{j+1} - 2u_0^{j+1} + u_B^{j+1}]. \tag{8.37}$$

Observe that in using Eq. (8.36), we make a vertical traverse through the nodes, computing new values for each column of nodes. Similarly, in using Eq. (8.37) we make a horizontal traverse, computing new values row by row. In effect, we consider u_L and u_R as fixed when we do a vertical traverse; we consider u_A and u_B as fixed for horizontal traverses.

EXAMPLE 8.11 A square plate of steel is 8 in. wide and 6 in. high. Initialty, all points on the plate are at 50°. The edges are suddenly brought to the temperatures shown in Figure 8.9 and held at

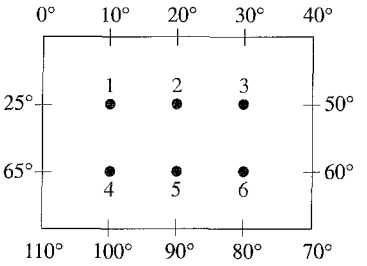


Figure 8.9

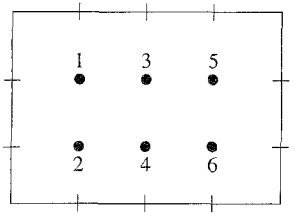


Figure 8.10

these temperatures. Trace the history of temperatures at nodes spaced 2 in. apart using the A.D.I. method, assuming that heat flows only in the x- and y-directions.

Figure 8.9 shows a numbering system for the internal nodes, all of which start at 50°, as well as the temperatures at boundary nodes.

Using Eq. (8.36), the typical equation for a vertical traverse is

$$-ru_A^{j+1} + (1 + 2r)u_0^{j+1} - ru_B^{j+1} = (ru_L + (1 - 2r)u_0 + ru_R)^j.$$

If we use this equation and the numbering system of Figure 8.9 to set up the equations for a vertical traverse, we do not get the tridiagonal system that we desire, but we do if the nodes are renumbered as shown in Figure 8.10. To keep track of the different numbering systems, we will use v for temperatures when a vertical traverse is made (numbered as in Fig. 8.10) and u when a horizontal traverse is made (numbered as in Fig 8.9).

This is the set of equations for a vertical traverse:

$$\begin{bmatrix} (1+2r) & -r & & & & \\ -r & (1+2r) & & & & \\ & & (1+2r) & -r & & \\ & & -r & (1+2r) & & \\ & & & & (1+2r) & -r \\ & & & & -r & (1+2r) \end{bmatrix} \begin{Bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{Bmatrix} = \begin{Bmatrix} r(25) + (1-2r)u_1 + ru_2 + r(10) \\ r(65) + (1-2r)u_4 + ru_5 + r(100) \\ ru_1 + (1-2r)u_2 + ru_3 + r(20) \\ ru_4 + (1-2r)u_5 + ru_6 + r(90) \\ ru_2 + (1-2r)u_3 + r(50) + r(30) \\ ru_5 + (1-2r)u_6 + r(60) + r(80) \end{Bmatrix}.$$

When we apply Eq. (8.36) to get a set of equations for a horizontal traverse, we get (the dashed lines show they break into subsets)

$$\begin{bmatrix} (1+2r) & -r & & & & \\ -r & (1+2r) & -r & & & \\ & -r & (1+2r) & & & \\ & & & (1+2r) & -r & \\ & & & -r & (1+2r) & -r \\ & & & & -r & (1+2r) \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \end{Bmatrix} = \begin{Bmatrix} r(10) + (1-2r)v_1 + rv_2 + r(25) \\ r(20) + (1-2r)v_3 + rv_4 \\ r(30) + (1-2r)v_5 + rv_6 + r(50) \\ rv_1 + (1-2r)v_2 + r(100) + r(65) \\ rv_3 + (1-2r)v_4 + r(90) \\ rv_5 + (1-2r)v_6 + r(80) + r(60) \end{Bmatrix}.$$

A value must be specified for r. Small r's give better accuracy but smaller Δt's, so more time steps are required to compute the history. If we take $r = 1$, Δt is 26.4 sec.

The first vertical traverse gives results for $t = 26.4$ sec. We get the first set of v's from

$$\begin{bmatrix} 3 & -1 & & & & \\ -1 & 3 & & & & \\ & & 3 & -1 & & \\ & & -1 & 3 & & \\ & & & & 3 & -1 \\ & & & & -1 & 3 \end{bmatrix} \begin{Bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{Bmatrix} = \begin{Bmatrix} 25 - (1)50 + 50 + 10 \\ 65 - (1)50 + 50 + 100 \\ 50 - (1)50 + 50 + 20 \\ 50 - (1)50 + 50 + 90 \\ 50 - (1)50 + 50 + 30 \\ 50 - (1)50 + 60 + 80 \end{Bmatrix} = \begin{Bmatrix} 35 \\ 165 \\ 70 \\ 140 \\ 80 \\ 140 \end{Bmatrix}.$$

Solving, we get these values:

$$\{33.75 \quad 66.25 \quad 43.75 \quad 61.25 \quad 47.5 \quad 62.5\}.$$

These values are used to build the right-hand sides for the next computations, a horizontal traverse, getting these equations for $t = 52.8$ sec:

$$\begin{bmatrix} 3 & -1 & & & & \\ -1 & 3 & -1 & & & \\ & -1 & 3 & & & \\ & & & 3 & -1 & \\ & & & -1 & 3 & -1 \\ & & & & -1 & 3 \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \end{Bmatrix} = \begin{Bmatrix} 10 - (1)33.75 + 66.25 + 25 \\ 20 - (1)43.75 + 61.25 \\ 30 - (1)47.5 + 62.5 + 50 \\ 33.75 - (1)66.25 + 100 + 65 \\ 43.75 - (1)61.25 + 90 \\ 47.5 - (1)62.5 + 80 + 60 \end{Bmatrix} = \begin{Bmatrix} 67.5 \\ 37.5 \\ 95 \\ 132.5 \\ 72.5 \\ 125 \end{Bmatrix},$$

which have the solution (a set of u's)

$$\{35.595 \quad 39.286 \quad 44.762 \quad 66.786 \quad 67.857 \quad 64.286\}.$$

We continue by alternating between vertical and horizontal traverses to get the results shown in Table 8.10. This also shows the steady-state temperatures that are reached after a long time. The steady-state temperatures could have been computed by the methods of Section 8.1. We observe that the A.D.I. algorithm for steady-state temperatures is essentially identical to what we have seen here.

The compensation of errors produced by this alternation of direction gives a scheme that is convergent and stable for all values of r, although accuracy requires that r not be too large. The 3-D analog alternates three ways, returning to each of the three formulas after every third step. [Unfortunately, the 3-D case is not stable for all fixed values of $r > 0$. A variant due to Douglas (1962) is unconditionally stable, however.] When the nodes are renumbered, in each case tridiagonal coefficient matrices result.

Note that the equations can be broken up into two independent subsets, corresponding to the nodes in each column or row. (See the first set of equations of Example 8.11.) This is always true in the A.D.I. method; each row gives a set independent of the equations from the other rows. For columns, the same thing occurs. For very large problems, this is important.

Table 8.10 Results for Example 8.11 using the A.D.I. method

```
AT START, TEMPS ARE
  0.0000      10.0000      20.0000      30.0000      40.0000
 25.0000      50.0000      50.0000      50.0000      50.0000
 65.0000      50.0000      50.0000      50.0000      60.0000
110.0000     100.0000      90.0000      80.0000      70.0000
AFTER ITERATION 1 TIME = 26.4 - VALUES ARE
  0.0000      10.0000      20.0000      30.0000      40.0000
 25.0000      33.7500      43.7500      47.5000      50.0000
 65.0000      66.2500      61.2500      62.5000      60.0000
110.0000     100.0000      90.0000      80.0000      70.0000
AFTER ITERATION 2 TIME = 52.8 - VALUES ARE
  0.0000      10.0000      20.0000      30.0000      40.0000
 25.0000      35.5952      69.2857      44.7619      50.0000
 65.0000      66.7857      67.8571      64.2857      60.0000
110.0000     100.0000      90.0000      80.0000      70.0000
AFTER ITERATION 3 TIME = 79.2 - VALUES ARE
  0.0000      10.0000      20.0000      30.0000      40.0000
 25.0000      35.2679      42.0536      45.8929      50.0000
 65.0000      67.1131      65.0893      63.1548      60.0000
110.0000     100.0000      90.0000      80.0000      70.0000
AFTER ITERATION 4 TIME = 105.6 - VALUES ARE
  0.0000      10.0000      20.0000      30.0000      40.0000
 25.0000      36.2443      41.8878      46.3832      50.0000
 65.0000      66.1366      65.2551      62.6644      60.0000
110.0000     100.0000      90.0000      80.0000      70.0000
STEADY-STATE TEMPERATURES:
  0.0000      10.0000      20.0000      30.0000      40.0000
 25.0000      35.8427      41.8323      46.1760      50.0000
 65.0000      66.5383      65.3106      62.8716      60.0000
110.0000     100.0000      90.0000      80.0000      70.0000
```

because it permits the ready overlay of main memory in solving the independent sets. Observe also that each subset can be solved at the same time by parallel processors.

Regions Not Fitted with a Uniform Grid

As discussed in Section 8.1, it is possible to place nodes unevenly and approximate the space derivatives differently, as in Eq. (8.19). Or we might use a different coordinate system (polar or spherical coordinates, for example). However, the most frequently used procedure in such a case is the finite-element method of Chapter 9.

8.3 Hyperbolic Equations

The third class of partial-differential equations, the hyperbolic, is time dependent. They describe vibrations within objects and especially how waves are propagated. Because of this, they are called *wave equations*.

The simplest of the wave equations is that for a vibrating string, the 1-D situation. Another example is that of waves traveling along the length of a long, narrow trough. In 2-D, you might imagine a drum head that is set to vibrating by the musician. The 3-D case is harder to visualize; one could think of a cherry suspended within a bowl of transparent gelatin that moves when the container is tapped with a spoon. In all cases, we want to model the motion and, in the real world, that motion decreases with time due to frictional forces that oppose the motion.

The Vibrating String

We can develop the 1-D wave equation, an example of hyperbolic partial-differential equations, by considering the oscillations of a taut string stretched between two fixed endpoints. Figure 8.11 shows the string with displacements from the straight line between the endpoints greatly exaggerated. The figure shows an element of the string of length dx between points A and B. We use u for the displacements, measured perpendicularly from the straight line between the ends of the string. We focus our attention on the element of the string in Figure 8.11. It is shown enlarged in Figure 8.12, which also shows the angles, α_A and α_B, between the ends of element and the horizontal. (The bending of the element between points A and B is exaggerated as are the displacements.) The figure also indicates that the tension in the stretched string is a force, T. Taking the upward direction as positive, we can write, for the upward forces at each end of the element (these are the vertical components of the tensions).

Upward force at point $A = -T\sin(\alpha_A)$,

Upward force at point $B = T\sin(\alpha_B)$.

Remembering that Figure 8.12 has displacements and angles greatly exaggerated, the tangents of these angles are essentially equal to the sines. We then can write

$$\text{Upward force at point } A = -T\tan(\alpha_A) = -T\left(\frac{\partial u}{\partial x}\right)_A,$$

$$\text{Upward force at point } B = T\tan(\alpha_B) = T\left(\frac{\partial u}{\partial x}\right)_B = T\left[\left(\frac{\partial u}{\partial x}\right)_A + \frac{\partial}{\partial x}\left(\frac{\partial u}{\partial x}\right)dx\right].$$

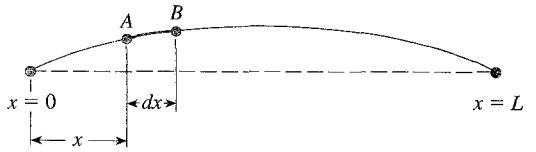


Figure 8.11

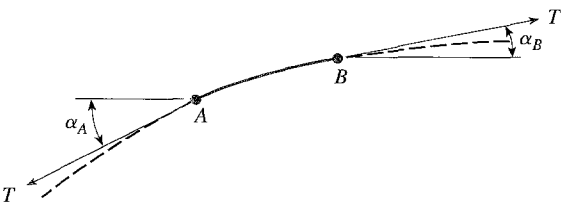


Figure 8.12

The net force acting on the element then is

$$T\left(\frac{\partial^2 u}{\partial x^2}\right) dx.$$

Now, using Newton's law, we equate the force to mass × acceleration (in the vertical direction). Our simplifying assumptions permit us to use $w\,dx$ as the weight (w is the weight per unit length), so

$$\left(T\frac{\partial^2 u}{\partial x^2}\right) dx = \left(w\frac{dx}{g}\right)\left(\frac{\partial^2 u}{\partial t^2}\right) \quad \text{or} \quad \frac{\partial^2 u}{\partial t^2} = \left(\frac{Tg}{w}\right)\frac{\partial^2 u}{\partial x^2}. \tag{8.38}$$

As pointed out in Section 8.1. when Eq. (8.38) is compared to the general form of second-order partial-differential equations, we see that $A = 1$, $B = 0$, and $C = -Tg/w$, and so this falls in the class of hyperbolic equations.

If we have a stretched membrane (like a drum head) instead of a string, the governing equation is

$$\frac{\partial^2 u}{\partial t^2} = \left(\frac{Tg}{w}\right)\nabla^2 u. \tag{8.39}$$

The solution to Eq. (8.38) or Eq. (8.39) must satisfy given boundary conditions along the boundary of the region of interest as well as given initial conditions at $t = 0$. Because the problem is of second order with respect to t, these initial conditions must include both the initial velocity and the initial displacements at all points within the region.

Solving the Vibrating String Problem

We can solve Eq. (8.38) numerically by replacing the derivatives with finite-difference approximations, preferring to use central differences in both cases. If we do this, we get

$$\frac{Tg}{w} * \frac{u_{i+1}^j - 2u_i^j + u_{i-1}^j}{(\Delta x)^2} = \frac{u_i^{j+1} - 2u_i^j + u_i^{j-1}}{(\Delta t)^2}$$

where the subscripts indicate x-values and the superscripts indicate t-values.* (If the boundary conditions involve derivatives, we will approximate them with central differences in the way that we are accustomed.) If we solve for the displacement at the end of the current time step, u_i^{j+1}, we get

$$u_i^{j+1} = \frac{Tg(\Delta t)^2}{w(\Delta x)^2}(u_{i+1}^j + u_{i-1}^j) - u_i^{j-1} + 2\left(1 - \frac{Tg(\Delta t)^2}{w(\Delta x)^2}\right)u_i^j.$$

If we make $Tg(\Delta t)^2/w(\Delta x)^2$ equal to 1, the maximum value that avoids instability, there is considerable simplification:

$$u_i^{j+1} = u_{i+1}^j + u_{i-1}^j - u_i^{j-1}, \qquad \Delta t = \frac{\Delta x}{\sqrt{(Tg/w)}}. \tag{8.40}$$

* We again assume evenly spaced nodes and evenly spaced time intervals.

Equation (8.40) shows how one can march through time: To get the new value for u at node i, we add the two u-values last computed at nodes to the right and left and subtract the value at node i at the time step before that. That is fine for the second time step; we have the initial u-values (at $t = 0$) and those for step 1 (at $t = \Delta t$). We also have the necessary information for all subsequent computations. But how do we get the value for the first time step? We seem to need the values of u one time step before the start!

That really is no problem if we recognize that the oscillation of the vibrating string is a periodic function and that the "starting point" is just an arbitrary instant of time at which we happen to know the displacement and the velocity. That suggests that we can get the u-values at $t = -\Delta t$ from the specified initial velocities. If we use a central-difference approximation:

$$\frac{u_i^1 - u_i^{-1}}{2\Delta t} = \frac{\partial u}{\partial t} \qquad \text{at } x_i \text{ and } t = 0,$$

$\partial u/\partial t$ at $t = 0$ is known; it is one of the initial conditions, call it $g(x)$. So we can write

$$u_i^{-1} = u_i^1 - 2g(x)\,\Delta t. \tag{8.41}$$

If we substitute Eq. (8.41) into Eq. (8.40), we have (but for $t = 0$ only),

$$u_i^1 = \frac{1}{2}\,(u_{i+1}^0 + u_{i-1}^0) + g(x)\,\Delta t. \tag{8.42}$$

Our procedure then is to use Eq. (8.42) for the first time step, then use Eq. (8.40) to march on through time after that first step.* As we will see, Eq. (8.40) is not only stable but also can give exact answers. It is interesting that using a value for $Tg(\Delta t)^2/w(\Delta x)^2$ less than 1, while stable, gives results that are less accurate.

An example will illustrate the technique.

EXAMPLE 8.12 A banjo string is 80 cm long and weighs 1.0 gm. It is stretched with a tension of 40,000 g. At a point 20 cm from one end it is pulled 0.6 cm from the equilibrium position and then released. Find the displacements along the string as a function of time. Use $\Delta x = 10$ cm. How long does it take to complete one cycle of motion? From this, compute the frequency of the vibrations.

If Eq. (8.42) is used to begin the calculations and Eq. (8.40) thereafter, the results are as shown in Table 8.11. The initial velocities are zero because the string is just released after being displaced. Observe that the displacements are reproduced every 16 time steps.

Figure 8.13 illustrates how the displacements change with time; it also shows that, after 16 Δt's, the original u-values are reproduced, which will be true for every 16 time steps. Because the original displacements are reproduced every 16 time steps, we can compute the frequency of the vibrations. Each time step is

$$\Delta t = \sqrt{\frac{w}{Tg}} * \Delta x = \sqrt{\frac{1.0/80}{40000 * 980}} * 10 = 0.000179 \text{ sec},$$

* There is a more accurate way to start the computations that we discuss a little later.

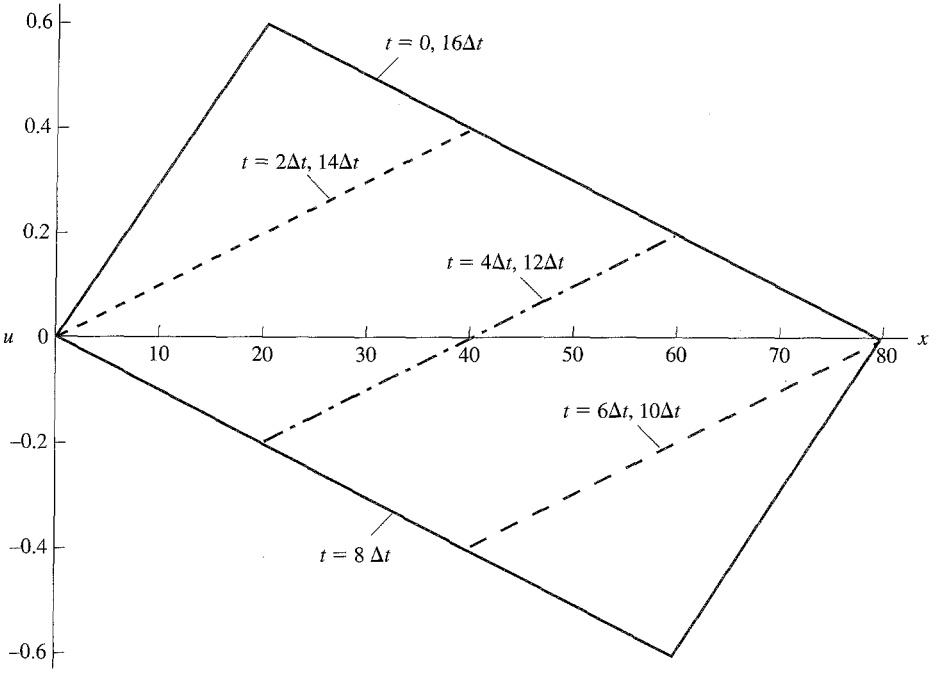


Figure 8.13

and the frequency is

$$f = \frac{1}{16 * 0.000179} = 350 \text{ hertz.}$$

The standard formula from physics is

$$f = \left(\frac{1}{2L}\right)\sqrt{\frac{Tg}{w}} = \left(\frac{1}{160}\right)\sqrt{\frac{40000 * 980}{1.0/80}} = 350 \text{ hertz,}$$

precisely the same!

It seems remarkable that we get exactly the correct frequency, but what about the accuracy of the displacements? We will find that these too are precisely correct, as the next discussion shows. It is also apparent that the computations are stable when $Tg(\Delta t)^2/w(\Delta x)^2$ equals 1.

The D'Alembert Solution

The simple vibrating string problem is one where the analytical solution is readily obtained. This analytical solution is called the *D'Alembert solution*. Consider this expression for $u(x, t)$:

$$u(x, t) = F(x + ct) + G(x - ct), \tag{8.43}$$

where F and G are arbitrary functions.

Table 8.11 Results for vibrating string example

Time steps	u-values at $x =$								
	0	10	20	30	40	50	60	70	80
0	0.00	0.30	0.60	0.50	0.40	0.30	0.20	0.10	0.00
1	0.00	0.30	0.40	0.50	0.40	0.30	0.20	0.10	0.00
2	0.00	0.10	0.20	0.30	0.40	0.30	0.20	0.10	0.00
3	0.00	−0.10	0.00	0.10	0.20	0.30	0.20	0.10	0.00
4	0.00	−0.10	−0.20	−0.10	0.00	0.10	0.20	0.10	0.00
5	0.00	−0.10	−0.20	−0.30	−0.20	−0.10	0.00	0.10	0.00
6	0.00	−0.10	−0.20	−0.30	−0.40	−0.30	−0.20	−0.10	0.00
7	0.00	−0.10	−0.20	−0.30	−0.40	−0.50	−0.40	−0.30	0.00
8	0.00	−0.10	−0.20	−0.30	−0.40	−0.50	−0.60	−0.30	0.00
9	0.00	−0.10	−0.20	−0.30	−0.40	−0.50	−0.40	−0.30	0.00
10	0.00	−0.10	−0.20	−0.30	−0.40	−0.30	−0.20	−0.10	0.00
11	0.00	−0.10	−0.20	−0.30	−0.20	−0.10	0.00	0.10	0.00
12	0.00	−0.10	−0.20	−0.10	0.00	0.10	0.20	0.10	0.00
13	0.00	−0.10	0.00	0.10	0.20	0.30	0.20	0.10	0.00
14	0.00	0.10	0.20	0.30	0.40	0.30	0.20	0.10	0.00
15	0.00	0.30	0.40	0.50	0.40	0.30	0.20	0.10	0.00
16	0.00	0.30	0.60	0.50	0.40	0.30	0.20	0.10	0.00
17	0.00	0.30	0.40	0.50	0.40	0.30	0.20	0.10	0.00
18	0.00	0.10	0.20	0.30	0.40	0.30	0.20	0.10	0.00
19	0.00	−0.10	0.00	0.10	0.20	0.30	0.20	0.10	0.00
20	0.00	−0.10	−0.20	−0.10	0.00	0.10	0.20	0.10	0.00

If we substitute this into the vibrating string equation, which we repeat,

$$\frac{\partial^2 u}{\partial t^2} = \frac{Tg}{w}\frac{\partial^2 u}{\partial x^2}, \tag{8.44}$$

we find that the partial-differential equation is satisfied, because

$$\frac{\partial u}{\partial t} = F'\frac{\partial(x+ct)}{\partial t} + G'\frac{\partial(x-ct)}{\partial t} = cF' - cG',$$
$$\frac{\partial^2 u}{\partial t^2} = c^2F'' + c^2G''; \tag{8.45}$$

$$\frac{\partial u}{\partial x} = F'\frac{\partial(x+ct)}{\partial x} + G'\frac{\partial(x-ct)}{\partial x} = F' + G',$$
$$\frac{\partial^2 u}{\partial x^2} = F'' + G''. \tag{8.46}$$

In Eqs. (8.45) and (8.46), the primes indicate derivatives of the arbitrary functions. Now, substituting these expressions for the second partials into Eq. (8.44), we see that the equation for the vibrating string is satisfied when $c^2 = (Tg/w)$. This means that we can get the solution to Eq. (8.44) if we can find functions F and G that satisfy the initial conditions

and the boundary conditions. That too is not difficult. Suppose we are given the initial conditions

$$u(x, 0) = f(x), \qquad \frac{\partial u}{\partial t}(x, 0) = g(x).$$

The combination

$$u(x, t) = \left(\frac{1}{2}\right)[f(x + ct) + f(x - ct)] + \left(\frac{1}{2c}\right)\int_{x-ct}^{x+ct} g(v)\, dv \tag{8.47}$$

is of the same form as Eq. (8.43). It certainly fulfills the boundary conditions, for substituting $t = 0$ in Eq. (8.47) gives $u(x, 0) = f(x)$ and differentiating with respect to t gives

$$\frac{1}{2}[f' * c + f' * (-c)] = 0$$

for the first term of Eq. (8.47), and

$$\left(\frac{1}{2c}\right)\{(c)[g(x + ct)] - (-c)[g(x - ct)]\} = g(x)$$

(when $t = 0$) for the second term.

We have thus shown that the solution to the vibrating string problem is exactly that given by Eq. (8.47). Now we ask "Does the simple algorithm of Eq. (8.40) match Eq. (8.47) for the example problem?" We can show that the answer to the question is yes in the following way.

First, for $Tg(\Delta t)^2/w(\Delta x)^2$ equal to 1, $\Delta x = c\Delta t$. Recalling that u_i^j represents the u-value at $x = x_i = i\Delta x$ and at $t = t_j = j\Delta t$, we see that $ct_j = cj\Delta t = j\Delta x$. If we write $u(x_i, t_j)$ using our subscript/superscript notation, it becomes

$$\begin{aligned} u_i^j &= F(x_i + ct_j) + G(x_i - ct_j) = F(i\Delta x + j\Delta x) + G(i\Delta x - j\Delta x) \\ &= F[(i + j)\,\Delta x] + G[(i - j)\,\Delta x]. \end{aligned} \tag{8.48}$$

Now let us use Eq. (8.48) to write each term on the right-hand side of Eq. (8.40), the algorithm that we used in the example.

$$\begin{aligned} u_{i+1}^j &= F[(i + 1 + j)\,\Delta x] + G[(i + 1 - j)\,\Delta x], \\ u_{i-1}^j &= F[(i - 1 + j)\,\Delta x] + G[(i - 1 - j)\,\Delta x], \\ u_i^{j-1} &= F[(i + j - 1)\,\Delta x] + G[(i - j + 1)\,\Delta x]. \end{aligned}$$

In the example, both F and G are linear functions of x, so that $F(a) + F(b) = F(a + b)$, and the same is true for G. If we combine these terms in Eq. (8.40),

$$\begin{aligned} u_{i+1}^j + u_{i-1}^j - u_i^{j-1} &= F[(i + 1 + j)]\,\Delta x + (i - 1 + j)\,\Delta x - (i + j - 1)\,\Delta x] \\ &\quad + G[(i + 1 - j)\,\Delta x + (i - 1 - j)\,\Delta x - (i - j + 1)\,\Delta x] \\ &= F\{[i + (j + 1)]\,\Delta x\} + G\{[i - (j + 1)]\,\Delta x\} \\ &= u_i^{j+1}, \end{aligned}$$

and the validity of Eq. (8.40) is proved. The important implication from this is that, if we have correct values for the u's at two successive time steps, all subsequent computed values will be correct.

When the Initial Velocity Is Not Zero

Example 8.12 had the string starting with zero velocity. What if the initial velocity is not zero? Equation (8.42) was a very simple way to begin the computations, but it gave correct results only because $g(x)$ was zero in Eq. (8.47). This next example shows that Eq. (8.42) is inadequate when $g(x) \neq 0$ and that there is a better way to begin.

EXAMPLE 8.13 A string is 9 units long. Initially, it is in its equilibrium position (just a straight line between the supports). It is set into motion by striking it so that it has an initial velocity given by $\partial u/\partial t = 3 \sin(\pi x/L)$. Take $\Delta x = 1$ unit and let $c^2 = Tg/w = 4$. When the ratio $c^2(\Delta t)^2/(\Delta x)^2 = 1$, the value of Δt is 0.5 time units. Find the displacements at the end of one Δt.

Because $\Delta x = 1$ and the length is 9, the string is divided into nine intervals; there are eight interior nodes. We are to compute the u-values at $t = \Delta t = 0.5$.

As we have seen, Eq. (8.42) is one way to get these starting values. However, looking at Eq. (8.47), we see that there is an alternative technique. If we substitute $t = \Delta t$ in that equation and remember that $c\Delta t = \Delta x$, we get for $u(x_i, \Delta t)$

$$\begin{aligned} u(x_i, \Delta t) &= \frac{1}{2}\left[f(x_i + \Delta x) + f(x_i - \Delta x)\right] + \left(\frac{1}{2c}\right)\int_{x-\Delta x}^{x+\Delta x} g(v)\, dv \\ &= \frac{1}{2}\left[u_{i+1}^0 + u_{i-1}^0\right] + \left(\frac{1}{2c}\right)\int_{x-\Delta x}^{x+\Delta x} g(v)\, dv. \end{aligned} \tag{8.49}$$

Equation (8.49) differs from Eq. (8.42) only in the last term. If $g(x) =$ a constant, the last terms are equal, but if $g(x)$ is not constant, we should do the integration in Eq. (8.49). Table 8.12 compares the results of both techniques and also gives the answers from the analytical solution. Only values for x between 1 and 4 are given as the displacements for the right half of the string are the same as for the left half. Simpson's rule was used to do

Table 8.12 Comparison of ways to begin the wave equation at $t = \Delta t$ with $\Delta x = 1$

	u = values from		
x	Eq. (8.42)	Eq. (8.49)	Analytical
1	0.5130	0.5027	0.50267
2	0.9642	0.9448	0.94472
3	1.2990	1.2729	1.27282
4	1.4772	1.4475	1.44740

the integrations. We see from the tabulated results that the values using Eq. (8.49) are almost exactly the same as the analytical values (they are the same within one in the fourth decimal place) but that the results from Eq. (8.42) are less accurate (they each differ by 2.0% from the analytical). We could improve the accuracy with Eq. (8.42) by decreasing the size of Δx (and reducing Δt correspondingly). By making $\Delta x = 0.5$, the errors are reduced fourfold as expected.

Stability of the Solution

We have said that the numerical solution of the vibrating string problem is stable if this ratio is not greater than 1:

$$\frac{Tg\,(\Delta t)^2}{w\,(\Delta x)^2} \leq 1.$$

Because we ordinarily set that ratio equal to 1, it is sufficient to demonstrate stability for that scheme.

For this demonstration, assume that all computations are correct up to a certain point in time, but then an error of size 1 occurs. If the method is stable, that error will not increase. Table 8.13 traces how this single error is propagated. It is allowable to think only of the effect of this single error because for a linear problem that this is, the *principal of superposition* says that we can add together the effects of each of the errors. Equation (8.40) was used and the ends of the string are specified so they are free of error.

Table 8.13 Propagation of single error in numerical solution to wave equation

Initially error-free values	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Error made here →	0.0	0.0	1.0	0.0	0.0	0.0	0.0
	0.0	1.0	0.0	1.0	0.0	0.0	0.0
	0.0	0.0	1.0	0.0	1.0	0.0	0.0
	0.0	0.0	0.0	1.0	0.0	1.0	0.0
	0.0	0.0	0.0	0.0	1.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	−1.0	0.0	0.0
	0.0	0.0	0.0	−1.0	0.0	−1.0	0.0
	0.0	0.0	−1.0	0.0	−1.0	0.0	0.0
	0.0	−1.0	0.0	−1.0	0.0	0.0	0.0
	0.0	0.0	−1.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	1.0	0.0	0.0	0.0	0.0
	0.0	1.0	0.0	1.0	0.0	0.0	0.0
	0.0	0.0	1.0	0.0	1.0	0.0	0.0

The Wave Equation in Two Dimensions

The finite-difference method can be applied to hyperbolic partial-differential equations in two or more space dimensions. A typical problem is the vibrating membrane. Consider a thin, flexible membrane stretched over a rectangular frame and set to vibrating. As we have seen, the equation is

$$\frac{\partial^2 u}{\partial t^2} = \frac{Tg}{w}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right),$$

in which u is the displacement, t is the time, x and y are the space coordinates, T is the uniform tension per unit length, g is the acceleration of gravity, and w is the weight per unit area. For simplification, let $Tg/w = c^2$. Replacing each derivative by its central-difference approximation, and using $h = \Delta x = \Delta y$, gives (we recognize the Laplacian on the right-hand side)

$$\frac{u_{i,j}^{k+1} - 2u_{i,j}^{k} + u_{i,j}^{k-1}}{(\Delta t)^2} = c^2 \frac{u_{i+1,j}^{k} + u_{i-1,j}^{k} + u_{i,j+1}^{k} + u_{i,j-1}^{k} - 4u_{i,j}^{k}}{h^2}. \tag{8.50}$$

Solving for the displacement at time t_{k+1}, we obtain

$$u_{i,j}^{k+1} = \frac{c^2(\Delta t)^2}{h^2}\begin{Bmatrix} & 1 & \\ 1 & 0 & 1 \\ & 1 & \end{Bmatrix} u_{i,j}^{k} - u_{i,j}^{k-1} + \left(2 - 4\frac{c^2(\Delta t)^2}{h^2}\right) u_{i,j}^{k}. \tag{8.51}$$

In Eqs. (8.50) and (8.51), we use superscripts to denote the time. If we let $c^2(\Delta t)^2/h^2 = \frac{1}{2}$, the last term vanishes and we get

$$u_{i,j}^{k+1} = \frac{1}{2}\begin{Bmatrix} & 1 & \\ 1 & 0 & 1 \\ & 1 & \end{Bmatrix} u_{i,j}^{k} - u_{i,j}^{k-1}. \tag{8.52}$$

For the first time step, we get displacements from Eq. (8.53), which is obtained by approximating $\partial u/\partial t$ at $t = 0$ by a central-difference approximation involving $u_{i,j}^{1}$ and $u_{i,j}^{-1}$.

$$u_{i,j}^{1} = \frac{1}{4}\begin{Bmatrix} & 1 & \\ 1 & 0 & 1 \\ & 1 & \end{Bmatrix} u_{i,j}^{0} + (\Delta t) g(x_i, y_j). \tag{8.53}$$

In Eq. (8.53), $g(x, y)$ is the initial velocity.

It should not surprise us to learn that this ratio $c^2(\Delta t)^2/h^2 = \frac{1}{2}$ is the maximum value for stability, in view of our previous experience with explicit methods. However, in contrast

with the wave equation in one space dimension, we do not get exact answers from the numerical procedure of Eq. (8.52), and we further observe that we must use smaller time steps in relation to the size of the space interval. Therefore, we advance in time more slowly. However, the numerical method is straightforward, as the following example will show.

EXAMPLE 8.14 A membrane for which $c^2 = Tg/w = 3$ is streched over a square frame that occupies the region $0 \le x \le 2$, $0 \le y \le 2$, in the xy-plane. It is given an initial displacement described by

$$u = x(2 - x)y(2 - y),$$

and has an initial velocity of zero. Find how the displacement varies with time.

We divide the region with $h = \Delta x = \Delta y = \frac{1}{2}$, obtaining nine interior nodes. Initial displacements are calculated from the initial conditions: $u^0(x, y) = x(2 - x)y(2 - y)$; Δt is taken at its maximum value for stability, $h/(\sqrt{2}\,c) = 0.2041$. The values at the end of one time step are given by

$$u^1_{i,j} = \frac{1}{4}\begin{Bmatrix} & 1 & \\ 1 & 0 & 1 \\ & 1 & \end{Bmatrix} u^0_{i,j},$$

Table 8.14 Displacements of a vibrating membrane—finite-difference method: $\Delta t = h/(\sqrt{2}c)$

	Grid location								
t	**(0.5, 0.5)**	**(1.0, 0.5)**	**(1.5, 0.5)**	**(0.5, 1.0)**	**(1.0, 1.0)**	**(1.5, 1.0)**	**(0.5, 1.5)**	**(1.0, 1.5)**	**(1.5, 1.5)**
0	0.5625 (0.5625)	0.750 (0.750)	0.5625	0.750	1.000 (1.000)	0.750	0.5625	0.750	0.5625
0.204	0.375 (0.380)	0.531 (0.536)	0.375	0.531	0.750 (0.755)	0.531	0.375	0.531	0.375
0.408	−0.031 (−0.044)	0.000 (−0.009)	−0.031	0.000	0.062 (0.083)	0.000	−0.031	0.000	−0.031
0.612	−0.375 (−0.352)	−0.531 (−0.539)	−0.375	−0.531	−0.750 (−0.813)	−0.531	−0.375	−0.531	−0.375
0.816	−0.500 (−0.502)	−0.750 (−0.746)	−0.500	−0.750	−1.125 (−1.114)	−0.750	−0.500	−0.750	−0.500
1.021	−0.375 (−0.407)	−0.531 (−0.535)	−0.375	−0.531	−0.750 (−0.691)	−0.531	−0.375	−0.531	−0.375
1.225	−0.031 (−0.015)	0.000 (0.008)	−0.031	0.000	0.062 (0.030)	0.000	−0.031	0.000	−0.031
1.429	0.375 (0.410)	0.531 (0.534)	0.375	0.531	0.750 (0.688)	0.531	0.375	0.531	0.375

Note: Analytical values are in parentheses.

because $g(x, y)$ in Eq. (8.53) is everywhere zero. For succeeding time steps, Eq. (8.52) is used. Table 8.14 gives the results of our calculations. Also shown in Table 8.14 (in parentheses) are analytical values, computed from the double infinite series:

$$u(x, y, t) = \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} B_{mn} \sin \frac{m\pi x}{a} \sin \frac{n\pi y}{b} \cos\left(c\pi t \sqrt{\frac{m^2}{a^2} - \frac{n^2}{b^2}}\right),$$

$$B_{mn} = \frac{16a^2b^2A}{\pi^6 m^3 n^3} (1 - \cos m\pi)(1 - \cos n\pi),$$

which gives the displacement of a membrane fastened to a rectangular framework, $0 \leq x \leq a$, $0 \leq y \leq b$, with initial displacements of $Ax(a - x)y(b - y)$.

We observe that the finite-difference results do not agree exactly with the analytical calculations. The finite-difference values are symmetrical with respect to position and repeat themselves with a regular frequency. The very regularity of the values itself indicates that the finite-difference computations are in error, because they predict that the membrane could emit a musical note. We know from experience that a drum does not give a musical tone when struck; therefore, the vibrations do not have a cyclic pattern of constant frequency, as exhibited by our numerical results.

Decreasing the ratio of $c^2(\Delta t)^2/h^2$ and using Eq. (8.51) gives little or no improvement in the average accuracy; to approach closely to the analytical results, $h = \Delta x = \Delta y$ must be made smaller. When this is done, Δt will need to decrease in proportion, requiring many time steps and leading to many repetitions of the algorithm and extravagant use of computer time. One remedy is the use of implicit methods, which allow the use of larger ratios of $c^2(\Delta t)^2/h^2$. However, with many nodes, this requires large, sparse matrices similar to the Crank–Nicolson method for parabolic equations in two space dimensions. A.D.I. methods have been used for hyperbolic equations—tridiagonal systems result. We do not discuss these methods.

As with other types of partial-differential equations, if the region is not rectangular or if we desire nodes closer together in some parts of the region, it is much preferred to employ the finite-element method, discussed in the next chapter.

Exercises

Section 8.1

1. Show that Eq. (8.2) results if the thickness of the slab varies with position (x, y).
2. Show that Eq. (8.3) applies if both thickness and thermal conductivity vary with position in a slab.
3. The mixed second derivative $\partial^2 u/(\partial x\, \partial y)$ can be considered as

$$\frac{\partial}{\partial x}\left(\frac{\partial u}{\partial y}\right) = \frac{\partial^2 u}{\partial x\, \partial y} = \frac{\partial}{\partial y}\left(\frac{\partial u}{\partial x}\right).$$

If the nodes are spaced apart a distance h in both the x- and y-directions, show that this derivative can be represented by the pictorial operator

$$\frac{1}{4h^2}\begin{Bmatrix} -1 & 1 \\ 1 & -1 \end{Bmatrix} + O(h^2).$$

4. What ordering of nodes in Example 8.1 will reduce the band width of the coefficient matrix to seven? Can this be done in more than one way? Can it be reduced to less than seven?

▶ **5.** If d^2u/dx^2 is represented as this fourth-order central-difference formula

$$\frac{d^2u}{dx^2} = \frac{-u_{i+2} + 16u_{i+1} - 30u_i + 16u_{i-1} - u_{i-2}}{12h^2},$$

find the fourth-order operator for the Laplacian. (This requires the function to have a continuous sixth derivative.)

6. Derive the nine-point approximation for the Laplacian of Eq. (8.6).

7. Solve Example 8.1 using the nine-point approximation to the Laplacian. What is the band width of the coefficient matrix if numbered as in Figure 8.2? What ordering of nodes will give the minimum band width? Is this the same as the preferred ordering of Exercise 4?

8. The coefficient matrix of Example 8.1 is bonded and symmetric. If it is solved taking advantage of this structure rather than as it is shown, how many fewer arithmetic operations will be needed to get the solution?

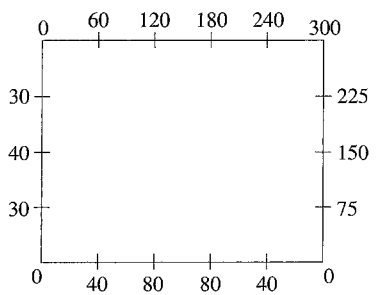
9. A rectangular plate of constant thickness has heat flow only in the x- and y-directions (k is constant). If the top and bottom edges are perfectly insulated and the left edge is at 100° and the right edge at 200°, it is obvious that there is no heat flow except in the x-direction and that temperatures vary linearly with x and are constant along vertical lines.

a. Show that such a temperature distribution satisfies both Eqs. (8.5) and (8.6).

b. Show that the temperatures also satisfy the relation derived in Exercise 5. How should nodes adjacent to the edges be handled?

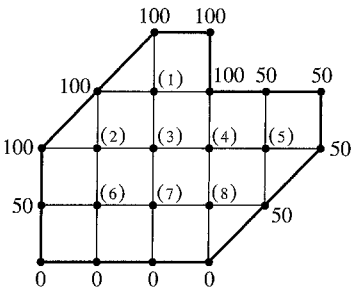
10. What is the operator equivalent to Eq. (8.7) for the nine-point formula?

▶**11.** Solve for the steady-state temperatures in the plate of the figure when the edge temperatures are as shown. The plate is 10 cm × 8 cm, and the nodal spacing is 2 cm.



12. Repeat Exercise 11, but with the nine-point formula. Get the solution both by Gaussian elimination and by iteration. How many iterations does it take to reach the solution with a maximum error of 0.001 at any node?

13. The region on which we solve Laplace's equation does not have to be rectangular. We can apply the methods of Section 8.1 to any region where the nodes fall on the boundary. Solve for the steady-state temperatures at the eight interior points of this figure.



▶**14.** Solve Exercise 11 by Liebmann's method with all elements of the initial u-vector equal to zero. Then repeat with all elements equal to 300°, the upper bound to the steady-state temperatures. Repeat again with the initial values all equal to the arithmetic average of the boundary temperatures. Compare the number of iterations needed to reach a given tolerance for convergence in each case. What is the effect of the tolerance value that is used?

15. Repeat Exercise 14, but now use overrelaxation with the factor given in Eq. (8.8).

16. Find the torsion function ϕ for a 2 in. × 2 in. square bar.

a. Subdivide the region into nine equal squares, so that there are four interior nodes. Because of symmetry, all of the nodes will have equal ϕ-values.

b. Repeat, but subdivide into 36 equal squares with 25 interior nodes. Use the results of part (a) to get starting values for iteration.

17. Solve

$$\nabla^2 u = 2 + x^2 + y^2$$

over a hollow square bar, 5 in. in outside dimension and with walls 2 in. thick (so that the inner square hole is 1 in. on a side). The origin for x and y is the center of the object. On the inner and outer surfaces, $u = 0$.

18. Solve

$$\nabla^2 u = 2 + x^2 + y^2$$

over a hollow square bar whose outside width is 5 in. There is an inner concentric square hole of width 2 in.

(so that the thickness of the wall is 1.5 in.). The origin for x and y is the center of the object. On the outer and inner surfaces, $u = 0$. Space nodes 0.5 in. apart.

19. Can Exercise 18 be solved by iterations as well as by elimination? Repeat it using a method other than the one you used in solving Exercise 18. Which method would be preferred if nodes are spaced very closely together, say, at 0.01 in.?

20. Repeat Exercise 17 but use overrelaxation. Find the optimal overrelaxation factor experimentally. Does this match to that from Eq. (8.8)?

▶**21.** Solve for the steady-state temperatures in the region of Exercise 13, except now the plate is insulated along the edge where the temperatures were zero. All temperatures on the other edges are as shown in the figure.

22. Solve a modification of Example 8.1, where along every edge there is an outward gradient of $-15°C/cm$. Is it possible to get a unique solution?

23. Solve Exercise 11 by the A.D.I. method using $\rho = 1.0$. Begin with the initial values equal to the arithmetic average of the boundary temperatures. Compare the number of iterations needed to those required with Liebmann's method (Exercise 14) and with those using S.O.R. with the optimal overrelaxation factor (Exercise 15).

24. Repeat Exercise 16 but now use the A.D.I. method. Vary the value of ρ to find the optimal value experimentally.

▶**25.** A cube is 7 cm along each edge. Two opposite faces are held at 100°, the other four faces are held at 0°. Find the interior temperatures at the nodes of a 1 cm network. Use the A.D.I. method

26. Repeat Exercise 25, but now the two opposite edges have a mixed condition: The outward normal gradient equals $0.25(u - 18)$, where u is the surface temperature.

Section 8.2

▶**27.** Suppose that the rod sketched in Figure 8.6 is tapered, with the diameter varying linearly from 2 in. at the left end to 1.25 in. at the right end; the rod is 14 in. long and is made of steel. If 200 BTU/hr of heat flows from left to right (the flow is the same at each x-value along the rod—steady state), what are the values of the gradient at

a. The left end?
b. The right end?
c. $x = 3$ in.?

28. Solve for the temperatures at $t = 2.06$ sec in the 2-cm thick steel slab of Example 8.8 if the initial temperatures are given by

$$u(x, t) = 100 \sin\left(\frac{\pi x}{2}\right).$$

Use the explicit method with $\Delta x = 0.25$ cm. Compare to the analytical solution: $100e^{-0.3738t} \sin(\pi x/2)$.

29. Repeat Exercise 28, but now with Crank–Nicolson.

30. Repeat Exercise 28, but now with the theta method:

a. $\theta = 2/3$.
b. $\theta = 0.878$.
c. $\theta = 1.0$.

▶**31.** Solve for the temperatures in a cylindrical copper rod that is 8 in. long and whose curved outer surface is insulated so that heat flows only in one direction. The initial temperature is linear from 0°C at one end to 100°C at the other, when suddenly the hot end is brought to 0°C and the cold end is brought to 100°C. Use $\Delta x = 1$ in. and an appropriate value of Δt so that $k\,\Delta t/c\rho(\Delta x)^2 = \frac{1}{2}$. Look up values for k, c, and ρ in a handbook. Carry out the solution for 10 time steps.

32. Repeat Exercise 31, but with $\Delta x = 0.5$ in., and compare the temperature at points 1 in., 3 in., and 6 in. from the cold end with those of the previous exercise. You will need to compute more time steps to match the 10 steps done previously.

You will find it instructive to graph the temperatures for both sets of computations.

33. Repeat Exercise 31 but with $\Delta x = 1.0$ and Δt such that the ratio $k\Delta t/c\rho(\Delta x^2) = 1/4$. Compare the results with both Exercises 31 and 32.

▶**34.** A rectangular plate 3 in. × 4 in. is initially at 50°. At $t = 0$, one 3-in. edge is suddenly raised to 100°, and one 4-in. edge is suddenly cooled to 0°. The temperature on these two edges is held constant at these temperatures. The other two edges are perfectly insulated. Use a 1 in. grid to subdivide the plate and write the A.D.I. equations for each of the six nodes where unknown temperatures are involved. Use $r = 2$, and solve the equations for four time steps.

35. A cube of aluminum is 4 in. on each side. Heat flows in all three directions. Three adjacent faces lose heat by conduction to a flowing fluid; the other faces are held at a constant temperature different from that of the fluid. Set up the equations that can be solved for the temperature at nodes using the explicit method with a 1-in. spacing between all nodes. How many time steps

are needed to reach 15.12 sec using the maximum r-value for stability? (Look up the properties of aluminum in a handbook). How many equations must be solved at each time step?

36. Repeat Exercise 35 for Crank–Nicolson with $r = 1$.

37. Repeat Exercise 35 for the implicit method with $r = 1$.

38. Repeat Exercise 35 for the A.D.I. method with $r = 1$.

▶**39.** Demonstrate that the explicit method is unstable with $r = 0.6$ by performing computation similar to that of Table 8.8

40. Demonstrate that the explicit method is stable if $r = 0.25$ by performing computations similar to that of Table 8.8. Do the errors damp out as rapidly?

▶**41.** Suppose that the end conditions are not $u =$ a constant as in Table 8.8 but rather $u_x = 0$. Demonstrate by performing calculations similar to those in Table 8.8 that the explicit method is still stable for $r = 0.5$ but that the errors damp out much more slowly. Observe that the errors at a later stage become a linear combination of earlier errors.

42. Demonstrate by performing calculations similar to those in Table 8.8 that the Crank–Nicolson method is stable even if $r = 10$. You will need to solve a system of equations in this exercise.

43. Compute the largest eigenvalue of the coefficient matrix in Eq. (8.33) for $r = 0.5$, then for $r = 0.6$. Do you find that the statements in the text relative to eigenvalues are confirmed?

▶**44.** Starting with the matrix form of the implicit method. show that for $A^{-1}B$ none of the eigenvalues exceed 1 in magnitude.

Section 8.3

45. Classify the following as elliptic, parabolic, or hyperbolic.

a. $(Tw_x)_x = p * g$.

b. $(xu_x)_x + u_y = \dfrac{(2 + x + y)}{(1 - x)}$.

c. $kU_{tt} + mU_{xt} - (au_x)_x + bU = f(x, t)$.

d. $(TW_x)_x - k^2W_t = 0,\ W(0) = 0,\ W(L) = 0$.

▶**46.** For what values of x and y is this equation elliptic, parabolic, hyperbolic?

$$(1 + y)u_{xx} + 2(1 - x)u_{xy} - (1 - y)u_{yy} = f(x, y).$$

47. Divide the (x, y)-plane into regions where this equation is elliptic, parabolic, hyperbolic:

$$x^3u_{xx} - 2x^2yu_{xy} + xu_{yy} = x^2 - u_x + u_y.$$

48. What would be the equivalent of Eq. (8.38) if the weight per unit length of the string is not constant but varies, $w = W(x)$?

49. If the banjo string of Example 8.12 is tightened or shortened (as by holding it down on a fret with a finger), the pitch of the sound is higher. What would be the frequency of the sound if the tension is made 42,500 gm and the effective length is 65 cm? Compare your answer to the analytical value that is given by

$$f = (1/2L)\sqrt{(Tg/w)}.$$

50. A vibrating string has $Tg/w = 4$ cm^2/sec^2 and is 48 cm long. Divide the length into subintervals so that $\Delta x = L/8$. Find the displacement for $t = 0$ to $t = L$ if both ends are fixed and the initial conditions are

▶ a. $y = x(x - L)/L^2,\ y_t = 0$. ($y_t$ is the velocity.)

b. the string is displaced $+2$ units at $L/4$ and -1 unit at $5L/8,\ y_t = 0$.

▶ c. $y = 0,\ y_t = x(L - x)/L^2$. (Use Eq. (8.42).)

▶ d. the string is displaced 1 unit at $L/2,\ y_t = -y$.

e. Compare part (a) to the analytical solution,

$$y = \frac{8}{\pi^3}\sum_{n=1}^{\infty}\frac{1}{(2n-1)^3}\sin\left[(2n-1)\frac{\pi x}{L}\right]\cos\left[(4n-2)\frac{\pi t}{L}\right].$$

51. The function u satisfies the equation

$$u_{xx} = u_{tt},$$

with boundary conditions of $u = 0$ at $x = 0$ and $u = 0$ at $x = 1$, and with initial conditions

$$u = \sin(\pi x), \qquad u_t = 0, \qquad \text{for } 0 \le x \le 1.$$

Solve by the finite-difference method and show that the results are the same as the analytical solution,

$$u(x, t) = \sin(\pi x)\cos(\pi t).$$

52. The ends of the vibrating string do not have to be fixed. Solve the equation $u_{xx} = u_{tt}$ with $y(x, 0) = 0,\ y_t(x, 0) = 0$ for $0 \le x \le 1$, and end conditions of

$$y(0, t) = 0, \qquad y(1, t) = \sin\left(\frac{\pi t}{4}\right), \qquad y_x(1, t) = 0.$$

53. If the initial velocity of a vibrating string is not zero, Eq. (8.42) is an inaccurate way to start the solution, so parts (c) and (d) of Exercise 50 are not exact. Repeat these computations, but use Eq. (8.49) employing Simpson's $\frac{1}{3}$ rule. How much difference does this make in the answers?

54. Repeat Exercise 53, but now use more points around x_i. Does this change the answers to Exercise 53?

▶**55.** A string that weighs w lb/ft is tightly stretched between $x = 0$ and $x = L$ and is initially at rest. Each point is given an initial velocity of

$$y_t(x, 0) = v_0 \sin^3\left(\frac{\pi x}{L}\right).$$

The analytical solution is

$$y(x, t) = \frac{v_0 L}{12a\pi}\left(9 \sin\frac{\pi x}{L}\sin\frac{a\pi t}{L} - \sin\frac{3\pi x}{L}\sin\frac{3a\pi t}{L}\right),$$

where $a = \sqrt{Tg/w}$, with T the tension and g the acceleration due to gravity. When $L = 3$ ft, $w = 0.02$ lb/ft, and $T = 5$ lb, with $v_0 = 1$ ft/sec, the analytical formula predicts $y = 0.081$ in. at the midpoint when $t = 0.01$ sec. Solve the problem numerically to confirm this. Does your solution agree with the analytical solution at other values of x and t?

56. Solve the vibrating membrane problem of Example 8.14 with different initial conditions:

$$u(x, y) = 0, \qquad u_t(x, y) = x^2(2 - x)y^2(2 - y).$$

57. Repeat Exercise 56 with the initial conditions reversed:

$$u(x, y) = x^2(2 - x)\, y^2(2 - y), \qquad u_t(x, y) = 0.$$

▶**58.** A membrane is stretched over a frame that occupies the region in the xy-plane bounded by

$$x = 0, \qquad x = 3, \qquad y = 0, \qquad y = 2.$$

At $t = 0$, the point on the membrane at (1, 1) is lifted 1 unit above the xy-plane and then released. If $T = 6$ lb/in. and $w = 0.55$ lb/in.^{2}, find the displacement of the point (2, 1) as a function of time.

59. How do the vibrations of Exercise 58 change if $w = 0.055$ with other parameters remaining the same?

60. The frame holding the membrane of Exercise 58 is distorted by lifting the corner at (3, 2) 1 unit above the xy-plane. (The members of the frame elongate so that the corner moves vertically.) The membrane is set to vibrating in the same way as in Exercise 58. Follow the vibrations through time. [Assume that the rest positions of points on the membrane lie on the two planes defined by the adjacent edges that meet at (0, 0) and at (3, 2).]

Applied Problems and Projects

APP1. A classic problem in elliptic partial-differential equations is to solve $\nabla^2 u = 0$ on a region defined by $0 \le x \le \pi$, $0 \le y \le \infty$, with boundary condition of $u = 0$ at $x = 0$, at $x = \pi$, and at $y = \infty$. The boundary at $y = 0$ is held at $u = F(x)$. This can be quite readily solved by the method of separation of variables, to give the series solution

$$u = \sum_{n=1}^{\infty} B_n e^{-ny} \sin nx,$$

with

$$B_n = 2\int_0^{\pi} F(x)\sin nx\, dx.$$

Solve this equation numerically for various definitions of $F(x)$. (You will need to redefine the region so that $0 \le y \le M$, where M is large enough that changes in u with y at $y = M$ are negligible.) Compare your results to the series solution. You might try

$$F(x) = 100 \sin(x); \qquad F(x) = 4x(\pi - x)/\pi^2; \qquad F(x) = 100(\pi - |2x - \pi|).$$

APP2. The equation

$$2\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} - \frac{\partial u}{\partial x} = 2$$

is an elliptic equation. Solve it on the unit square, subject to $u = 0$ on the boundaries. Approximate the first derivative by a central-difference approximation. Investigate the effect of size of Δx on the results, to determine at what size reducing it does not have further effect.

APP3. If you write out the equations for Example 8.1, you will find that the coefficient matrix is symmetric and banded. How can you take advantage of this in solving the equations by Gaussian elimination? Would Gauss–Jordan be preferred? Is the matrix still symmetric and banded if the nodes are numbered by columns?

APP4. A symmetric banded coefficient matrix of width b can be stored in an $n \times (b + 1)/2$ array. Develop an algorithm for reducing the coefficient matrix by Gaussian elimination. Test it with program using a system of width 5. How many fewer operations are needed compared to elimination when the matrix is not compressed ($n \times 5$ versus $n \times 3$)?

APP5. If we want to improve the accuracy of the solution to Example 8.6, there are several alternative strategies, including

a. Recompute with nodes more closely spaced but still in a uniform grid.
b. Use a higher-order approximation, such as Eq. (8.6).
c. Add additional nodes only near the right and left sides because the gradient is large there (see Table 8.2) and errors will be greater.

Discuss the pros and cons of each of these choices. Be sure to consider how boundary conditions will be handled. In part (c), how should equations be written where the nodal spacing changes?

APP6. Solve Example 8.1 by S.O.R. with different values for ω. What value is optimal? How do the starting values that are used affect this?

APP7. A vibrating string, with a damping force-opposing its motion that is proportional to the velocity, follows the equation

$$\frac{\partial^2 y}{\partial t^2} = \frac{Tg}{w} * \frac{\partial^2 y}{\partial x^2} - B\frac{\partial y}{\partial t}.$$

where B is the magnitude of the damping force. Solve the problem if the length of the string is 5 ft with $T = 24$ lb, $w = 0.1$ lb/ft, and $B = 2.0$. Initial conditions are

$$y(x)|_{t=0} = \frac{x}{3}, \qquad 0 \le x < 3,$$
$$y(x)|_{t=0} = \frac{5}{2} - \frac{x}{2}, \qquad 3 \le x \le 5,$$
$$\left.\frac{\partial y}{\partial t}\right|_{t=0} = x(x - 5).$$

Compute a few points of the solution by difference equations.

APP8. When steel is forged, billets are heated in a furnace until the metal is of the proper temperature, between 2000°F and 2300°F. It can then be formed by the forging press into rough shapes that are later given their final finishing operations. To produce a certain machine part, a billet of size 4 in. × 4 in. × 20 in. is heated in a furnace whose temperature is maintained at 2350°F. You have been requested to estimate how long it will take all parts of the billet to reach a temperature above 2000°F. Heat transfers to the surface of the billet at a very high rate, principally through radiation. It has been suggested that you can solve the problem by assuming that the surface temperature becomes 2250°F instantaneously and remains at that temperature. Using this assumption, find the required heating time.

Because the steel piece is relatively long compared to its width and thickness, it may not introduce significant error to calculate as if it were infinitely long. This will simplify the problem, permitting a two-dimensional treatment rather than a three-dimensional one. Such a calculation should also give a more conservative estimate of heating time. Compare the estimates from two- and three-dimensional approaches.

APP9. After you have calculated the answers to APP8, your results have been challenged on the basis of assuming constant surface temperature of the steel. Radiation of heat flows according to the equation

$$q = E\sigma(u_F^4 - u_S^4) \text{ Btu/(hr * ft}^2)$$

where E = emissivity (use 0.80), σ is the Stefan–Boltzmann constant (0.171×10^{-8} Btu/(hr * ft^2 * °R^4), u_F and u_S are the furnace and surface absolute temperatures, respectively (°F + 460°).

The heat radiating to the surface must also flow into the interior of the billet by conduction, so

$$q = -k\frac{\partial u}{\partial x},$$

where k is the thermal conductivity of steel (use 26.2 Btu/(hr * ft^3 * (°F/ft)) and $(\partial u/\partial x)$ is the temperature gradient at the surface in a direction normal to the surface. Solve the problem with this boundary condition, and compare your solution to that of APP8. (Observe that this is now a nonlinear problem. Think carefully how your solution can cope with it.)

APP10. A horizontal elastic rod is initially undeformed and is at rest. One end, at $x = 0$, is fixed, and the other end, at $x = L$ (when $t = 0$), is pulled with a steady force of F lb/ft^2. It can be shown that the displacements $y(x, t)$ of points originally at the point x are given by

$$\frac{\partial^2 y}{\partial t^2} = a^2\frac{\partial^2 y}{\partial x^2}, \qquad y(0, t) = 0, \qquad \left.\frac{\partial y}{\partial t}\right|_{x=L} = \frac{F}{E}.$$

$$y(x, 0) = 0, \qquad \left.\frac{\partial y}{\partial t}\right|_{t=0} = 0.$$

where $a^2 = Eg/\rho$; E = Young's modulus (lb/ft^2); g = acceleration of gravity; ρ = density (lb/ft^3). Find y versus t for the midpoint of a 2-ft-long piece of rubber for which $E = 1.8 \times 10^6$ and $\rho = 70$ if $F/E = 0.7$.

APP11. A circular membrane, when set to vibrating, obeys the equation (in polar coordinates)

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2 u}{\partial \theta^2} = \frac{w}{Tg}\frac{\partial^2 u}{\partial t^2}.$$

A 3-ft-diameter kettledrum is started to vibrating by depressing the center $\frac{1}{2}$ in. If $w = 0.072$ lb/ ft^2 and $T = 80$ lb/ft, find how the displacements at 6 in. and 12 in. from the center vary with time. The problem can be solved in polar coordinates, or it can be solved in rectangular coordinates using the method of Eq. (8.19) to approximate $\nabla^2 u$ near the boundaries.

APP12. A flexible chain hangs freely, as shown in Figure 8.14. For small disturbances from its equilibrium position (hanging vertically), the equation of motion is

$$x\frac{\partial^2 y}{\partial x^2} + \frac{\partial y}{\partial x} = \frac{1}{g}\frac{\partial^2 y}{\partial t^2}.$$

In this equation, x is the distance from the end of the chain, y is the displacement from the equilibrium position, t is the time, and g is the acceleration of gravity. A 10-ft-long chain is originally hanging freely. It is set into motion by striking it sharply at its midpoint, imparting a velocity there of 1 ft/sec. Find how the chain moves as a result of the blow. If you find you need additional information at $t = 0$, make reasonable assumptions.

APP13. Shipment of liquefied natural gas by refrigerated tankers to industrial nations may become an important means of supplying the world's energy needs. It must be stored at the receiving port, however.

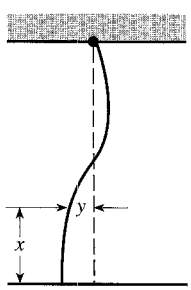


Figure 8.14

[A. R. Duffy and his coworkers (1967) discuss the storage of liquefied natural gas in underground tanks.] A commercial design, based on experimental verification of its feasibility, contemplated a prestressed concrete tank 270 ft in diameter and 61 ft deep, holding some 600,000 bbl of liquefied gas at −258°F. Convection currents in the liquid were shown to keep the temperature uniform at this value, the boiling point of the liquid.

Important considerations of the design are the rate of heat gained from the surroundings (causing evaporation of the liquid gas) and variation of temperatures in the earth below the tank (relating to the safety of the tank, which could be affected by possible settling or frost-heaving.)

The tank itself is to be made of concrete 6 in. thick, covered with 8 in. of insulation (on the liquid side). (A sealing barrier keeps the insulation free of liquid, otherwise, its insulating capacity would be impaired.) The experimental tests showed that there is a very small temperature drop through the concrete: 12°F. This observed 12°F temperature difference seems reasonable in light of the relatively high thermal conductivity of concrete. We expect then that most of the temperature drop occurs in the insulation or in the earth below the tank.

Because the commercial-design tank is very large, if we are interested in ground temperatures near the center of the tank (where penetration of cold will be a maximum), it should be satisfactory to consider heat flowing in only one dimension, in a direction directly downward from the base of the tank. Making this simplifying assumption, compute how long it will take for the temperature to decrease to 32°F (freezing point of water) at a point 8 ft away from the tank wall. The necessary thermal data are

	Insulation	Concrete	Earth
Thermal conductivity (Btu/(hr * ft * °F))	0.013	0.90	2.6
Density (lb/ft^3)	2.0	150	132
Specific heat (Btu/(lb * °F))	0.195	0.200	0.200

Assume the following initial conditions: temperature of liquid, −258°F; temperature of insulation, −258°F to 72°F (inner surface to outer); temperature of concrete, 72°F to 60°F; temperature of earth, 60°F.

APP14. XYZ Metallurgical has a problem. A slab of steel, 6 ft long, 12 in. wide, and 3 in. thick, must be heat treated and it is a rush job. Unfortunately, their large furnace is down for repairs and the only furnace that can be used will hold just three feet of the slab. It has been proposed that it would be possible to use this furnace if the three feet of the slab that protrude from the furnace are well insulated. (See the figure.) The heat treating requires that all of the slab be held between 950°F and 900°F for at least an hour. The portion that is outside the furnace is covered with a 1 in. thickness of insulation whose thermal conductivity, k, is 0.027 Btu/(hr * ft * °F). Even though you are a new employee, the manager has asked you to determine three things:

(1) Is one inch of this insulation sufficient for all of the slab to reach 900°F with the furnace at 950°F?
(2) If it is, how long will it take for the end of the slab to reach that temperature?
(3) If one inch is insufficient, how much of this same insulation should be used?