8

COMPUTATION OF SOLUTIONS

We have found formulas for many solutions to PDEs, but other problems encountered in practice are not as simple and cannot be solved by formula. Even when there is a formula, it might be so complicated that we would prefer to visualize a typical solution by looking at its graph. The opportunity presented in this chapter is to reduce the process of solving a PDE with its auxiliary conditions to a finite number of arithmetical calculations that can be carried out by computer. All the problems we have studied can be so reduced. However, there are dangers in doing so. If the method is not carefully chosen, the numerically computed solution may not be anywhere close to the true solution. The other danger is that the computation (for a difficult problem) could easily take so long that it would take more computer time than is practical to carry out (years, millenia, ...). The purpose of this chapter is to illustrate the most important techniques of computation using quite simple equations as examples.

8.1 OPPORTUNITIES AND DANGERS

The best known method, *finite differences*, consists of replacing each derivative by a difference quotient. Consider, for instance, a function $u(x)$ of one variable. Choose a *mesh size* Δx. Let's approximate the value $u(j\Delta x)$ for $x = j\Delta x$ by a number u_j indexed by an integer j:

$$u_j \sim u(j\Delta x).$$

Then the three standard approximations for the *first derivative* $\dfrac{\partial u}{\partial x}(j\Delta x)$ are:

$$\boxed{\begin{aligned}&\text{The backward difference: } \frac{u_j - u_{j-1}}{\Delta x} &(1)\\ &\text{The forward difference: } \frac{u_{j+1} - u_j}{\Delta x} &(2)\\ &\text{The centered difference: } \frac{u_{j+1} - u_{j-1}}{2\Delta x}. &(3)\end{aligned}}$$

Each of them is a correct approximation because of the Taylor expansion:

$$u(x + \Delta x) = u(x) + u'(x)\Delta x + \tfrac{1}{2}u''(x)(\Delta x)^2 + \tfrac{1}{6}u'''(x)(\Delta x)^3 + O(\Delta x)^4.$$

[It is valid if $u(x)$ is a C^4 function.] Replacing Δx by $-\Delta x$, we get

$$u(x - \Delta x) = u(x) - u'(x)\Delta x + \tfrac{1}{2}u''(x)(\Delta x)^2 - \tfrac{1}{6}u'''(x)(\Delta x)^3 + O(\Delta x)^4.$$

From these two expansions we deduce that

$$\begin{aligned}u'(x) &= \frac{u(x) - u(x - \Delta x)}{\Delta x} + O(\Delta x)\\ &= \frac{u(x + \Delta x) - u(x)}{\Delta x} + O(\Delta x)\\ &= \frac{u(x + \Delta x) - u(x - \Delta x)}{2\Delta x} + O(\Delta x)^2.\end{aligned}$$

We have written $O(\Delta x)$ to mean any expression that is bounded by a constant times Δx, and so on. Replacing x by $j\,\Delta x$, we see that (1) and (2) are correct approximations to the order $O(\Delta x)$ and (3) is correct to the order $O(\Delta x)^2$.

For the *second derivative*, the simplest approximation is the

$$\boxed{\text{centered second difference:} \quad u''(j\,\Delta x) \sim \frac{u_{j+1} - 2u_j + u_{j-1}}{(\Delta x)^2}.} \quad (4)$$

This is justified by the same two Taylor expansions given above which, when added, give

$$u''(x) = \frac{u(x + \Delta x) - 2u(x) + u(x - \Delta x)}{(\Delta x)^2} + O(\Delta x)^2.$$

That is, (4) is valid with an error of $O(\Delta x)^2$.

For functions of two variables $u(x, t)$, we choose a mesh size for both variables. We write

$$\boxed{u(j\,\Delta x, n\,\Delta t) \sim u_j^n,}$$

where the n is a superscript, not a power. Then we can approximate, for instance,

$$\frac{\partial u}{\partial t}(j\Delta x,\ n\,\Delta t) \sim \frac{u_j^{n+1} - u_j^n}{\Delta t}, \tag{5}$$

the forward difference for $\partial u/\partial t$. Similarly, the forward difference for $\partial u/\partial x$ is

$$\frac{\partial u}{\partial x}(j\Delta x,\ n\,\Delta t) \sim \frac{u_{j+1}^n - u_j^n}{\Delta x}, \tag{6}$$

and we can write similar expressions for the differences (1)–(4) in either the t or x variables. □

Two kinds of errors can be introduced in a computation using such approximations. *Truncation error* refers to the error introduced in the solutions by the approximations themselves, that is, the $O(\Delta x)$ terms. Although the error in the equation may be $O(\Delta x)$, the error in the solutions (the truncation error) may or may not be small. This error is a complicated combination of many small errors. We want the truncation error to tend to zero as the mesh size tends to zero. Thinking of Δx as a very small number, it is clear that $O(\Delta x)^2$ is a much smaller error than $O(\Delta x)$. The errors written in (1)–(4) are, strictly speaking, called *local* truncation errors. They occur in the approximation of the individual terms in a differential equation. *Global* truncation error is the error introduced in the actual solutions of the equation by the cumulative effects of the local truncation errors. The passage from local to global errors is usually too complicated to follow in any detail.

Roundoff error occurs in a real computation because only a certain number of digits, typically 8 or 16, are retained by the computer at each step of the computation. For instance, if all numbers are rounded to eight digits, the dropping of the ninth digit could introduce big cumulative errors in a large computation. We have to prevent these little errors from accumulating.

Example 1.

Let's solve the very simple problem

$$u_t = u_{xx}, \qquad u(x,0) = \phi(x)$$

using finite differences. We use a forward difference for u_t and a centered difference for u_{xx}. Then the *difference equation* is

$$\boxed{\frac{u_j^{n+1} - u_j^n}{\Delta t} = \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{(\Delta x)^2}.} \tag{7}$$

It has a local truncation error of $O(\Delta t)$ (from the left side) and $O(\Delta x)^2$ (from the right side).

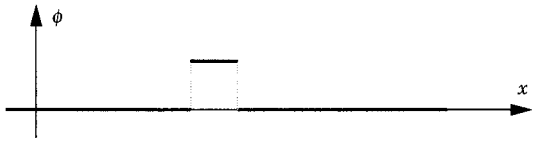


Figure 1

Suppose that we choose a very small value for Δx and choose $\Delta t = (\Delta x)^2$. Then (7) simplifies to

$$u_j^{n+1} = u_{j+1}^n - u_j^n + u_{j-1}^n. \tag{8}$$

Let's take $\phi(x)$ to be the very simple step function (see Figure 1), which is to be approximated by the values ϕ_j:

$$0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \qquad \rightarrow x.$$

A sample calculation with these simple initial data can be done by hand by simply "marching in time." That is, $\phi(x)$ provides u_j^0, then the "scheme" (8) gives u_j^1, then (8) gives u_j^2, and so on. We can summarize (8) schematically using the diagram

$$\begin{array}{ccccc} & & * & & \\ \bullet & +1 & \bullet & -1 & \bullet \quad +1 \end{array}$$

(called a *template*), which means that in order to get u_j^{n+1} you just add or subtract its three lower neighbors as indicated. Thus simple arithmetic gives us the result shown in Figure 2. (Verify it!) The values of u_j^n are written in the (j, n) location. This is supposed to be an approximate solution.

The result is horrendous! It is nowhere near the true solution of the PDE. We know that by the maximum principle, the true solution of the diffusion equation will always be between zero and one, but the difference equation has given us an "approximation" with the value 19 and growing! □

In the next section we analyze what went wrong.

$n=4$	1	−4	10	−16	19	−16	10	−4	1	
$n=3$	0	1	−3	6	−7	6	−3	1	0	
$n=2$	0	0	1	−2	3	−2	1	0	0	
$n=1$	0	0	0	1	−1	1	0	0	0	
$n=0$	0	0	0	0	1	0	0	0	0	$\rightarrow x$

Figure 2

EXERCISES

1. The Taylor expansion written in Section 8.1 is valid if u is a C^4 function. If $u(x)$ is merely a C^3 function, the best we can say is that the Taylor expansion is valid only with a $o(\Delta x)^3$ error. [This notation means that the error is $(\Delta x)^3$ times a factor that tends to zero as $\Delta x \to 0$.] If merely a C^2 function, it is only valid with a $o(\Delta x)^2$ error, and so on.
   (a) If $u(x)$ is merely a C^3 function, what is the error in the first derivative due to its approximation by the centered difference?
   (b) What if $u(x)$ is merely a C^2 function?
2. (a) If $u(x)$ is merely a C^3 function, what is the error in the *second* derivative due to its approximation by a centered second difference?
   (b) What if $u(x)$ is merely a C^2 function?
3. Suppose that we wish to approximate the first derivative $u'(x)$ of a very smooth function with an error of only $O(\Delta x)^4$. Which difference approximation could we use?

8.2 APPROXIMATIONS OF DIFFUSIONS

We take up our discussion of the diffusion equation $u_t = u_{xx}$ again. There is nothing obviously wrong with the scheme we used, as each derivative is appropriately approximated with a small local truncation error. Somehow the little errors have accumulated! What turns out to be wrong, but this is *not* obvious at this point, is the choice of the mesh Δt relative to the mesh Δx. Let's make no assumption now about these meshes; in fact, let

$$\boxed{s = \frac{\Delta t}{(\Delta x)^2}.} \tag{1}$$

As before, we can solve the scheme (8.1.7) for u_j^{n+1}:

$$\boxed{u_j^{n+1} = s\left(u_{j+1}^n + u_{j-1}^n\right) + (1-2s)u_j^n.} \tag{2}$$

The scheme is said to be *explicit* because the values at the $(n+1)$st time step are given explicitly in terms of the values at the earlier times.

Example 1.

To be specific, let's consider the standard problem:

$$\begin{aligned} u_t &= u_{xx} && \text{for } 0 < x < \pi, t > 0 \\ u &= 0 && \text{at } x = 0, \pi \\ u(x,0) = \phi(x) &= \begin{cases} x & \text{in } \left(0, \frac{\pi}{2}\right) \\ \pi - x & \text{in } \left(\frac{\pi}{2}, \pi\right). \end{cases} \end{aligned}$$

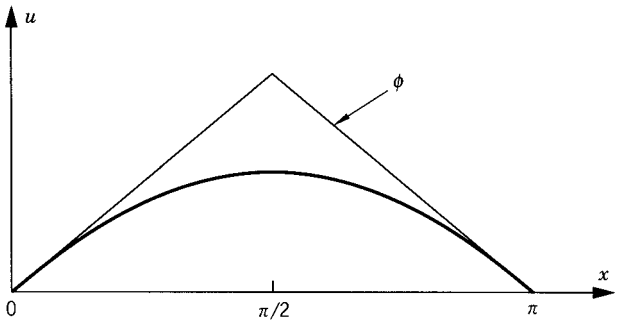


Figure 1

Its exact solution from Section 5.1 is

$$u(x,t) = \sum_{k=1}^{\infty} b_k \sin kx\, e^{-k^2 t}, \tag{3}$$

where $b_k = 4(-1)^{(k+1)/2}/\pi k^2$ for odd k, and $b_k = 0$ for even k. It looks like Figure 1 for some $t > 0$ ($t = 3\pi^2/80$).

We approximate this problem by the scheme (2) for $j = 0, 1, \ldots, J-1$ and $n = 0, 1, 2, \ldots$ together with the discrete boundary and initial conditions

$$u_0^n = u_J^n = 0 \qquad \text{and} \qquad u_j^0 = \phi(j\Delta x).$$

For $J = 20$, $\Delta x = \pi/20$, and $s = \frac{5}{11}$, the result of the calculation (from page 6 of [RM]) is shown in Figure 2 (exactly on target!). However, if we repeat the calculation for $J = 20$, $\Delta x = \pi/20$, and $s = \frac{5}{9}$, the result is as shown in Figure 3 (wild oscillations as in Section 8.1!). Thus the choice $s = \frac{5}{11}$ is stable, whereas $s = \frac{5}{9}$ is clearly unstable. □

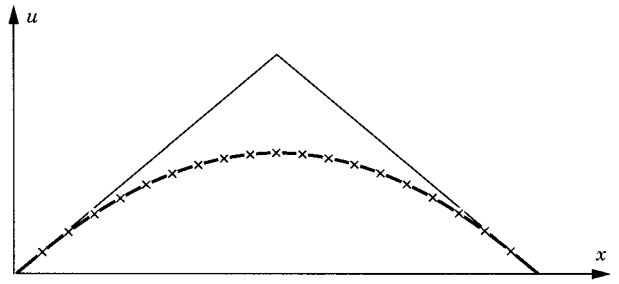


Figure 2

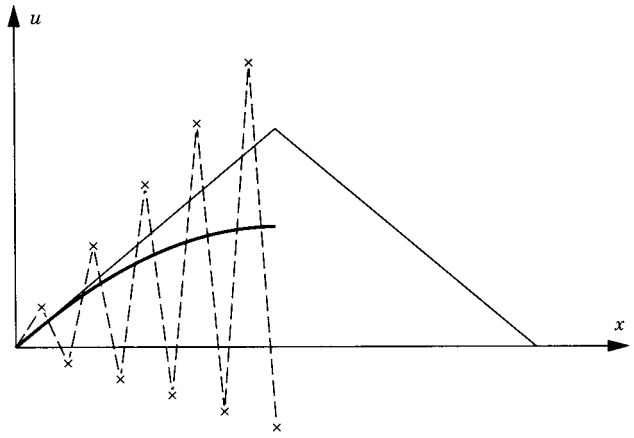


Figure 3

STABILITY CRITERION

The primary distinction between these two calculations turns out to be whether s is bigger or smaller than $\frac{1}{2}$. We might have gotten a suspicion of this from the scheme (2) itself, because when $s < \frac{1}{2}$, the coefficients in (2) are positive. But to actually demonstrate that this is the stability condition, we *separate the variables in the difference equation*. Thus we look for solutions of equation (2) of the form

$$u_j^n = X_j T_n. \tag{4}$$

Thus

$$\frac{T_{n+1}}{T_n} = 1 - 2s + s\frac{X_{j+1} + X_{j-1}}{X_j}. \tag{5}$$

Both sides of (4) must be a constant ξ independent of j and n. Therefore,

$$\boxed{T_n = \xi^n T_0} \tag{6}$$

and

$$s\frac{X_{j+1} + X_{j-1}}{X_j} + 1 - 2s = \xi. \tag{7}$$

To solve the spatial equation (7), we argue that it is a discrete version of a second-order ODE which has sine and cosine solutions. Therefore, we guess solutions of (7) of the form

$$X_j = A\cos j\theta + B\sin j\theta$$

for some θ, where A and B are arbitrary. The boundary condition $X_0 = 0$ at $j = 0$ implies that $A = 0$. So we can freely set $B = 1$. Then $X_j = \sin j\theta$.

Furthermore, the boundary condition $X_J = 0$ at $j = J$ implies that $\sin J\theta = 0$. Thus $J\theta = k\pi$ for some integer k. But the discretization into J equal intervals of length Δx means that $J = \pi/\Delta x$. Therefore, $\theta = k\,\Delta x$ and

$$\boxed{X_j = \sin(jk\Delta x).} \tag{8}$$

Now (7) takes the form

$$s\frac{\sin((j+1)k\Delta x) + \sin((j-1)k\Delta x)}{\sin(jk\Delta x)} + 1 - 2s = \xi$$

or

$$\boxed{\xi = \xi(k) = 1 - 2s[1 - \cos(k\Delta x)].} \tag{9}$$

According to (6), the growth in time $t = n\,\Delta t$ at the *wave number* k is governed by the powers $\xi(k)^n$. *So*

unless $|\xi(k)| \le 1$ *for all* k, *the scheme is unstable*

and could not possibly approximate the true (exact) solution. (Recall that the true solution tends to zero as $t \to \infty$.) Now we analyze (9) to determine whether $|\xi(k)| \le 1$ or not. Since the factor $1 - \cos(k\Delta x)$ ranges between 0 and 2, we have $1 - 4s \le \xi(k) \le 1$. So stability requires that $1 - 4s \ge -1$, which means that

$$\boxed{\frac{\Delta t}{(\Delta x)^2} = s \le \frac{1}{2}.} \tag{10}$$

Thus (10) is the condition required for stability.

This condition explains the instability that we observed in Section 8.1. It means that in practice the time steps must be taken quite short. For instance, if $\Delta x = 0.01$, an apparently reasonable choice, then Δt can be at most 0.00005. Then solving up to time $t = 1$ would require 20,000 time steps!

The analysis above shows that it is precisely the wave number k for which $\xi(k) = -1$, which is the most dangerous for stability. That critical situation happens when $\cos(k\Delta x) = -1$, that is, when $k = \pi/\Delta x$. In practice, this is a fairly high wave number.

By the way, the complete solution of the difference scheme (2), together with the discrete boundary conditions, is the "Fourier series"

$$u_j^n = \sum_{k=-\infty}^{\infty} b_k \sin(jk\Delta x)\,[\xi(k)]^n. \tag{11}$$

Let's see how it could be that this "discrete" series converges to the "true" series (3). In fact, the Taylor series of (9) is

$$\xi(k) = 1 - 2sk^2(\Delta x)^2/2! + \cdots \simeq 1 - k^2\Delta t$$

if $k\Delta x$ is small. Taking the nth power and letting $j\Delta x = x$ and $n\Delta t = t$, we have

$$\xi(k)^n \simeq (1 - k^2\Delta t)^{t/\Delta t} \simeq e^{-k^2 t}$$

in the limit as $\Delta t \to 0$, using the well-known limit for the exponential. So the series (11) looks like it tends to the series (3), as it should. Of course, this could not possibly be a proof of convergence (since we know it does not even converge at all if $s > \frac{1}{2}$). An actual proof for $s \leq \frac{1}{2}$, which we omit, would require a careful analysis of the approximations.

The example discussed above indicates that the general procedure to determine stability in a diffusion or wave problem is to separate the variables in the difference equation. For the time factor we obtain a simple equation like (6) which has an *amplification factor* $\xi(k)$. In the analysis above we used the stability condition $|\xi(k)| \leq 1$. More precisely, it can be shown that the correct condition necessary for stability is

$$\boxed{|\xi(k)| \leq 1 + O(\Delta t) \qquad \text{for all } k} \tag{12}$$

and for small Δt. (We omit the proof.) This is the *von Neumann stability condition* [RM]. The extra term in (12) is irrelevant for the example above but important for problems where the exact solution may grow in time (as in Exercise 11).

In practice we could go more quickly from (7) to (9) simply by assuming that

$$\boxed{X_j = (e^{ik\Delta x})^j} \tag{13}$$

is an exponential. (*This is the procedure to be followed in doing the exercises.*) Plugging (13) into (7), we immediately have

$$\xi = 1 - 2s + s(e^{ik\Delta x} + e^{-ik\Delta x}).$$

Thus we again recover equation (9) for the amplification factor ξ.

NEUMANN BOUNDARY CONDITIONS

Suppose that the interval is $0 \leq x \leq l$ and the boundary conditions are

$$u_x(0, t) = g(t) \qquad \text{and} \qquad u_x(l, t) = h(t).$$

Although the simplest approximations would be

$$\frac{u_1^n - u_0^n}{\Delta x} = g^n \qquad \text{and} \qquad \frac{u_J^n - u_{J-1}^n}{\Delta x} = h^n,$$

they would introduce local truncation errors of order $O(\Delta x)$, bigger than the $O(\Delta x)^2$ errors in the equation. To introduce $O(\Delta x)^2$ errors only, we prefer to use centered differences for the derivatives on the boundary.

To accomplish this, we introduce "ghost points" u_{-1}^n and u_{J+1}^n in addition to $u_0^n, \ldots, u_J^n$. The discrete boundary conditions then are

$$\frac{u_1^n - u_{-1}^n}{2\,\Delta x} = g^n \qquad \text{and} \qquad \frac{u_{J+1}^n - u_{J-1}^n}{2\,\Delta x} = h^n. \tag{14}$$

At the nth time step, we can calculate $u_0^n, \ldots, u_J^n$ using the scheme for the PDE, and then calculate the values at the ghost points using (14).

CRANK-NICOLSON SCHEME

We could try to avoid the restrictive stability condition (10) by using another scheme. There is a class of schemes that is stable no matter what the value of s. In fact, let's denote the centered second difference by

$$\frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{(\Delta x)^2} = (\delta^2 u)_j^n.$$

Pick a number θ *between* 0 *and* 1. Consider the scheme

$$\boxed{\frac{u_j^{n+1} - u_j^n}{\Delta t} = (1-\theta)(\delta^2 u)_j^n + \theta(\delta^2 u)_j^{n+1}.} \tag{15}$$

We'll call it *the* θ *scheme*. If $\theta = 0$, it reduces to the previous scheme. If $\theta > 0$, the scheme is *implicit*, since u^{n+1} appears on both sides of the equation. Therefore, (15) means that we solve a set ($n = 1$) of algebraic linear equations to get u_j^1, another set ($n = 2$) to get u_j^2, and so on.

Let us analyze the stability of this scheme by plugging in a separated solution

$$u_j^n = (e^{ik\Delta x})^j (\xi(k))^n$$

as before. Then

$$\xi(k) = \frac{1 - 2(1-\theta)s(1 - \cos k\Delta x)}{1 + 2\theta s(1 - \cos k\Delta x)},$$

where $s = \Delta t/(\Delta x)^2$ (see Exercise 9).

Again we look for the stability condition: $|\xi(k)| \le 1$ for all k. It is always true that $\xi(k) \le 1$, but the condition $\xi(k) \ge -1$ requires that

$$s(1 - 2\theta)(1 - \cos k\Delta x) \le 1.$$

(Why?) If $1 - 2\theta \leq 0$, it is always true! This means that

$$\boxed{\text{if } \tfrac{1}{2} \leq \theta \leq 1, \text{ there is no restriction on the size of } s} \tag{16}$$

for stability to hold. Such a scheme is called *unconditionally stable*.

The special case $\theta = \frac{1}{2}$ is called the *Crank–Nicolson scheme*. It has the template

$$\begin{array}{ccc} \frac{1}{2}\frac{s}{1+s}\bullet & * & \bullet\frac{1}{2}\frac{s}{1+s} \\ \frac{1}{2}\frac{s}{1+s}\bullet & \frac{1-s}{1+s}\bullet & \bullet\frac{1}{2}\frac{s}{1+s} \end{array}$$

On the other hand, in case $\theta < \frac{1}{2}$, a necessary condition for stability is $s \leq (2 - 4\theta)^{-1}$. Thus (15) is expected to be a stable scheme if

$$\boxed{\frac{\Delta t}{(\Delta x)^2} = s < \frac{1}{2 - 4\theta}.} \tag{17}$$

EXERCISES

1. (a) Solve the problem $u_t = u_{xx}$ in the interval $[0, 4]$ with $u = 0$ at both ends and $u(x, 0) = x(4 - x)$, using the forward difference scheme with $\Delta x = 1$ and $\Delta t = 0.25$. Calculate four time steps (up to $t = 1$).
   (b) Do the same with $\Delta x = 0.50$ and $\Delta t = 0.0625 = \frac{1}{16}$. Calculate four time steps (up to $t = 0.25$).
   (c) Compare your answers with each other. How close are they at $x = 2.0$, $t = 0.25$?
2. Do the same with $\Delta x = 1$ and $\Delta t = 1$. Calculate by hand or by computer up to $t = 7$.
3. Solve $u_t = u_{xx}$ in the interval $[0, 5]$ with $u(0, t) = 0$ and $u(5, t) = 1$ for $t \geq 0$, and with $u(x, 0) = 0$ for $0 < x < 5$.
   (a) Compute $u(3, 3)$ using the mesh sizes $\Delta x = 1$ and $\Delta t = 0.5$.
   (b) Write the exact solution as an infinite series. Calculate $u(3, 3)$ to three decimal places exactly. Compare with your answer in (a).
4. Solve by hand the problem $u_t = u_{xx}$ in the interval $[0, 1]$ with $u_x = 0$ at both ends. Use the forward scheme (2) for the PDE, and the scheme (14) for the boundary conditions. Assume $\Delta x = \frac{1}{5}$, $\Delta t = \frac{1}{100}$, and start with the initial data: 0 0 64 0 0 0. Compute for four time steps.
5. Using the forward scheme (2), solve $u_t = u_{xx}$ in $[0, 5]$ with the mixed boundary conditions $u(0, t) = 0$ and $u_x(5, t) = 0$ for $t \geq 0$, and the initial condition $u(x, 0) = 25 - x^2$ for $0 < x < 5$. Use $\Delta x = 1$ and $\Delta t = \frac{1}{2}$. Compute $u(3, 3)$ approximately.

6. Do the same with the conditions $u_x(0, t) = u(5, t) = 0$ and $u(x, 0) = x$.
7. Show that the local truncation error in the Crank-Nicolson scheme is $O((\Delta x)^2 + (\Delta t)^2)$.
8. (a) Write down the Crank–Nicolson scheme ($\theta = \frac{1}{2}$) for $u_t = u_{xx}$.
   (b) Consider the solution in the interval $0 \le x \le 1$ with $u = 0$ at both ends. Assume $u(x, 0) = \phi(x)$ and $\phi(1 - x) = \phi(x)$. Show, using uniqueness, that the exact solution must be even around the midpoint $x = \frac{1}{2}$. [That is, $u(x, t) = u(1 - x, t)$.]
   (c) Let $\Delta x = \Delta t = \frac{1}{6}$. Let the initial data be 0 0 0 1 0 0 0. Compute the solution by the Crank–Nicolson scheme for one time step ($t = \frac{1}{6}$). (*Hint:* Use part (b) to halve the computation.)
9. For the θ scheme (15) for the diffusion equation, provide the details of the derivation of the stability conditions (16) and (17).
10. For the diffusion equation $u_t = u_{xx}$, use centered differences for both u_t and u_{xx}.
    (a) Write down the scheme. Is it explicit or implicit?
    (b) Show that it is unstable, no matter what Δx and Δt are.
11. Consider the equation $u_t = au_{xx} + bu$, where a and b are constants and $a > 0$. Use forward differences for u_t, use centered differences for u_{xx}, and use bu_j^n for the last term.
    (a) Write the scheme. Let $s = \Delta t/(\Delta x)^2$.
    (b) Find the condition on s for numerical stability. (*Hint:* check condition (12).)
12. (a) Solve by hand the nonlinear PDE $u_t = u_{xx} + (u)^3$ for all x using the standard forward difference scheme with $(u)^3$ treated as $(u_j^n)^3$. Use $s = \frac{1}{4}$, $\Delta t = 1$, and initial data $u_j^0 = 1$ for $j = 0$ and $u_j^0 = 0$ for $j \neq 0$. Solve for u_0^3.
    (b) Compare your answer to the same problem without the nonlinear term.
    (c) Exactly solve the ODE $dv/dt = (v)^3$ with the condition $v(0) = 1$. Use it to explain why u_0^3 is so large in part (a).
    (d) Repeat part (a) with the same initial data but for the PDE $u_t = u_{xx} - (u)^3$. Compare with the answer in part (a) and explain.
13. Consider the following scheme for the diffusion equation:

$$\frac{u_j^{n+1} - u_j^{n-1}}{2\,\Delta t} = \frac{u_{j+1}^n + u_{j-1}^n - u_j^{n+1} - u_j^{n-1}}{(\Delta x)^2}.$$

    It uses a centered difference for u_t and a modified form of the centered difference for u_{xx}.
    (a) Solve it for u_j^{n+1} in terms of s and the previous time steps.
    (b) Show that it is stable for all s.
14. (a) Formulate an explicit scheme for $u_t = u_{xx} + u_{yy}$.

(b) What is the stability condition for your scheme in terms of $s_1 = \Delta t/(\Delta x)^2$ and $s_2 = \Delta t/(\Delta y)^2$?

15. Formulate the Crank-Nicolson scheme for $u_t = u_{xx} + u_{yy}$.

8.3 APPROXIMATIONS OF WAVES

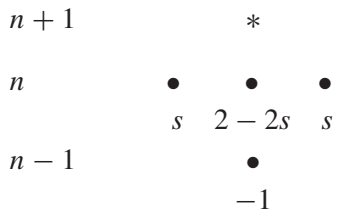
In this section we continue our discussion of finite difference approximations for some very simple PDEs. Although the PDEs are simple, the methods we develop can be used for more difficult, even nonlinear, equations. For the one-dimensional wave equation $u_{tt} = c^2 u_{xx}$ the simplest scheme is the one using centered differences for both terms:

$$\frac{u_j^{n+1} - 2u_j^n + u_j^{n-1}}{(\Delta t)^2} = c^2 \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{(\Delta x)^2}. \tag{1}$$

It is explicit since the $(n+1)$st time step appears only on the left side. Thus

$$\boxed{u_j^{n+1} = s\left(u_{j+1}^n + u_{j-1}^n\right) + 2(1-s)u_j^n - u_j^{n-1},} \tag{2}$$

where we now denote $s = c^2(\Delta t)^2/(\Delta x)^2$. Its template diagram is



Notice that the value at the $(n+1)$st time step depends on the *two* previous steps, because the wave equation has time derivatives of second order. Therefore, the first *two* rows u_j^0 and u_j^1 must be given as initial conditions.

Example 1.

If we pick $s = 2$, the scheme simplifies to

$$u_j^{n+1} = 2\left(u_{j+1}^n + u_{j-1}^n - u_j^n\right) - u_j^{n-1} \tag{3}$$

and it is easy to compute by hand the solution shown in Figure 1, given its first two rows. This horrendous solution bears no relationship to the true solution of the wave equation, which is a pair of waves traveling to the left and right, $u(x,t) = \frac{1}{2}[\phi(x+ct) + \phi(x-ct)]$. The scheme for $s = 2$ is highly unstable. □

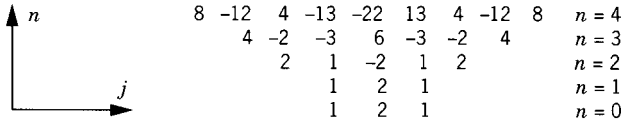


Figure 1

Example 2.

For $s = 1$ we have $\Delta x = c\,\Delta t$ and the scheme

$$\boxed{u_j^{n+1} = u_{j+1}^n + u_{j-1}^n - u_j^{n-1}.} \tag{4}$$

The same initial data as above lead to the solution shown in Figure 2. This is an excellent approximation to the true solution! □

INITIAL CONDITIONS

How do we handle the initial conditions? We approximate the conditions $u(x, 0) = \phi(x)$ and $\partial u/\partial t(x, 0) = \psi(x)$ by

$$u_j^0 = \phi(j\Delta x), \qquad \frac{u_j^1 - u_j^{-1}}{2\,\Delta t} = \psi(j\Delta x). \tag{5}$$

This approximation is chosen to have a $O(\Delta x)^2$ local truncation error in order to match the $O(\Delta x)^2 + O(\Delta t)^2$ truncation error of the scheme (2). (If we only used a simpler approximation with a $O(\Delta x)$ error, the initial conditions would contaminate the solution with too big an error.) Let's abbreviate $\phi_j = \phi(j\Delta x)$ and $\psi_j = \psi(j\Delta x)$. Now (2) in the case $n = 0$ is

$$u_j^1 + u_j^{-1} = s\left(u_{j+1}^0 + u_{j-1}^0\right) + 2(1 - s)u_j^0.$$

Together with (5), this gives us the starting values

$$\begin{aligned} u_j^0 &= \phi_j, \\ u_j^1 &= \frac{s}{2}(\phi_{j+1} + \phi_{j-1}) + (1 - s)\phi_j + \psi_j \Delta t, \end{aligned} \tag{6}$$

the first two rows of the computation. Then we march ahead in time to get u_j^2, u_j^3, and so on, using (2).

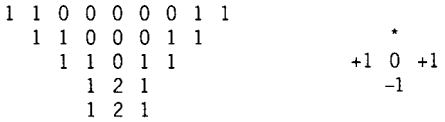


Figure 2

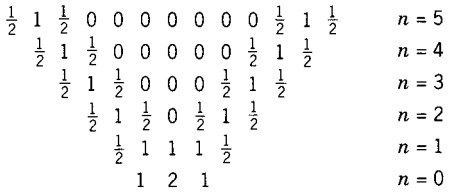


Figure 3

Example 3.

For instance, let the initial data be

$$\phi(x) = 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 2 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0$$

and $\psi(x) \equiv 0$. Let $s = 1$. Then from (6) we get the starting values (the first two rows)

$$\begin{array}{ccccccccccccccc} 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 1 & 1 & 1 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0. \end{array}$$

If we use (4), we get the solution shown in Figure 3. This is an even better approximation to the true solution than that shown in Figure 2. □

STABILITY CRITERION

Now let's analyze the stability by the method of Section 8.2. Again, a clue may be found in the values of the coefficients. None are negative if $s \leq 1$. Once again this simple observation turns out to be the correct stability condition. However, proceeding more logically, we separate the variables

$$u_j^n = (\eta)^j(\xi)^n \qquad \text{where } \eta = e^{ik\Delta x}.$$

From (1) we get

$$\xi + \frac{1}{\xi} - 2 = s\left(\eta + \frac{1}{\eta} - 2\right) = 2s\,[\cos(k\,\Delta x) - 1]. \tag{7}$$

Letting $p = s[\cos(k\Delta x) - 1]$ for the sake of brevity, (7) can be written as

$$\xi^2 - 2(1 + p)\xi + 1 = 0, \text{ which has the roots } \xi = 1 + p \pm \sqrt{p^2 + 2p}. \tag{8}$$

Note that $p \leq 0$. If $p < -2$, then $p^2 + 2p > 0$ and there are two real roots, one of which is less than -1. Thus for one of the roots we have $|\xi| > 1$, so that the scheme is unstable. On the other hand, if $p > -2$, then $p^2 + 2p < 0$ and there are two complex conjugate roots $1 + p \pm i\sqrt{-p^2 - 2p}$. These complex roots satisfy

$$|\xi|^2 = (1 + p)^2 - p^2 - 2p = 1.$$

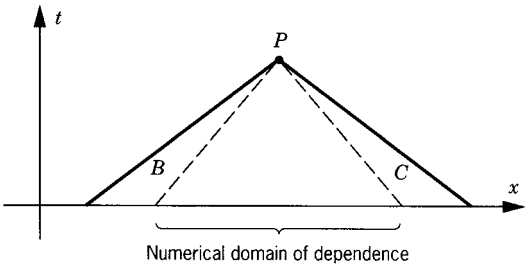


Figure 4

So $\xi = \cos\theta + i\sin\theta$ for some real number θ. In this case the solutions oscillate in time (just as they ought to for the wave equation). Finally, if $p = -2$, then $\xi = -1$.

Thus a *necessary condition for stability* is that $p \geq -2$ for all k. This means that

$$s \leq \frac{2}{1 - \cos(k\,\Delta x)}$$

for all k. Thus stability requires that

$$\boxed{s = c^2 \frac{(\Delta t)^2}{(\Delta x)^2} \leq 1.} \tag{9}$$

There is a nice way to understand this condition (9). At each time step Δt the values of the numerical solution spread out by one unit Δx. So the ratio $\Delta x/\Delta t$ is the propagation speed of the numerical scheme. The propagation speed for the exact wave equation is c. So the stability condition requires the numerical propagation speed to be at least as large as the continuous propagation speed. In Figure 4 we have sketched the domains of dependence of the true and the computed solutions for the case $c = 1$ and $\Delta t/\Delta x = 2$ (so that $s = 4$). The computed solution at the point P does not make use of the initial data in the regions B and C as it ought to. Therefore, the scheme leads to entirely erroneous values of the solution.

On the other hand, even the stable schemes do not do a very good job at resolving singularities in the true solution. For instance, one solution of the nice scheme (4) with $s = 1$ is shown in Figure 5. This initial condition is

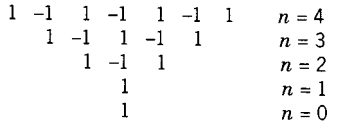


Figure 5

"singular" because it has a sudden up and down jump. The solution in Figure 5 isn't as unstable as the one in Figure 1, but it surely is a poor approximation to the true solution. (It's a good approximation only for someone who wears fuzzy glasses.) The difficulty here is that the initial function $\phi(x)$ has a significant "jump" at one point; the earlier cases illustrated in Figures 2 and 3 were at least slightly gradual. More sophisticated schemes must be used to solve problems with singularities, as in shock wave problems.

There are also implicit schemes for the wave equation (like the Crank–Nicolson scheme) but they are less urgently needed here since the stability condition (9) for the explicit scheme does not require Δt to be so much smaller than Δx.

Example 4.

For a more interesting PDE, let's consider the *nonlinear* wave equation

$$u_{tt} - \Delta u + u + [u]^7 = 0 \tag{10}$$

in three dimensions (x, y, z), where $[u]^7$ denotes the seventh power. Let (r, θ, ϕ) be the usual spherical coordinates. We shall make the calculation manageable by computing only those solutions that are independent of θ and ϕ. Then the equation takes the form

$$u_{tt} - u_{rr} - \frac{2}{r}u_r + u + [u]^7 = 0$$

by (6.1.7), which is a modification of the one-dimensional wave equation. To get rid of the middle term, it is convenient to change variables $v(r, t) = ru(r, t)$ to get

$$\begin{cases} v_{tt} - v_{rr} + v + r^{-6}[v]^7 = 0 \qquad (0 < r < \infty) \\ v(0, t) = 0. \end{cases} \tag{11}$$

The last condition comes from the definition of v.

Now we use the scheme (1) with $s = 1$ and with suitable additional terms to get

$$\frac{v_j^{n+1} - 2v_j^n + v_j^{n-1}}{(\Delta t)^2} = \frac{v_{j+1}^n - 2v_j^n + v_{j-1}^n}{(\Delta r)^2}$$

$$-\frac{1}{2}\left(v_j^{n+1} + v_j^{n-1}\right) - \frac{1}{8}(j\Delta r)^{-6}\frac{\left(v_j^{n+1}\right)^8 - \left(v_j^{n-1}\right)^8}{v_j^{n+1} - v_j^{n-1}} \tag{12}$$

One reason for this treatment of the additional terms is that this scheme has a constant energy (independent of n), an analog of the continuous energy of Section 2.2 (see Exercise 9).

Using the mesh sizes $\Delta r = \Delta t = 0.002$ and certain initial data, the computed solution is graphically presented in Figure 6 (see [SV]). The

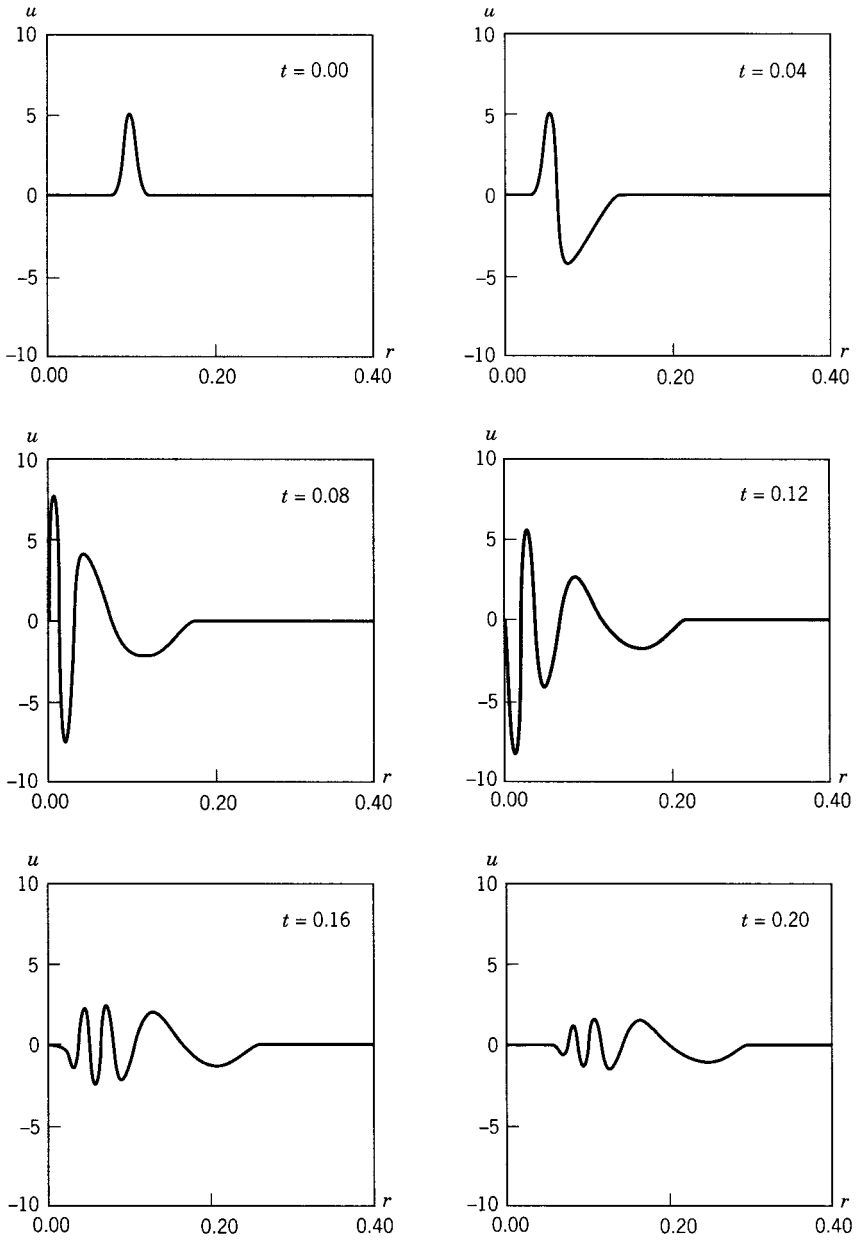


Figure 6

effect of the nonlinear term is visible in the oscillations of fairly large amplitude which reflect at the origin. □

EXERCISES

1. (a) Write the scheme (2) for the wave equation in the case $s = \frac{1}{4}$ and draw the template.
   (b) Compute the solution by hand for five time levels with the same starting values as in Figure 2.
   (c) Convince yourself that the computed solution is not too accurate but is "in the right ballpark." When interpreting the solution remember that $\Delta x/\Delta t = 2$.
2. Solve by hand for a few time steps the numerical scheme (2) for $u_{tt} = u_{xx}$, with $u(x, 0) \equiv 0$, with

$$\psi_j = \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 2 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0$$

   and with the starting scheme (6).
   (a) First use $\Delta t = 1$ and $\Delta x = 0.5$.
   (b) Then use $\Delta t = 1$ and $\Delta x = 1$.
   (c) Compare your answers to parts (a) and (b).
3. (a) Use the scheme (2) with $\Delta x = \Delta t = 0.2$ to approximately solve $u_{tt} = u_{xx}$ with $u(x, 0) = x^2$ and $u_t(x, 0) = 1$. Solve it in the region $\{0 \le t \le 1, |x| \le 2 - t\}$.
   (b) Solve the problem exactly and compare the exact and approximate solutions.
4. (a) Use the scheme (2) with $\Delta x = \Delta t = 0.25$ to solve $u_{tt} = u_{xx}$ approximately in the interval $0 \le x \le 1$ with $u = 0$ at both ends and $u(x, 0) = \sin \pi x$ and $u_t(x, 0) = 0$. Show that the solution is periodic.
   (b) Compare your answer to the exact solution. What is its period?
5. Solve by hand for a few time steps the equation $u_{tt} = u_{xx}$ in the finite interval $0 \le x \le 1$, with $u_x = 0$ at both ends, using $\Delta t = \Delta x = \frac{1}{6}$ and the initial conditions

$$u(x, 0) = \quad 0 \quad 0 \quad 0 \quad 1 \quad 2 \quad 1 \quad 0 \quad 0 \quad 0 \qquad \text{and} \qquad u_t(x, 0) \equiv 0.$$

   Use central differences for the boundary derivatives as in (8.2.14) and use second-order-accurate initial conditions as in (6). Do you see the reflections at the boundary?
6. Consider the wave equation on the half-line $0 < x < \infty$, with the boundary condition $u = 0$ at $x = 0$. With the starting values $u_4^0 = u_5^0 = u_4^1 = u_5^1 = 1$ and $u_j^0 = u_j^1 = 0$ for all other j ($j = 1, 2, \ldots$), compute the solution by hand up to 10 time steps. Observe the reflection at the boundary and compare with Section 3.2.

7. Solve by hand the nonlinear equation $u_{tt} = u_{xx} + u^3$ up to $t = 4$, using the same initial conditions as in Figure 3, replacing the cubic term by $(u_j^n)^3$, and using $\Delta x = \Delta t = 1$. What is the effect of the nonlinear term? Compare with the linear problem in Figure 3.
8. Repeat Exercise 7 by computer for the equation $u_{tt} = u_{xx} - u^3$ using an implicit scheme like (12) with $\Delta t = \Delta x = 1$.
9. Consider the scheme (12) for the nonlinear wave equation (10). Let the *discrete energy* be defined as

$$\frac{E_n}{\Delta r} = \frac{1}{2}\sum_j \left(\frac{v_j^{n+1} - v_j^n}{\Delta t}\right)^2 + \frac{1}{2}\sum_j \left(\frac{v_{j+1}^{n+1} - v_j^{n+1}}{\Delta r}\right)\left(\frac{v_{j+1}^n - v_j^n}{\Delta r}\right)$$
$$+ \frac{1}{4}\sum_j \left[\left(v_j^{n+1}\right)^2 + \left(v_j^n\right)^2\right] + \frac{1}{16}\sum_j \frac{\left(v_j^{n+1}\right)^8 + \left(v_j^n\right)^8}{(j\Delta r)^6}.$$

   By multiplying (12) by $\frac{1}{2}(v_j^{n+1} - v_j^{n-1})$, show that $E_n = E_{n-1}$. Conclude that E_n does not depend on n.
10. Consider the equation $u_t = u_x$. Use forward differences for both partial derivatives.
    (a) Write down the scheme.
    (b) Draw the template.
    (c) Find the separated solutions.
    (d) Show that the scheme is stable if $0 < \Delta t/\Delta x \leq 1$.
11. Consider the first-order equation $u_t + au_x = 0$.
    (a) Solve it exactly with the initial condition $u(x, 0) = \phi(x)$.
    (b) Write down the finite difference scheme which uses the forward difference for u_t and the centered difference for u_x.
    (c) For which values of Δx and Δt is the scheme stable?

8.4 APPROXIMATIONS OF LAPLACE'S EQUATION

For Dirichlet's problem in a domain of irregular shape, it may be more convenient to compute numerically than to try to find the Green's function. As with the other equations, the same ideas of numerical computation can easily be carried over to more complicated equations. For Laplace's equation

$$u_{xx} + u_{yy} = 0$$

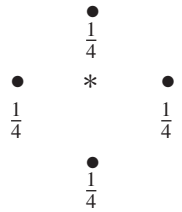
the natural approximation is that of centered differences,

$$\frac{u_{j+1,k} - 2u_{j,k} + u_{j-1,k}}{(\Delta x)^2} + \frac{u_{j,k+1} - 2u_{j,k} + u_{j,k-1}}{(\Delta y)^2} = 0. \qquad (1)$$

Here $u_{j,k}$ is an approximation to $u(j\Delta x, k\Delta y)$. The relative choice of mesh sizes turns out not to be critical so we just choose $\Delta x = \Delta y$. Then (1) can be written as

$$\boxed{u_{j,k} = \tfrac{1}{4}(u_{j+1,k} + u_{j-1,k} + u_{j,k+1} + u_{j,k-1}).} \tag{2}$$

Thus $u_{j,k}$ is the *average* of the values at the four neighboring sites. The template is



The scheme (2) has some nice properties. The most obvious one is the *mean value property*, the exact analog of the same property for the Laplace equation. In its discrete version (2), the difference equation and the mean value property become identical! It follows that a solution u_{jgk} cannot take its maximum or minimum value at an interior point, unless it is a constant; for otherwise it couldn't be the average of its neighbors. Thus, if (2) is valid in a region, the maximum and minimum values can be taken only at the boundary.

To solve the Dirichlet problem for $u_{xx} + u_{yy} = 0$ in D with given boundary values, we draw a grid covering D and approximate D by a union of squares (see Figure 1). Then the discrete solution is specified on the boundary of the "discrete region." Our task is to fill in the interior values so as to satisfy (2). In contrast to time-dependent problems, no marching method is available. If N is the number of interior grid points, *the equations* (2) *form a system of N linear equations in N unknowns*. For instance, if we divide x and y each into 100 parts, we get about 10,000 little squares. Thus N can be very large.

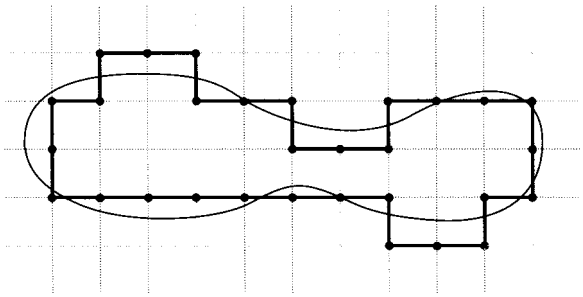


Figure 1

Figure 2

The system we get in this way has *exactly one* solution. To prove this, suppose that there were two solutions, $\{u_{j,k}\}$ and $\{v_{j,k}\}$, of (2) in D with identical boundary values. Their difference $\{u_{j,k} - v_{j,k}\}$ also satisfies (2) in D but with zero boundary values. By the maximum principle stated above, $u_{j,k} - v_{j,k} \leq 0$, and by the minimum principle, $u_{j,k} - v_{j,k} \geq 0$. Hence $u_{j,k} = v_{j,k}$. So there is at most one solution. But this means that the determinant of the linear system of N equations is not zero, which means that there exists exactly one solution.

Example 1.

As a baby example, consider solving (2) in the square with the boundary values indicated in Figure 2(*a*). This is a set of four linear equations, one for each interior point. The solution is given in Figure 2(*b*). Notice that each interior entry is indeed the average of its four neighbors. □

JACOBI ITERATION

In the absence of a marching method to solve (2), several techniques are available. One is called Jacobi iteration. We start from any reasonable first approximation $u_{j,k}^{(1)}$. Then we successively solve

$$u_{j,k}^{(n+1)} = \tfrac{1}{4}\left[u_{j+1,k}^{(n)} + u_{j-1,k}^{(n)} + u_{j,k+1}^{(n)} + u_{j,k-1}^{(n)}\right]. \tag{3}$$

It can be shown that $u_{j,k} = \lim u_{j,k}^{(n)}$ converges as $n \to \infty$ to a limit which is a solution of (2). It converges, however, very slowly and so Jacobi iteration is never used in practice. Since N is usually quite large, a more efficient method is needed.

It might be noticed that (3) is exactly the same calculation as if one were solving the two-dimensional heat equation $v_t = v_{xx} + v_{yy}$ using centered differences for v_{xx} and v_{yy} and using the forward difference for v_t, with $\Delta x = \Delta y$ and $\Delta t = (\Delta x)^2/4$ (see Exercise 11). In effect, we are solving the Dirichlet problem by taking the limit of the discretized $v(x, t)$ as $t \to \infty$.

GAUSS–SEIDEL METHOD

This method improves the rate of convergence. Here it is important to specify the order of operations. Let's compute $u_{j,k}^{(n+1)}$ one row at a time starting at

the bottom row and let's go from left to right. But every time a calculation is completed, we'll throw out the old value and update it by its newly calculated one. This procedure means that

$$u_{j,k}^{(n+1)} = \tfrac{1}{4}\Big[u_{j+1,k}^{(n)} + u_{j-1,k}^{\mathbf{(n+1)}} + u_{j,k+1}^{(n)} + u_{j,k-1}^{\mathbf{(n+1)}}\Big]. \tag{4}$$

The new values (with superscript $n+1$) are used to the *left* and *below* the (j, k) location. It turns out that Gauss–Seidel works about twice as fast as Jacobi.

SUCCESSIVE OVERRELAXATION

This method is still faster. It is the scheme

$$u_{j,k}^{(n+1)} = u_{j,k}^{(n)} + \omega\Big[u_{j+1,k}^{(n)} + u_{j-1,k}^{(n+1)} + u_{j,k+1}^{(n)} + u_{j,k-1}^{(n+1)} - 4u_{j,k}^{(n)}\Big]. \tag{5}$$

If $\omega = \frac{1}{4}$, it is the same as Gauss–Seidel. It is quite surprising that a different choice of ω could make a significant improvement, but it does. But how to choose the relaxation factor ω in practice is an art whose discussion we leave to more specialized texts. Note again that once we know that $u_{j,k} = \lim u_{j,k}^{(n)}$ exists, the limit must satisfy

$$u_{j,k} = u_{j,k} + \omega(u_{j+1,k} + u_{j-1,k} + u_{j,k+1} + u_{j,k-1} - 4u_{j,k})$$

and hence it satisfies (2).

EXERCISES

1. Set up the linear equations to find the four unknown values in Figure 2(a), write them in vector-matrix form, and solve them. You should deduce the answer in Figure 2(b).
2. Apply Jacobi iteration to the example of Figure 2(a) with zero initial values in the interior. Compute six iterations.
3. Apply four Gauss–Seidel iterations to the example of Figure 2(a).
4. Solve the example of Figure 2(a) but with the boundary conditions (by rows, top to bottom) 0, 48, 0, 0; 0, *, *, 24; 0, *, *, 0; 0, 0, 0, 0.
5. Consider the PDE $u_{xx} + u_{yy} = 0$ in the unit square $\{0 \le x \le 1, 0 \le y \le 1\}$ with the boundary conditions:

   $u = 0$ on $x = 0$, on $x = 1$, and on $y = 1$
   $u = 324\,x^2(1 - x)$ on $y = 0$.

   Calculate the approximation to the solution using finite differences (2) with the very coarse grid $\Delta x = \Delta y = \frac{1}{3}$.
   (*Hint:* You may use Figure 2 if you wish.)
6. (a) Write down the scheme using centered differences for the equation $u_{xx} + u_{yy} = f(x, y)$.

(b) Use it with $\Delta x = \Delta y = 0.5$ to solve the problem $u_{xx} + u_{yy} = 1$ in the square $0 \le x \le 1, 0 \le y \le 1$ with $u = 0$ on the boundary.
(c) Repeat with $\Delta x = \Delta y = \frac{1}{3}$.
(d) Compute the exact value at the center of the square and compare with your answer to part (b).

7. Solve $u_{xx} + u_{yy} = 0$ in the unit square $\{0 \le x \le 1, 0 \le y \le 1\}$ with the boundary conditions: $u(x, 0) = u(0, y) = 0,\ u(x, 1) = x,\ u(1, y) = y$. Use $\Delta x = \Delta y = \frac{1}{4}$, so that there are nine interior points for the scheme (2).
   (a) Use two steps of Jacobi iteration, with the initial guess that the value at each of the nine points equals 1.
   (b) Use two steps of Gauss–Seidel iteration, with the same initial guess.
   (c) Compare parts (a) and (b) and the exact solution.
8. Formulate a finite difference scheme for $u_{xx} + u_{yy} = f(x, y)$ in the unit square $\{0 \le x \le 1, 0 \le y \le 1\}$ with Neumann conditions $\partial u/\partial n = g(x, y)$ on the boundary. Use $u_{j,k}$ for $-1 \le j \le N + 1$ and $-1 \le k \le N + 1$ and use centered differences for the normal derivative, such as $(u_{j,N+1} - u_{j,N-1})/2\,\Delta y$. [That is, use ghost points as in (8.2.14).]
9. Apply Exercise 8 to approximately find the harmonic function in the unit square with the boundary conditions $u_x(0, y) = 0,\ u_x(1, y) = -1$, $u_y(x, 0) = 0, u_y(x, 1) = 1$. Formulate a Gauss–Seidel method of solving the difference scheme and compute two iterations with $\Delta x = \Delta y = \frac{1}{3}$. Compare with the exact solution $u = \frac{1}{2}y^2 - \frac{1}{2}x^2$. You may use a computer program.
10. Try to do the same with the boundary conditions $u_x(0, y) = 0$, $u_x(1, y) = 1, u_y(x, 0) = 0,\ u_y(x, 1) = 1$. What's wrong?
11. Show that performing Jacobi iteration (3) is the same as solving the two-dimensional diffusion equation $v_t = v_{xx} + v_{yy}$ using centered differences for v_{xx} and v_{yy} and using the forward difference for v_t, with $\Delta x = \Delta y$ and $\Delta t = (\Delta x)^2/4$.
12. Do the same (solving the diffusion equation) with $\Delta t = \omega(\Delta x)^2$ and compare with successive overrelaxation.

8.5 FINITE ELEMENT METHOD

All computational methods reduce PDEs to discrete form. But there are other methods besides finite differences. Here we briefly discuss the finite element method. The idea is to divide the domain into simple pieces (polygons) and to approximate the solution by extremely simple functions on these pieces. In one of its incarnations, the one we shall discuss, the simple pieces are triangles and the simple functions are linear. The finite element method was developed by engineers to handle curved or irregularly shaped domains. If D is a circle, say, they were having trouble using finite differences, which are not

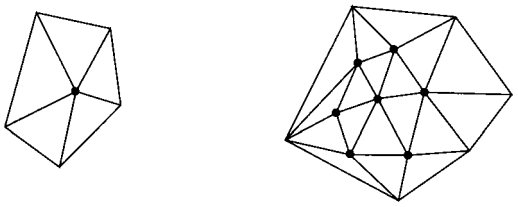


Figure 1

particularly efficient simply because a circle is not very accurately partitioned into rectangles.

Let's consider the Dirichlet problem for Poisson's equation in the plane

$$-\Delta u = f \quad \text{in } D, \quad u = 0 \quad \text{on bdy} D. \tag{1}$$

First, D is *triangulated*; that is, D is approximated by a region D_N which is the union of a finite number of triangles (see Figure 1). Let the interior vertices be denoted by $V_1, \ldots, V_N$.

Next, we pick N *trial functions*, $v_1(x, y), \ldots, v_N(x, y)$, one for each interior vertex. Each trial function $v_i(x, y)$ is chosen to equal 1 at "its" vertex V_i and to equal 0 at all the other vertices (see Figure 2). Inside each triangle, each trial function is a *linear* function: $v_i(x, y) = a + bx + cy$. (The coefficients a, b, c are different for each trial function and for each triangle.) This prescription determines $v_i(x, y)$ uniquely. In fact, its graph is simply a pyramid of unit height with its summit at V_i and it is identically zero on all the triangles that do not touch V_i.

We shall approximate the solution $u(x, y)$ by a linear combination of the $v_i(x, y)$:

$$\boxed{u_N(x, y) = U_1 v_1(x, y) + \cdots + U_N v_N(x, y).} \tag{2}$$

How do we choose the coefficients $U_1, \ldots, U_N$?

To motivate our choice we need a digression. Let's rewrite the problem (1) using Green's first identity [formula (G1) from Section 7.1]. We multiply

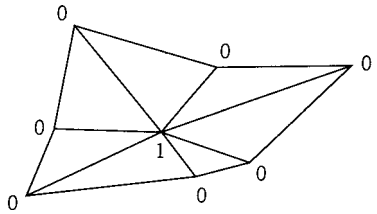


Figure 2

Poisson's equation by any function $v(x, y)$ that vanishes on the boundary. Then

$$\iint_D \nabla u \cdot \nabla v \, dx \, dy = \iint_D f v \, dx \, dy. \tag{3}$$

Rather than requiring (3) to be valid for $u_N(x, y)$ for *all* functions $v(x, y)$, we ask only that it be valid for the first N special trial functions $v = v_j \, (j = 1, \ldots, N)$. Thus, with $u(x, y) = u_N(x, y)$ and $v(x, y) = v_j(x, y)$, (3) becomes

$$\sum_{i=1}^{N} U_i \left(\iint_D \nabla v_i \cdot \nabla v_j \, dx \, dy \right) = \iint_D f v_j \, dx \, dy.$$

This is a system of N linear equations $(j = 1, \ldots, N)$ in the N unknowns $U_1, \ldots, U_N$. If we denote

$$m_{ij} = \iint_D \nabla v_i \cdot \nabla v_j \, dx \, dy \quad \text{and} \quad f_j = \iint_D f v_j \, dx \, dy, \tag{4}$$

then the system takes the form

$$\boxed{\sum_{i=1}^{N} m_{ij} U_i = f_j \qquad (j = 1, \ldots, N).} \tag{5}$$

The finite element method consists of calculating m_{ij} and f_j from (4) *and solving* (5). *The approximate value of the solution $u(x, y)$ then is given by* (2).

The trial functions v_j are completely explicit and depend only on the geometry. The approximate solution u_N automatically vanishes on the boundary of D_N. Notice also that, at a vertex $V_k = (x_k, y_k)$,

$$u_N(x_k, y_k) = U_1 v_1(x_k, y_k) + \cdots + U_N v_N(x_k, y_k) = U_k,$$

since $v_i(x_{k,} y_k)$ equals 0 for $i \neq k$ and equals 1 for $i = k$. Thus *the coefficients are precisely the values of the approximate solution at the vertices*.

Another way to understand $u_N(x, y)$ is that it is a continuous and piecewise-linear function (linear on each triangle), simply because it is a sum of such functions. In fact, it is the unique piecewise-linear continuous function on the triangulation such that $u_N(x_k, y_k) = U_k \, (k = 1, \ldots, N)$.

Notice also that the matrix m_{ij} is "sparse": $m_{ij} = 0$ whenever V_i and V_j are not neighboring vertices. Furthermore, for a pair of neighboring vertices, m_{ij} is easy to compute since each $v_i(x, y)$ is linear on each triangle.

Triangulations with linear functions are not the only versions of the finite element method used in practice. Two other versions in two variables are as follows.

(i) *Bilinear elements on rectangles:* D is divided into rectangles on each of which the solution is approximated using trial functions

$v_i(x, y) = a + bx + cy + dxy$. Each trial function is associated with a corner of a rectangle.

(ii) *Quadratic elements on triangles:* D is triangulated and the trial functions have the form $v_i(x, y) = a + bx + cy + dx^2 + exy + fy^2$. Each trial function is associated with one of the six "nodes" of a triangle, namely, the three vertices and the midpoints of the three sides.

For a reference, see [TR].

As a further example, consider solving the diffusion equation

$$u_t = \kappa u_{xx} + f(x, t); \qquad u = 0 \text{ at } x = 0, l; \quad u = \phi(x) \text{ at } t = 0.$$

Suppose, for simplicity, that l is an integer. Partition the interval $[0, l]$ into l equal subintervals. We assign the trial function $v_j(x)$ to each of the $N = l - 1$ interior vertices, where $v_j(x)$ is the linear element of Exercise 3. Now we multiply the diffusion equation by any function $v(x)$ that vanishes on the boundary. Integrating by parts, we get

$$\frac{d}{dt}\int_0^l uv\,dx = -\kappa\int_0^l \frac{\partial u}{\partial x}\frac{dv}{dx}\,dx + \int_0^l f(x,t)\,v(x)\,dx. \tag{6}$$

In order to use the finite-element method, we look for a solution of the form

$$u(x,t) = \sum_{i=1}^{N} U_i(t)\,v_i(x)$$

and we merely require (6) to hold for $v = v_1, \ldots, v_N$. Then

$$\sum_{i=1}^{N}\left(\int_0^l v_i v_j dx\right)\frac{dU_i}{dt} = -\kappa\sum_{i=1}^{N}\left(\int_0^l \frac{dv_i}{dx}\frac{dv_j}{dx}dx\right)U_i(t) + \int_0^l f(x,t)v_j(x)\,dx.$$

This is a system of ODEs for $U_1, \ldots, U_N$ that can be written as a vector equation as follows.

Let U be the column vector $[U_1, \ldots, U_N]$ and let F be the column vector $[F_1(t), \ldots, F_N(t)]$ with $F_j(t) = \int_0^l f(x,t)v_j(x)dx$. Let M be the matrix with entries m_{ij} and K be the matrix with entries k_{ij} where

$$k_{ij} = \int_0^l v_i v_j\,dx, \qquad m_{ij} = \int_0^l \frac{dv_i}{dx}\frac{dv_j}{dx}\,dx.$$

Then the system of N ODEs in N unknowns takes the simple vector form

$$K\frac{dU}{dt} = -\kappa MU(t) + F(t). \tag{7}$$

M is called the stiffness matrix, K the mass matrix, and F the forcing vector. In Exercise 3(a), the stiffness and mass matrices are computed to be

$$M = \begin{pmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \cdots & 0 \\ & \cdots & & & \\ 0 & \cdots & 0 & -1 & 2 \end{pmatrix}, \quad K = \begin{pmatrix} \frac{2}{3} & \frac{1}{6} & 0 & \cdots & 0 \\ \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & \cdots & 0 \\ & \cdots & & & \\ 0 & \cdots & 0 & \frac{1}{6} & \frac{2}{3} \end{pmatrix}$$

The matrices M and K have two important features. They are sparse and they depend only on the trial functions. So they remain the same as we change the data. We also have the initial condition

$$U_i(0) = \Phi_i \equiv \int_0^l \phi(x) v_i(x)\, dx. \tag{8}$$

This ODE system (7)-(8) can be solved numerically by any of a number of methods. One simple method is Euler's. One chooses $t_p = p\Delta t$ for $p = 0, 1, 2, \ldots$ and then solves

$$U^{(p+1)} = U^{(p)} + \Delta t\, W^{(p)},$$
$$K W^{(p)} = -\kappa M U^{(p)} + F(t_p).$$

Another method is the backwards Euler method, in which we solve

$$K\left[\frac{U^{(p+1)} - U^{(p)}}{\Delta t}\right] = -\kappa M U^{(p+1)} + F(t_{p+1}).$$

This is the same as

$$[K + \kappa \Delta t M]\, U^{(p+1)} = K U^{(p)} + \Delta t\, F(t_{p+1}),$$

which is solved recursively for $U^{(1)}, U^{(2)}, \ldots$.

EXERCISES

1. Consider the problem $u_{xx} + u_{yy} = -4$ in the unit square with $u(0, y) = 0$, $u(1, y) = 0$, $u(x, 0) = 0$, $u(x, 1) = 0$. Partition the square into four triangles by drawing its diagonals. Use the finite element method to find the approximate value $u(\frac{1}{2}, \frac{1}{2})$ at the center.
2. (a) Find the area A of the triangle with three given vertices (x_1, y_1), (x_2, y_2), and (x_3, y_3).
   (b) Let (x_1, y_1) be a vertex in the finite element method and let $v(x, y)$ be its trial function. Let T be one of the triangles with that vertex and let (x_2, y_2) and (x_3, y_3) be its other two vertices. Find the formula for $v(x, y)$ on T.
3. (*Linear elements on intervals*) In one dimension the geometric building blocks of the finite element method are the intervals. Let the trial function $v_j(x)$ be the "tent" function defined by $v_j(x) = 1 - j + x$ for $j - 1 \le x \le j$, $v_j(x) = 1 + j - x$ for $j \le x \le j + 1$, and $v_j(x) = 0$ elsewhere.

That is, $v_j(x)$ is continuous and piecewise-linear with $v_j(j) = 1$ and $v_j(k) = 0$ for all integers $k \neq j$.

(a) Show that $\int [v_j(x)]^2 dx = 2$ and $\int v_j(x) v_{j+1}(x)\, dx = -1$.

(b) Deduce that the one-dimensional analog of the matrix m_{ij} is the tridiagonal matrix with 2 along the diagonal and -1 next to the diagonal.

4. (*Finite elements for the wave equation*) Consider the problem $u_{tt} = u_{xx}$ in $[0, l]$, with $u = 0$ at both ends, and some initial conditions. For simplicity, suppose that l is an integer and partition the interval into l equal sub-intervals. Each of the $l - 1 = N$ interior vertices has the trial function defined in Exercise 3. The approximate solution is defined by the formula $u_N(x) = U_1(t)v_1(x) + \cdots + U_N(t)v_N(x)$, where the coefficients are unknown functions of t.

(a) Show that a reasonable requirement is that

$$\sum_{i=1}^{N} U_i''(t) \int_0^l v_i(x) v_j(x)\, dx + \sum_{i=1}^{N} U_i(t) \int_0^l \frac{\partial v_i}{\partial x} \frac{\partial v_j}{\partial x} dx = 0$$

for $j = 1, \ldots, N$.

(b) Show that the finite element method reduces in this case to a system of ODEs: $K d^2U/dt^2 + MU = 0$ with an initial condition $U(0) = \Phi$. Here K and M are $N \times N$ matrices, $U(t)$ is an N-vector function, and Φ is an N-vector.

5. (*Bilinear elements on rectangles*) On the rectangle with vertices $(0, 0), (A, 0), (0, B)$, and (A, B), find the bilinear function $v(x, y) = a + bx + cy + dxy$ with the values U_1, U_2, U_3, and U_4, respectively.