LESSON 37

Numerical Solutions (Elliptic Problems)

PURPOSE OF LESSON: To show how a partial differential equation can be changed to a system of algebraic equations by replacing the *partial derivatives* in the differential equation with their *finite-difference approximations*. The system of algebraic equations can then be solved numerically by an iterative process in order to obtain an approximate solution to the PDE.

It is also pointed out that the reader can obtain an existing computer package (ELLPACK) that will solve general elliptic problems.

So far, we have studied several techniques for solving linear PDEs. However, most of the equations we've attacked were reasonably simple, had reasonably simple BCs, and had reasonably shaped domains. But many problems cannot be simplified to fit this general mold and must be solved by numerical approximations. Over the past ten years, scientists and engineers have begun to attack many more problems as a result of more computing power and more sophisticated numerical methods. Several new techniques have been developed to take advantage of high-speed computing machinery. Nonlinear problems in fluid dynamics, elasticity, and potential theory involving two and three dimensions are being solved today that were not even considered ten years ago.

There are several procedures that come under the name of numerical methods. The reader can look in reference 1 of the recommended reading for a more complete discussion of these techniques. This lesson and the next two show how the very popular *finite-difference method* can be used to solve elliptic, hyperbolic, and parabolic equations.

To begin, we introduce the idea of *finite differences*. We then show how to use these finite differences to solve a Dirichlet problem inside a square.

Finite-Difference Approximations

First, we recall the Taylor series expansion of a function $f(x)$

$$f(x + h) = f(x) + f'(x)h + \frac{f''(x)}{2!}h^2 + \dots$$

If we *truncate* this series after two terms, we have the approximation

$$f(x + h) \cong f(x) + f'(x)h$$

Hence, we can solve for $f'(x)$

$$f'(x) \cong \frac{f(x + h) - f(x)}{h} \tag{37.1}$$

which is called the **forward-difference approximation** to the first derivative $f'(x)$.

We could also replace h by $-h$ in the Taylor series and arrive at the **backward-difference approximation**

$$f'(x) \cong \frac{f(x) - f(x - h)}{h} \tag{37.2}$$

or by subtracting

$$f(x - h) \cong f(x) - f'(x)h$$

from

$$f(x + h) \cong f(x) + f'(x)h$$

we can obtain the **central-difference approximation**

$$f'(x) \cong \frac{1}{2h}[f(x + h) - f(x - h)] \tag{37.3}$$

By retaining *another term* in the Taylor series, this type of analysis can be extended to arrive at the central-difference approximation of the second derivative $f''(x)$

$$f''(x) \cong \frac{1}{h^2}[f(x + h) - 2f(x) + f(x - h)] \tag{37.4}$$

We now extend the finite-difference approximations to *partial derivatives*. If we begin with the Taylor series expansion in two variables

$$u(x + h, y) = u(x,y) + u_x(x,y)h + u_{xx}(x,y)\frac{h^2}{2!} + \dots$$

$$u(x - h, y) = u(x,y) - u_x(x,y)h + u_{xx}(x,y)\frac{h^2}{2!} - \dots$$

we can deduce the following:

$$u_x(x,y) \cong \frac{u(x + h, y) - u(x,y)}{h} \qquad \text{(Forward difference)}$$

$$u_{xx}(x,y) \cong \frac{1}{h^2}[u(x + h, y) - 2u(x,y) + u(x - h, y)]$$

$$u_y(x,y) \cong \frac{u(x, y + k) - u(x,y)}{k}$$

$$u_{yy}(x,y) \cong \frac{1}{k^2}[u(x, y + k) - 2u(x,y) + u(x, y - k)]$$

Which approximation to use (forward, central, or backward) depends on the problem, but in this lesson, we will use the central-difference approximation.

To illustrate how to use these approximations, we consider the simple Dirichlet problem.

Dirichlet Problem Solved by the Finite-Difference Method

(37.5)

PDE	$u_{xx} + u_{yy} = 0$	$0 < x < 1 \quad 0 < y < 1$
BCs	$u = 0$	On the top and sides of the square
	$u(x,0) = \sin(\pi x)$	$0 \le x \le 1$

We begin this problem by drawing the grid system on the xy-plane shown in Figure 37.1.

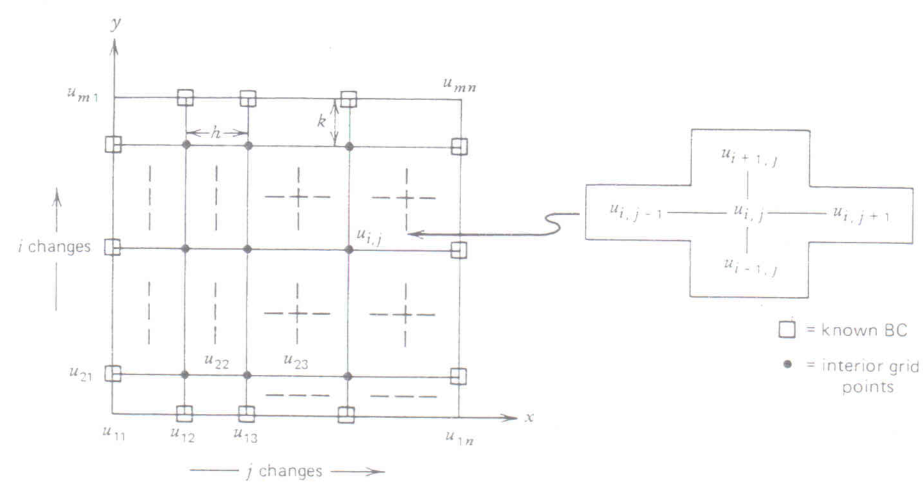

FIGURE 37.1 Grid lines for the Dirichlet problem inside a square.

It is also convenient (especially if we want to use a computer) to use the following notation:

$$u(x,y) = u_{i,j}$$

$$u(x, y + k) = u_{i+1,j}$$

$$u(x,y-k) = u_{i-1,j}$$

$$u(x+h,y) = u_{i,j-1}$$

$$u(x-h,y) = u_{i,j-1}$$

$$u_x(x,y) = \frac{1}{2h}(u_{i,j+1} - u_{i,j-1})$$

$$u_y(x,y) = \frac{1}{2k}(u_{i-1,j} - u_{i-1,j})$$

$$u_{xx}(x,y) = \frac{1}{h^2}(u_{i,j+1} - 2u_{i,j} + u_{i,j-1}) \qquad \text{(Central-difference formulas)}$$

$$u_{yy}(x,y) = \frac{1}{k^2}(u_{i+1,j} - 2u_{i,j} + u_{i-1,j})$$

Our strategy for solving this Dirichlet problem is to replace the partial derivatives in Laplace's equation

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

by their finite-difference approximations. Doing this and using the compact notation $u_{i,j}$, we have the following *difference equation:*

$$\nabla^2 u = \frac{1}{h^2}(u_{i,j+1} - 2u_{i,j} + u_{i,j-1}) + \frac{1}{k^2}(u_{i-1,j} - 2u_{i,j} + u_{i-1,j}) = 0$$

By letting the two discretization sizes h and k be the same, Laplace's equation is replaced by

$$(u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j}) = 0 \qquad (37.6)$$

or solving for $u_{i,j}$

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

Note that here the $u_{i,j}$'s would stand for the solution at the *interior* grid points. This last equation says that we can approximate the solution $u_{i,j}$ by *averaging* the solution at the *four neighboring grid points*. Hence, we can devise a numerical strategy for solving the problem.

Numerical Algorithm for Solving the Dirichlet Problem (Liebmann's method)

STEP 1 Seek the solution $u_{i,j}$ at the interior grid points by setting them equal to the *average* of all the BCs (reasonable start).

STEP 2 Systematically run over all the *interior* grid points, replacing the old estimates by the average of its four neighbors. It doesn't make much difference in what order this process is carried out, but, generally, it is done in a row by row (or column by column) manner. After a few iterations, this process will converge to an approximate solution of the problem. The rate of change of this process is generally slow but can be speeded up in a number of ways; interested readers should consult reference 1 of the recommended reading.

This completes the discussion of our Dirichlet problem; the reader is asked to carry out three iterations of Liebmann's method in the problems.

NOTES

1. If we write equations (37.6) for four interior grid points (that is, $m = n = 4$), we will get the four algebraic equations:

$$\begin{aligned}
-4u_{22} + 0 + \sin(\pi/3) + u_{23} + u_{32} &= 0 \\
-4u_{23} + u_{22} + \sin(2\pi/3) + 0 + u_{33} &= 0 \\
-4u_{32} + 0 + u_{22} + u_{33} + 0 &= 0 \\
-4u_{33} + u_{32} + u_{23} + 0 + 0 &= 0
\end{aligned} \qquad (37.7)$$

from which we can solve for u_{22}, u_{23}, u_{32}, and u_{33}. The solution of these equations can be found by *iterative methods*, and Liebmann's method is one of them.

2. If we made our discretization sizes h and k smaller (so that we had more grid points), the analysis would be similar except that the system of algebraic equations (37.7) would be larger. In general, the number of *equations* will be equal to the number of *interior grid points*.
3. The system of equations (37.7) can be written in matrix form

$$\begin{bmatrix} -4 & 1 & 1 & 0 \\ 1 & -4 & 0 & 1 \\ 1 & 0 & -4 & 1 \\ 0 & 1 & 1 & -4 \end{bmatrix} \begin{bmatrix} u_{22} \\ u_{23} \\ u_{32} \\ u_{33} \end{bmatrix} = \begin{bmatrix} -\sin(\pi/3) \\ -\sin(2\pi/3) \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -0.86 \\ -0.86 \\ 0 \\ 0 \end{bmatrix}$$

In general, when we have several equations (maybe 1,000) this coefficient matrix takes on a specific form with many zeros. The solution of these sparse systems of equations can be found by special numerical methods. Iterative procedures, such as Jacobi's method, Gauss Seidel, and successive over-relaxation (SOR) are commonly used (along with techniques for speeding up convergence).

4. To solve the Neumann problem where there are *derivatives* on the boundary, we must also replace these derivatives by some finite difference approximation.

5. We can also solve equations like:
   (a) $u_{xx} + u_{yy} = f(x,y)$ (Nonhomogeneous equations)
   (b) $xu_{xx} + u_{yy} + 2u = \sin(x - y)$ (Variable coefficients; nonhomogeneous)
   (c) $\sin xu_{xx} + u_{xy} + 3u = 0$ (Variable coefficients)

   by the finite-difference method.
6. If the domain of the problem is an *irregularly* shaped region, we can overlay the region with grid lines and then approximate the solution at nearby grid points by interpolating the boundary conditions. After doing this, we can proceed in the usual manner. See Figure 37.2.

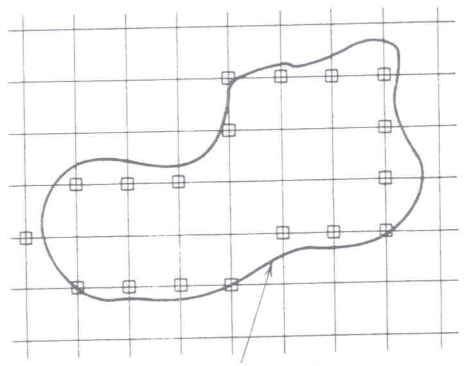

FIGURE 37.2

7. Several journals list computer programs for solving PDEs; some of them are:
   (a) *ACM Transactions on Mathematical Software*
   (b) *Computer Journal*
   (c) *Numerische Mathematik*
   (d) *BIT*

In addition, an extensive package of programs, called ELLPACK, has recently been designed for the purpose of solving fairly general elliptic boundary-value problems. This package will solve a wide variety of problems in two or three dimensions, various coordinate systems, arbitrary boundaries, general BCs, by an assortment of different methods.*

* Anyone interested in obtaining information about this program should contact Dr. John Rice, ELLPACK User's Guide CSD-TR 226, Computer Center, Purdue University, West Lafayette, Indiana 47907.

PROBLEMS

1. Derive approximation equation (37.4) for the second derivative $f''(x)$

$$f''(x) = \frac{1}{h^2}[f(x + h) - 2f(x) + f(x - h)]$$

2. Carry out the computation for two iterations in Dirichlet problem (37.5) using the Liebmann iterative process. Is the method converging?
3. What algebraic equations must be solved when you use finite-difference approximations to solve the following Poisson equation inside the square:

$$\begin{array}{lll} \text{PDE} & u_{xx} + u_{yy} = f(x,y) & 0 < x < 1 \quad 0 < y < 1 \\ \text{BC} & u(x,y) = g(x,y) & \text{On the boundary} \end{array}$$

4. What algebraic equations must you solve when replacing the derivatives in

$$\begin{array}{lll} \text{PDE} & u_{xx} + u_{yy} + 2u = 0 & 0 < x < 1 \quad 0 < y < 1 \\ \text{BC} & u(x,y) = g(x,y) & \text{On the boundary} \end{array}$$

   by their finite differences?
5. How would you solve the Neumann problem inside the square

$$\text{PDE} \quad u_{xx} + u_{yy} = 0 \quad 0 < x < 1 \quad 0 < y < 1$$

$$\text{BC} \quad \begin{cases} u = 0 & \text{On the top, bottom, and left-hand side of the square} \\ \dfrac{\partial u}{\partial x}(1,y) = 1 & 0 \le y \le 1 \end{cases}$$

   by the finite-difference method?
6. Write a *flow diagram* to solve the Dirichlet problem inside the square

$$\begin{array}{lll} \text{PDE} & u_{xx} + u_{yy} = f(x,y) & 0 < x < 1 \quad 0 < y < 1 \\ \text{BC} & u(x,y) = g(x,y) & \text{On the boundary} \end{array}$$

   with an arbitrary number of grid lines. If you know a computer language, write a program to carry out these computations.

OTHER READING

1. *Numerical Methods for Partial Differential Equations* by W. F. Ames. Academic Press, 1977. An up-to-date authoritative text on numerical techniques.

2. *Numerical Analysis* by S. S. Kunz. McGraw-Hill, 1964. Chapter 13 offers a clear, precise summary of some numerical methods in PDE theory.

3. *Numerical Solution of Partial Differential Equations* by G. D. Smith. Oxford University Press, 1965. A concise book describing finite difference methods in PDE theory; clearly written.

LESSON 38

An Explicit Finite-Difference Method

PURPOSE OF LESSON: To introduce the idea of explicit finite-difference methods and show how they can be used to solve hyperbolic and parabolic problems. The basic idea is that after a PDE like

$$u_t = u_{xx}$$

is replaced by its finite-difference approximation, we can solve for the solution explicitly at *one value of time* in terms of the solution at *earlier values of time*. In this way, an initial-boundary-value problem (hyperbolic or parabolic) can be solved by consecutively finding the solution at larger and larger values of time.

A problem we face is that as we make the grid sizes *small* so that the finite differences accurately represent the derivatives, the number of computations *increases*, and so the roundoff error increases.

In the previous lesson, we solved elliptic boundary-value problems (steady-state problems) where the PDE was satisfied in a given region of space, and the solution (or its derivative) was specified on the boundary. In those types of problems, we found the approximate solution at the *interior grid points* by solving a system of algebraic equations. In other words, the solution at all the interior grid points was found *simultaneously*.

In this lesson, we will show how *time-dependent problems* can be solved by finite-difference approximations. The idea here is that if we are given the solution when time is *zero*, we can then find the solution for $t = \Delta t, 2\Delta t, 3\Delta t, \ldots$ by means of a *marching process*. Replacing both the *space* and *time* derivatives by their finite-difference approximations, we can then solve for the solution $u_{i,j}$ in the difference equation *explicitly* in terms of the solution at earlier values of time. This process is called an **explicit-type marching process**, since we find the solution at a *single* value of time in terms of the solution at earlier values of time.

To show how this method works, we consider a representative problem from heat flow.