10.3 Elliptic Equations

As examples of elliptic partial differential equations, we consider the Laplace, Poisson, and Helmholtz equations. Recall that the Laplacian of the function $u(x, y)$ is

$$\nabla^2 u = u_{xx} + u_{yy}. \tag{1}$$

With this notation, we can write the Laplace, Poisson, and Helmholtz equations in the following forms:

$$\nabla^2 u = 0 \qquad \text{Laplace's equation,} \tag{2}$$

$$\nabla^2 u = g(x, y) \qquad \text{Poisson's equation,} \tag{3}$$

$$\nabla^2 u + f(x, y)u = g(x, y) \qquad \text{Helmholtz's equation.} \tag{4}$$

It is often the case that the boundary values for the function u are known at all points on the sides of a rectangular region R in the plane. In this case, each of these equations can be solved by the numerical technique known as the finite-difference method.

Laplacian Difference Equation

The Laplacian operator must be expressed in a discrete form suitable for numerical computations. The formula for approximating $f''(x)$ is obtained from

$$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} + \boldsymbol{O}(h^2). \tag{5}$$

When this is applied to the function $u(x, y)$ to approximate $u_{xx}(x, y)$ and $u_{yy}(x, y)$ and the results are added, we obtain

$$\nabla^2 u = \frac{u(x+h, y) + u(x-h, y) + u(x, y+h) + u(x, y-h) - 4u(x, y)}{h^2} + \boldsymbol{O}(h^2). \tag{6}$$

Assume that the rectangle $R = \{(x, y) : 0 \le x \le a, 0 \le y \le b,$ where $b/a = m/n\}$ is subdivided into $n - 1 \times m - 1$ squares with side h (i.e., $a = nh$ and $b = mh$), as shown in Figure 10.14.

To solve Laplace's equation, we impose the approximation

$$\frac{u(x+h, y) + u(x-h, y) + u(x, y+h) + u(x, y-h) - 4u(x, y)}{h^2} = 0, \tag{7}$$

which has order of accuracy $\boldsymbol{O}(h^2)$ at all interior grid points $(x, y) = (x_i, y_j)$ for $i = 2, \ldots, n-1$ and $j = 2, \ldots, m-1$. The grid points are uniformly spaced: $x_{i+1} = x_i + h$, $x_{i-1} = x_i - h$, $y_{i+1} = y_i + h$, and $y_{i-1} = y_i - h$. Using the approximation $u_{i,j}$ for $u(x_i, y_j)$, equation (7) can be written in the form

$$\nabla^2 u_{i,j} \approx \frac{u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j}}{h^2} = 0, \tag{8}$$

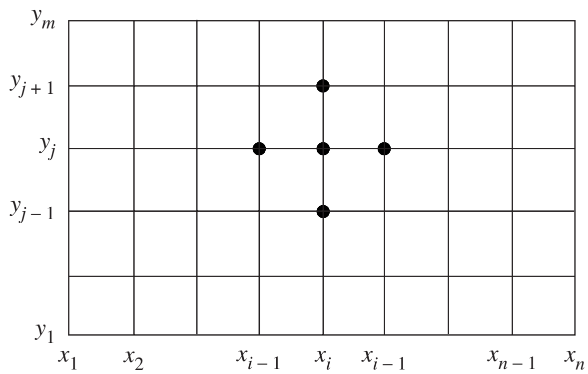

Figure 10.14 The grid used with Laplace's difference equation.

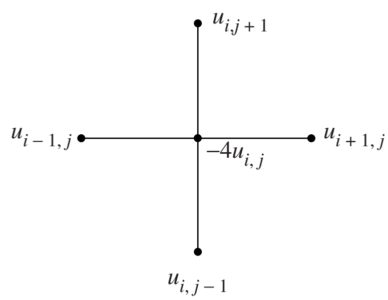

Figure 10.15 The Laplace stencil.

which is known as the ***five-point difference formula*** for Laplace's equation. This formula relates the function value $u_{i,j}$ to its four neighboring values $u_{i+1,j}$, $u_{i-1,j}$, $u_{i,j+1}$, and $u_{i,j-1}$, as shown in Figure 10.15. The term h^2 can be eliminated in (8) to obtain the Laplacian computational formula

$$u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j} = 0. \tag{9}$$

Setting Up the Linear System

Assume that the values $u(x, y)$ are known at the following boundary grid points:

$$\begin{aligned}
u(x_1, y_j) &= u_{1,j} && \text{for } 2 \le j \le m-1 && \text{(on the left)},\\
u(x_i, y_1) &= u_{i,1} && \text{for } 2 \le i \le n-1 && \text{(on the bottom)},\\
u(x_n, y_j) &= u_{n,j} && \text{for } 2 \le j \le m-1 && \text{(on the right)},\\
u(x_i, y_m) &= u_{i,m} && \text{for } 2 \le i \le n-1 && \text{(on the top)}.
\end{aligned}$$

Then applying the Laplacian computational formula (9) at each of the interior points of R will create a linear system of $(n-2)$ equations in $(n-2)$ unknowns, which is

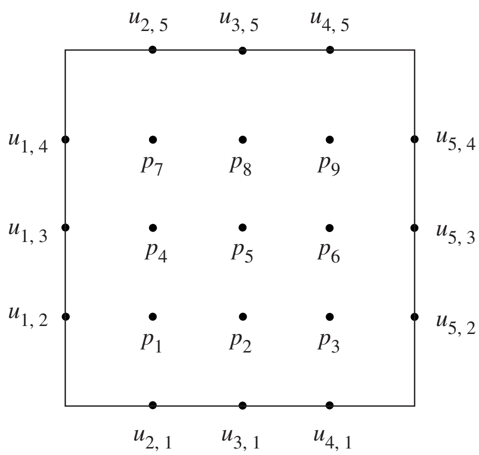

Figure 10.16 A 5 × 5 grid for boundary values only.

solved to obtain approximations to $u(x, y)$ at the interior points of R. For example, suppose that the region is a square, that $n = m = 5$, and that the unknown values of $u(x_i, y_j)$ at the nine interior grid points are labeled $p_1, p_2, \ldots, p_9$ and positioned in the grid as shown in Figure 10.16.

The Laplacian computational formula (9) is applied at each of the interior grid points, and the result is the system $\boldsymbol{AP} = \boldsymbol{B}$ of nine linear equations:

$$\begin{aligned}
-4p_1 + p_2 + p_4 &= -u_{2,1} - u_{1,2}\\
p_1 - 4p_2 + p_3 + p_5 &= -u_{3,1}\\
p_2 - 4p_3 + p_6 &= -u_{4,1} - u_{5,2}\\
p_1 - 4p_4 + p_5 + p_7 &= -u_{1,3}\\
p_2 + p_4 - 4p_5 + p_6 + p_8 &= 0\\
p_3 + p_5 - 4p_6 + p_9 &= -u_{5,3}\\
p_4 - 4p_7 + p_8 &= -u_{2,5} - u_{1,4}\\
p_5 + p_7 - 4p_8 + p_9 &= -u_{3,5}\\
p_6 + p_8 - 4p_9 &= -u_{4,5} - u_{5,4}.
\end{aligned}$$

Example 10.5. Find an approximate solution to Laplace's equation $\nabla^2 u = 0$ in the rectangle $R = \{(x, y) : 0 \le x \le 4, 0 \le y \le 4\}$, where $u(x, y)$ denotes the temperature at the point (x, y) and the boundary values are

$$u(x, 0) = 20 \quad \text{and} \quad u(x, 4) = 180 \quad \text{for} \quad 0 < x < 4,$$

and

$$u(0, y) = 80 \quad \text{and} \quad u(4, y) = 0 \quad \text{for} \quad 0 < y < 4.$$

See Figure 10.17 for the grid to be used.

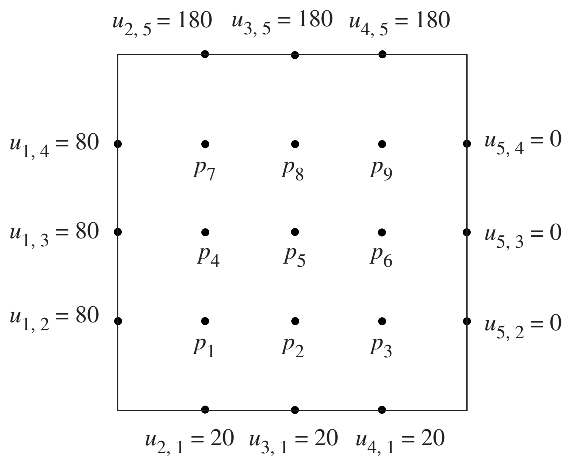

Figure 10.17 The 5×5 grid in Example 10.5.

Applying formula (9) in this case, the linear system $\boldsymbol{AP} = \boldsymbol{B}$ is

$$\begin{aligned}
-4p_1 + p_2 + p_4 &= -100 \\
p_1 - 4p_2 + p_3 + p_5 &= -20 \\
p_2 - 4p_3 + p_6 &= -20 \\
p_1 - 4p_4 + p_5 + p_7 &= -80 \\
p_2 + p_4 - 4p_5 + p_6 + p_8 &= 0 \\
p_3 + p_5 - 4p_6 + p_9 &= 0 \\
p_4 - 4p_7 + p_8 &= -260 \\
p_5 + p_7 - 4p_8 + p_9 &= -180 \\
p_6 + p_8 - 4p_9 &= -180
\end{aligned}$$

The solution vector $\boldsymbol{P}$ can be obtained by Gaussian elimination (or more efficient schemes can be devised, such as the extension of the tridiagonal algorithm to pentadiagonal systems). The temperatures at the interior grid points are expressed in vector form

$$\begin{aligned}
\boldsymbol{P} &= \begin{bmatrix} p_1 & p_2 & p_3 & p_4 & p_5 & p_6 & p_7 & p_8 & p_9 \end{bmatrix}' \\
&= [55.7143 \;\; 43.2143 \;\; 27.1429 \;\; 79.6429 \;\; 70.0000 \\
&\quad\; 45.3571 \;\; 112.857 \;\; 111.786 \;\; 84.2857]'.
\end{aligned}$$ ■

Derivative Boundary Conditions

The Neumann boundary conditions specify the directional derivative of $u(x, y)$ normal to an edge. For our illustration we will use the zero normal derivative condition,

$$\frac{\partial}{\partial N} u(x, y) = 0. \tag{10}$$

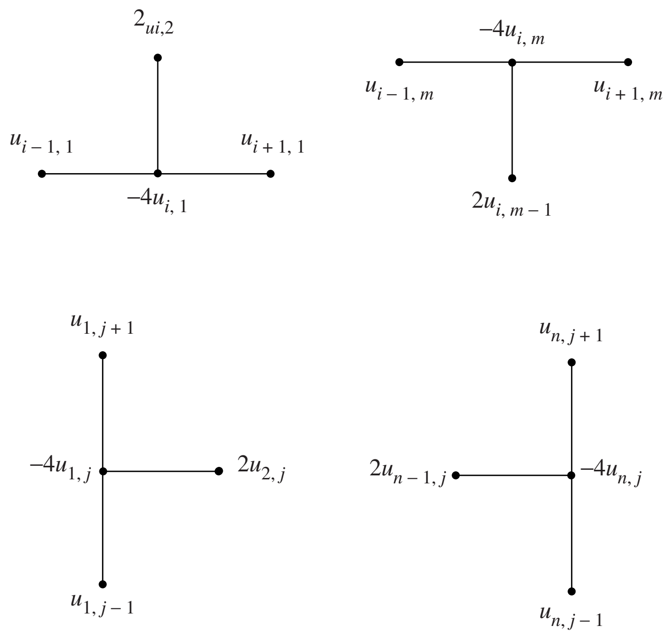

Figure 10.18 The Neumann stencils.

For applications in the area of heat flow, this means that the edge is thermally insulated and the heat flux throughout the edge is zero.

Suppose that $x = x_n$ is held fixed and that we are considering the right edge $x = a$ of the rectangle $R = \{(x, y) : 0 \le x \le a, 0 \le y \le b\}$. The normal boundary condition to be used along this edge is

$$\frac{\partial}{\partial x}u(x_n, y_j) = u_x(x_n, y_j) = 0. \tag{11}$$

Then the Laplace difference equation for the point (x_n, y_j) is

$$u_{n+1,j} + u_{n-1,j} + u_{n,j+1} + u_{n,j-1} - 4u_{n,j} = 0. \tag{12}$$

The value $u_{n+1,j}$ is unknown, because it lies outside the region R. However, we can use the numerical differentiation formula

$$\frac{u_{n+1,j} - u_{n-1,j}}{2h} \approx u_x(x_n, y_j) = 0 \tag{13}$$

and obtain the approximation $u_{n+1,j} \approx u_{n-1,j}$, which has order of accuracy $\boldsymbol{O}(h^2)$. When this approximation is used in (12), the result is

$$2u_{n-1,j} + u_{n,j+1} + u_{n,j-1} - 4u_{n,j} = 0.$$

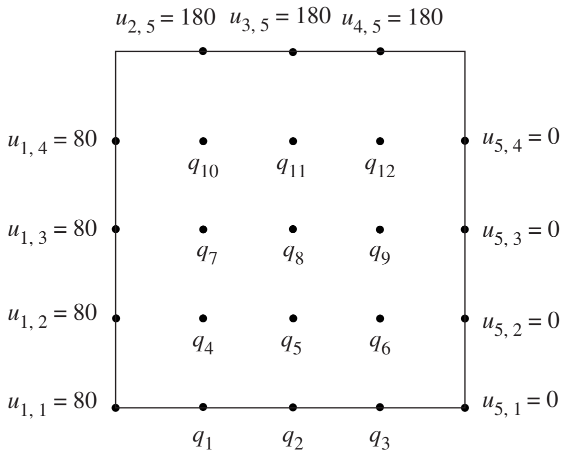

Figure 10.19 The 5×5 grid in Example 10.6.

This formula relates the function value $u_{n,j}$ to its three neighboring values $u_{n-1,j}$, $u_{n,j+1}$, and $u_{n,j-1}$.

The computational stencils for the other edges can be derived similarly (see Figure 10.18). The four cases for the Neumann computational stencils are summarized next:

(14) $$2u_{i,2} + u_{i-1,1} + u_{i+1,1} - 4u_{i,1} = 0 \qquad \text{(bottom edge)},$$

(15) $$2u_{i,m-1} + u_{i-1,m} + u_{i+1,m} - 4u_{i,m} = 0 \qquad \text{(top edge)},$$

(16) $$2u_{2,j} + u_{1,j-1} + u_{1,j+1} - 4u_{1,j} = 0 \qquad \text{(left edge)},$$

(17) $$2u_{n-1,j} + u_{n,j-1} + u_{n,j+1} - 4u_{n,j} = 0 \qquad \text{(right edge)}.$$

Suppose that the derivative condition $\partial u(x, y)/\partial N = 0$ is used along part of the boundary of R, and that known boundary values of $u(x, y)$ are used on the other portions of the boundary; then we have a mixed problem. The equations for determining approximations for $u(x_i, y_j)$ at boundary points will involve appropriate Neumann computational stencils (14) to (17). The Laplacian computational formula (9) is still used to determine approximations for $u(x_i, y_j)$ at the interior points of R.

Example 10.6. Find an approximate solution to Laplace's equation $\nabla^2 u = 0$ in the rectangle $R = \{(x, y) : 0 \le x \le 4, 0 \le y \le 4\}$, where $u(x, y)$ denotes the temperature at the point (x, y) and the boundary values are shown in Figure 10.19:

$$\begin{aligned}
u(x, 4) &= 180 && \text{for } 0 < x < 4,\\
u_y(x, 0) &= 0 && \text{for } 0 < x < 4,\\
u(0, y) &= 80 && \text{for } 0 \le y < 4,\\
u(4, y) &= 0 && \text{for } 0 \le y < 4.
\end{aligned}$$

The Neumann computational formula (14) is applied at the boundary points q_1, q_2, and q_3, and the Laplace computational stencil (9) is applied at the other points q_4, q_5, ..., q_{12}. The result is a linear system $\boldsymbol{A}\boldsymbol{Q} = \boldsymbol{B}$ involving 12 equations in 12 unknowns:

$$
\begin{aligned}
-4q_1 + q_2 + 2q_4 &= -80\\
q_1 - 4q_2 + q_3 + 2q_5 &= 0\\
q_2 - 4q_3 + 2q_6 &= 0\\
q_1 - 4q_4 + q_5 + q_7 &= -80\\
q_2 + q_4 - 4q_5 + q_6 + q_8 &= 0\\
q_3 + q_5 - 4q_6 + q_9 &= 0\\
q_4 - 4q_7 + q_8 + q_{10} &= -80\\
q_5 + q_7 - 4q_8 + q_9 + q_{11} &= 0\\
q_6 + q_8 - 4q_9 + q_{12} &= 0\\
q_7 - 4q_{10} + q_{11} &= -260\\
q_8 + q_{10} - 4q_{11} + q_{12} &= -180\\
q_9 + q_{11} - 4q_{12} &= -180
\end{aligned}
$$

The solution vector $\boldsymbol{Q}$ can be obtained by Gaussian elimination (or more efficient schemes can be devised, such as the extension of the tridiagonal algorithm to pentadiagonal systems). The temperatures at the interior grid points and along the lower edge are expressed in vector form as

$$
\begin{aligned}
\boldsymbol{Q} &= \begin{bmatrix} q_1 & q_2 & q_3 & q_4 & q_5 & q_6 & q_7 & q_8 & q_9 & q_{10} & q_{11} & q_{12} \end{bmatrix}'\\
&= [71.8218 \;\; 56.8543 \;\; 32.2342 \;\; 75.2165 \;\; 61.6806 \;\; 36.0412\\
&\qquad 87.3636 \;\; 78.6103 \;\; 50.2502 \;\; 115.628 \;\; 115.147 \;\; 86.3492]'.
\end{aligned}
$$

■

Iterative Methods

The preceding method showed how to solve Laplace's difference equation by constructing a certain system of linear equations and solving it. The shortcoming of this method is storage; each interior grid point introduces an equation to be solved. Since better approximations require a finer mesh grid, many equations might be needed. For example, the solution of Laplace's equation with the Dirichlet boundary conditions requires solving a system of $(n-2)(m-2)$ equations. If R is divided into a modest number of squares, say 10 by 10, there would be 91 equations involving 91 unknowns. Hence it is sensible to develop techniques that will reduce the amount of storage. An iterative method would require only the storage of the 100 numerical approximations $\{u_{i,j}\}$ throughout the grid.

Let us start with Laplace's difference equation

$$u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j} = 0 \tag{18}$$

and suppose that the boundary values $u(x, y)$ are known at the following grid points:

$$
\begin{aligned}
&u(x_1, y_j) = u_{1,j} &&\text{for } 2 \le j \le m-1 &&\text{(on the left)},\\
&u(x_i, y_1) = u_{i,1} &&\text{for } 2 \le i \le n-1 &&\text{(on the bottom)},\\
&u(x_n, y_j) = u_{n,j} &&\text{for } 2 \le j \le m-1 &&\text{(on the right)},\\
&u(x_i, y_m) = u_{i,m} &&\text{for } 2 \le i \le n-1 &&\text{(on the top)}.
\end{aligned} \tag{19}
$$

Equation (18) is rewritten in the following form that is suitable for iteration:

$$u_{i,j} = u_{i,j} + r_{i,j}, \tag{20}$$

where

$$r_{i,j} = \frac{u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j}}{4} \tag{21}$$

for $2 \le i \le n-1$ and $2 \le j \le m-1$.

Starting values for all interior grid points must be supplied. The constant K, which is the average of the $2n + 2m - 4$ boundary values given in (19), can be used for this purpose. One iteration consists of sweeping formula (20) throughout all of the interior points of the grid. Successive iterations sweep the interior of the grid with the Laplace iterative operator (20) until the residual term $r_{i,j}$ on the right side of equation (20) is "reduced to zero" (i.e., $|r_{i,j}| < \epsilon$ holds for each $2 \le i \le n-1$ and $2 \le j \le m-1$). The speed of convergence for reducing all the residuals $\{r_{i,j}\}$ to zero is increased by using the method called *successive overrelaxation* (SOR). The SOR method uses the iteration formula

$$
\begin{aligned}
u_{i,j} &= u_{i,j} + \omega\left(\frac{u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j}}{4}\right)\\
&= u_{i,j} + \omega r_{i.j},
\end{aligned} \tag{22}
$$

where the parameter ω lies in the range $1 \le \omega < 2$. In the SOR method, formula (22) is swept across the grid until $|r_{i,j}| < \epsilon$. The optimal choice for ω is based on the study of eigenvalues of iteration matrices for linear systems and is given in this case by the formula

$$\omega = \frac{4}{2 + \sqrt{4 - \left(\cos\left(\dfrac{\pi}{n-1}\right) + \cos\left(\dfrac{\pi}{m-1}\right)\right)^2}}. \tag{23}$$

If the Neumann boundary condition is specified on some portion of the boundary, we must rewrite equations (14) through (17) in a form that is suitable for iteration. The

Table 10.6 Approximate Solution to Laplace's Equation with Dirichlet Conditions

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
y_9	130.000	180.000	180.000	180.000	180.000	180.000	180.000	180.000	90.0000
y_8	80.000	124.821	141.172	145.414	144.005	137.478	122.642	88.6070	0.0000
y_7	80.000	102.112	113.453	116.479	113.126	103.266	84.4844	51.7856	0.0000
y_6	80.000	89.1736	94.0499	93.9210	88.7553	77.9737	60.2439	34.0510	0.0000
y_5	80.000	80.5319	79.6515	76.3999	70.0003	59.6301	44.4667	24.1744	0.0000
y_4	80.000	73.3023	67.6241	62.0267	55.2159	46.0796	33.8184	18.1798	0.0000
y_3	80.000	65.0528	55.5159	48.8671	42.7568	35.6543	26.5473	14.7266	0.0000
y_2	80.000	51.3931	40.5195	35.1691	31.2899	27.2335	21.9900	14.1791	0.0000
y_1	50.000	20.0000	20.0000	20.0000	20.0000	20.0000	20.0000	20.0000	10.0000

four cases are summarized next and include the relaxation parameter ω:

(24) $$u_{i,1} = u_{i,1} + \omega\left(\frac{2u_{i,2} + u_{i-1,1} + u_{i+1,1} - 4u_{i,1}}{4}\right) \qquad \text{(bottom edge)},$$

(25) $$u_{i,m} = u_{i,m} + \omega\left(\frac{2u_{i,m-1} + u_{i-1,m} + u_{i+1,m} - 4u_{i,m}}{4}\right) \qquad \text{(top edge)},$$

(26) $$u_{i,j} = u_{i,j} + \omega\left(\frac{2u_{2,j} + u_{1,j-1} + u_{1,j+1} - 4u_{1,j}}{4}\right) \qquad \text{(left edge)},$$

(27) $$u_{n,j} = u_{n,j} + \omega\left(\frac{2u_{n-1,j} + u_{n,j-1} + u_{n,j+1} - 4u_{n,j}}{4}\right) \qquad \text{(right edge)}.$$

Example 10.7. Use an iterative method to compute an approximate solution to Laplace's equation $\nabla^2 = 0$ in $R = \{(x, y) : 0 \le x \le 4, 0 \le y \le 4\}$, where the boundary values are

$$u(x, 0) = 20 \quad \text{and} \quad u(x, 4) = 180 \quad \text{for } 0 < x < 4,$$

and

$$u(0, y) = 80 \quad \text{and} \quad u(4, y) = 0 \quad \text{for } 0 < y < 4.$$

For illustration, the square is divided into 64 squares with sides $\Delta x = h = 0.5$ and $\Delta y = h = 0.5$. The initial value at the interior grid points was set at $u_{i,j} = 70$ for each $i = 2, \ldots, 8$ and $j = 2, \ldots, 8$. The SOR method was used with the parameter $\omega = 1.44646$ (substitute $n = 9$ and $m = 9$ in formula (23)). After 19 iterations, the residual was uniformly reduced (i.e.,$|r_{i,j}| \le 0.000606 < 0.001$). The resulting approximations are given in Table 10.6. Because of the discontinuity of the boundary function at the corners, the boundary values $u_{1,1} = 50$, $u_{9,1} = 10$, $u_{1,9} = 130$, and $u_{9,9} = 90$ have been introduced in Table 10.6 and Figure 10.20; they were not used in the computations at the interior grid points. A three-dimensional presentation of the data in Table 10.6 is given in Figure 10.20.

■

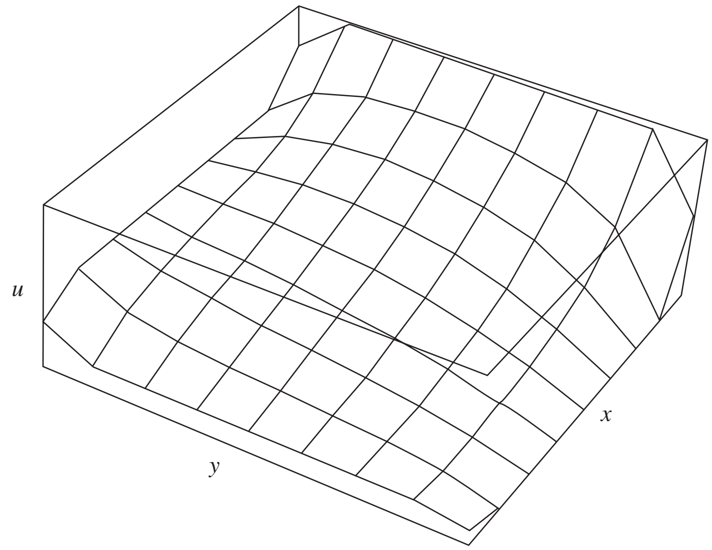

Figure 10.20 $u = u(x, y)$ with Dirichlet boundary values.

Example 10.8. Use an iterative method to compute an approximate solution to Laplace's equation $\nabla^2 u = 0$ in $R = \{(x, y) : 0 \le x \le 4, 0 \le y \le 4\}$, where the boundary values are

$$
\begin{array}{llll}
u(x,4) = 180 & \text{for} \quad y = 4 & \text{and} & 0 < x < 4, \\
u_y(x,0) = 0 & \text{for} \quad y = 0 & \text{and} & 0 < x < 4, \\
u(0,y) = 80 & \text{for} \quad x = 0 & \text{and} & 0 \le y < 4, \\
u(4,y) = 0 & \text{for} \quad x = 4 & \text{and} & 0 \le y < 4.
\end{array}
$$

For illustration, the square is divided into 64 squares with sides $\Delta x = h = 0.5$ and $\Delta y = h = 0.5$. Starting values using linear interpolation were used along the edge where $y = y_1 = 0$. The initial value at the interior grid points was set at $u_{i,j} = 70$ for each $i = 2, \ldots, 8$ and $j = 2, \ldots, 8$. Then the SOR method was employed with the parameter $\omega = 1.44646$ (as in Example 10.7). After 29 iterations, the residual was reduced uniformly; (i.e., $|r_{i,j}| \le 0.000998 < 0.001$). The resulting approximations are given in Table 10.7. Because of the discontinuity of the boundary functions at the corners, the boundary values $u_{1,9} = 130$ and $u_{9,9} = 90$ have been introduced in Table 10.7 and Figure 10.21; they were not used in the computations at the interior grid points. A three-dimensional presentation of the data in Table 10.7 is given in Figure 10.21. ■

Poisson's and Helmholtz's Equations

Consider Poisson's equation

$$\nabla^2 u = g(x, y). \tag{28}$$

Table 10.7 Approximate Solution to Laplace's Equation with Mixed Boundary Conditions

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
y_9	130.000	180.000	180.000	180.000	180.000	180.000	180.000	180.000	90.0000
y_8	80.000	126.457	142.311	146.837	145.468	138.762	123.583	89.1008	0.0000
y_7	80.000	103.518	115.951	119.568	116.270	105.999	86.4683	52.8201	0.0000
y_6	80.000	91.6621	98.4053	99.2137	94.0461	82.4936	63.4715	35.7113	0.0000
y_5	80.000	84.7247	86.7936	84.8347	78.2063	66.4578	49.2124	26.5538	0.0000
y_4	80.000	80.4424	79.2089	75.1245	67.4860	55.9185	40.3665	21.2915	0.0000
y_3	80.000	77.8354	74.4742	68.9677	60.6944	49.3635	35.0435	18.2459	0.0000
y_2	80.000	76.4244	71.8842	65.5772	56.9600	45.7972	32.1981	16.6485	0.0000
y_1	80.000	75.9774	71.0605	64.4964	55.7707	44.6670	31.3032	16.1500	0.0000

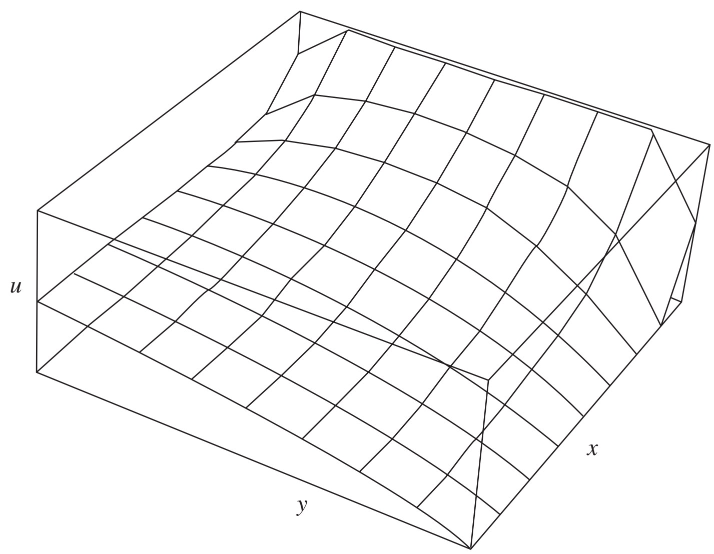

Figure 10.21 $u = u(x, y)$ for a mixed problem.

Using the notation $g_{i,j} = g(x_i, y_j)$, the generalization of formula (20) for solving (28) over the rectangular grid is

$$u_{i,j} = u_{i,j} + \frac{u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j} - h^2 g_{i,j}}{4}. \tag{29}$$

Consider Helmholtz's equation

$$\nabla^2 u + f(x, y)u = g(x, y). \tag{30}$$

Using the notation $f_{i,j} = f(x_i, y_j)$, the generalization of formula (20) for solving (30) over the rectangular grid is

$$(31)\quad u_{i,j} = u_{i,j} + \frac{u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - (4 - h^2 f_{i,j})u_{i,j} - h^2 g_{i,j}}{4 - h^2 f_{i,j}}.$$

These formulas are explored in greater detail in the exercises.

Improvements

A modification of (8) that can be employed is the ***nine-point difference formula*** for Laplace's equation:

$$\begin{aligned}\nabla^2 u_{i,j} \approx \frac{1}{6h^2}&(u_{i+1,j-1} + u_{i-1,j-1} + u_{i+1,j+1} + u_{i-1,j+1} \\ &+ 4u_{i+1,j} + 4u_{i-1,j} + 4u_{i,j+1} + 4u_{i,j-1} - 20u_{i,j}) = 0.\end{aligned}$$

The truncation errors for the nine- and five-point formulas (see formula (8)) are $\boldsymbol{O}(h^4)$ and $\boldsymbol{O}(h^2)$, respectively. Thus there is an advantage to using the nine-point difference formula.

Numerical Methods Using Matlab, 4^{th} Edition, 2004

John H. Mathews and Kurtis K. Fink

ISBN: 0-13-065248-2

Prentice-Hall Inc.
Upper Saddle River, New Jersey, USA
http://vig.prenhall.com/