CHAPTER FOUR

Solution Methods of Finite Difference Equations

In this chapter, solution methods for elliptic, parabolic, hyperbolic equations, and Burgers' equations are presented. These equations do not represent actual fluid dynamics problems, but the methods discussed in this chapter will form the basis for solving incompressible and compressible flow problems which are presented in Chapters 5 and 6, respectively. Although the computational schemes for these equations have been in existence for many years and are well documented in other text books, they are summarized here merely for the sake of completeness and for references in later chapters.

4.1 ELLIPTIC EQUATIONS

Elliptic equations represent one of the fundamental building blocks in fluid mechanics. Steady heat conduction, diffusion processes in viscous, turbulent, and boundary layer flows, as well as chemically reacting flows are characterized by the elliptic nature of the governing equations. Various difference schemes for the elliptic equations and some solution methods are also presented in this chapter.

4.1.1 FINITE DIFFERENCE FORMULATIONS

Consider the Laplace equation which is one of the typical elliptic equations,

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \tag{4.1.1}$$

The five-point and nine-point finite differences for the Laplace equation are, respectively,

$$\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\Delta x^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{\Delta y^2} = 0 \tag{4.1.2}$$

$$\frac{-u_{i-2,j} + 16u_{i-1,j} - 30u_{i,j} + 16u_{i+1,j} - u_{i+2,j}}{12\Delta x^2} + \frac{-u_{i,j-2} + 16u_{i,j-1} - 30u_{i,j} + 16u_{i,j+1} - u_{i,j+2}}{12\Delta y^2} = 0 \tag{4.1.3}$$

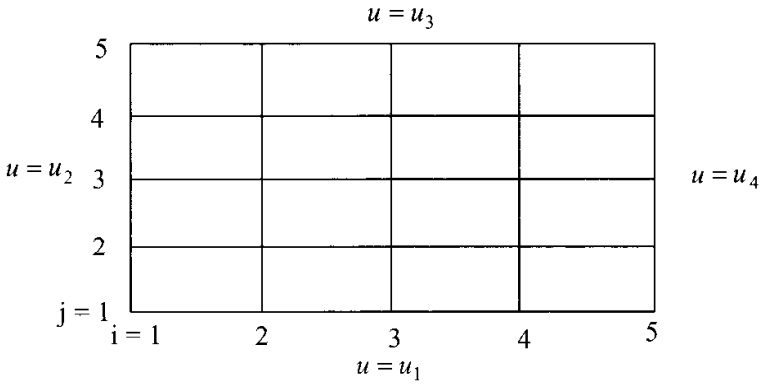

Figure 4.1.1 Finite difference grids with Dirichlet boundary conditions specified at all boundary nodes.

as discussed in Chapter 3. For illustration, let us consider the five-point scheme (4.1.2) for the geometry given in Figure 4.1.1.

$$u_{i+1,j} + u_{i-1,j} + \beta^2 u_{i,j+1} + \beta^2 u_{i,j-1} - 2(1+\beta^2)u_{i,j} = 0 \tag{4.1.4}$$

where β is defined as $\beta = \Delta x/\Delta y$. For Dirichlet boundary conditions, the values of u at all boundary nodes are given. Thus, writing (4.1.4) at all interior nodes and setting

$$\gamma = -2(1+\beta^2)$$

we obtain for the discretization as shown in Figure 4.1.1,

$$\begin{bmatrix} \gamma & 1 & 0 & \beta^2 & 0 & 0 & 0 & 0 & 0 \\ 1 & \gamma & 1 & 0 & \beta^2 & 0 & 0 & 0 & 0 \\ 0 & 1 & \gamma & 0 & 0 & \beta^2 & 0 & 0 & 0 \\ \beta^2 & 0 & 0 & \gamma & 1 & 0 & \beta^2 & 0 & 0 \\ 0 & \beta^2 & 0 & 1 & \gamma & 1 & 0 & \beta^2 & 0 \\ 0 & 0 & \beta^2 & 0 & 1 & \gamma & 0 & 0 & \beta^2 \\ 0 & 0 & 0 & \beta^2 & 0 & 0 & \gamma & 1 & 0 \\ 0 & 0 & 0 & 0 & \beta^2 & 0 & 1 & \gamma & 1 \\ 0 & 0 & 0 & 0 & 0 & \beta^2 & 0 & 1 & \gamma \end{bmatrix} \begin{bmatrix} u_{2,2} \\ u_{3,2} \\ u_{4,2} \\ u_{2,3} \\ u_{3,3} \\ u_{4,3} \\ u_{2,4} \\ u_{3,4} \\ u_{4,4} \end{bmatrix} = \begin{bmatrix} -u_{1,2} - \beta^2 u_{2,1} \\ -\beta^2 u_{3,1} \\ -u_{5,2} - \beta^2 u_{4,1} \\ -u_{1,3} \\ 0 \\ -u_{5,3} \\ -u_{1,4} - \beta^2 u_{2,5} \\ -\beta^2 u_{3,5} \\ -u_{5,4} - \beta^2 u_{4,5} \end{bmatrix} \tag{4.1.5}$$

Notice that the matrix on the left-hand side is always pentadiagonalized for the five-point scheme. The nine-point schemes given by (4.1.3), although more complicated, can be written similarly as in (4.1.5).

There are two types of solution methods for the linear algebraic equations of the form (4.1.5). The first kind includes the direct methods such as Gauss elimination, Thomas algorithm, Chelosky method, etc. The second kind includes the iterative methods such as Jacobi iteration, point Gauss-Seidel iteration, line Gauss-Seidel iteration, point-successive over-relaxation (PSOR), line successive over-relaxation (LSOR), alternating direction implicit (ADI), and so on.

The disadvantage of the direct methods is that they are more time consuming than iterative methods. Additionally, direct methods are susceptible to round-off errors which, in large systems of equations, can be catastrophic. In contrast, errors in each step of an iterative method are corrected in the subsequent step, thus round-off errors are usually not a concern. We elaborate on some of the iterative methods in Section 4.1.2, and a direct method of Gaussian elimination in Section 4.1.3. Other methods will be presented in later chapters, including conjugate gradient methods (CGM) (Section 10.3.1) and generalized minimal residual (GMRES) algorithm (Section 11.5.3).

4.1.2 ITERATIVE SOLUTION METHODS

Jacobi Iteration Method

In this method, the unknown u at each grid point is solved in terms of the initial guess values or previously computed values. Thus, from (4.1.4), we compute a new value of $u_{i,j}$ at the new iteration $k+1$ level as

$$u_{i,j}^{k+1} = \frac{1}{2(1+\beta^2)}\left[u_{i+1,j}^{k} + u_{i-1,j}^{k} + \beta^2\left(u_{i,j+1}^{k} + u_{i,j-1}^{k}\right)\right] \tag{4.1.6}$$

where k represents the previously computed values or the initial guesses for the first round of computations. The computation is carried out until a specified convergence criterion is achieved.

We may use the newly computed values of the dependent variables to compute the neighboring points when available. This process leads to efficient schemes such as the Gauss-Seidel method.

Point Gauss-Seidel Iteration Method

In this method, the current values of the dependent variables are used to compute neighboring points as soon as they are available. This will increase the convergence rate. The solution for the independent variables is obtained as

$$u_{i,j}^{k+1} = \frac{1}{2(1+\beta^2)}\left[u_{i+1,j}^{k} + u_{i-1,j}^{k+1} + \beta^2\left(u_{i,j+1}^{k} + u_{i,j-1}^{k+1}\right)\right] \tag{4.1.7}$$

The $k+1$ level on the right-hand side of (4.1.7) indicates that the solution process takes advantage of the values at $i-1$ and $j-1$ which have just been calculated in the previous step.

Line Gauss-Seidel Iteration Method

Equation (4.1.5) may be solved for the three unknowns at $(i-1, j)$, (i, j), $(i+1, j)$, as follows:

$$u_{i-1,j}^{k+1} - 2(1+\beta^2)u_{i,j}^{k+1} + u_{i+1,j}^{k+1} = -\beta^2\left(u_{i,j+1}^{k} + u_{i,j-1}^{k+1}\right) \tag{4.1.8}$$

which leads to a tridiagonal matrix. Note that $u_{i,j-1}^{k+1}$ is known at the $k+1$ level, whereas $u_{i,j+1}^{k}$ was determined at the kth level. This method converges faster than the point Gauss-Seidel method, but it takes more computer time per iteration. The line iteration technique is useful when the variable changes more rapidly in the direction of the iteration because of the use of the updated values.

Point Successive Over-Relaxation Method (PSOR)

Convergence of the point Gauss-Seidel method can be accelerated by rearranging (4.1.7),

$$u_{i,j}^{k+1} = u_{i,j}^{k} + \frac{1}{2(1+\beta^2)}\left[u_{i+1,j}^{k} + u_{i-1,j}^{k+1} + \beta^2\left(u_{i,j+1}^{k} + u_{i,j-1}^{k+1}\right) - 2(1+\beta^2)u_{i,j}^{k}\right] \tag{4.1.9}$$

The idea is to make $u_{i,j}^{k}$ approach $u_{i,j}^{k+1}$ faster. To this end, we introduce the relaxation parameter, ω, to be multiplied to the terms with brackets on the right-hand side of (4.1.9),

$$u_{i,j}^{k+1} = u_{i,j}^{k} + \frac{\omega}{2(1+\beta^2)}\left[u_{i+1,j}^{k} + u_{i-1,j}^{k+1} + \beta^2\left(u_{i,j+1}^{k} + u_{i,j-1}^{k+1}\right) - 2(1+\beta^2)u_{i,j}^{k}\right]$$

or

$$u_{i,j}^{k+1} = (1-\omega)u_{i,j}^{k} + \frac{\omega}{2(1+\beta^2)}\left[u_{i+1,j}^{k} + u_{i-1,j}^{k+1} + \beta^2\left(u_{i,j+1}^{k} + u_{i,j-1}^{k+1}\right)\right] \tag{4.1.10}$$

where we choose $1 < \omega < 2$ for convergence. This is known as the point successive over-relaxation procedure. For certain problems, however, a better convergence may be achieved by under-relaxation, where the relaxation parameter is chosen as $0 < \omega < 1$. Note that for $\omega = 1$ we recover the Gauss-Seidel iteration method.

For a rectangular domain subjected to Dirichlet boundary conditions with constant step size, we obtain the optimum relaxation parameter

$$\omega_{opt} = \frac{2 - \sqrt{1-a}}{a} \tag{4.1.11}$$

with

$$a = \left[\frac{\cos\left(\dfrac{\pi}{IM-1}\right) + \beta^2 \cos\left(\dfrac{\pi}{JM-1}\right)}{1+\beta^2}\right]^2 \tag{4.1.12}$$

where IM and JM refer to the maximum numbers of i and j, respectively. Further details are found in Wachspress [1966] and Hageman and Young [1981].

Line Successive Over-Relaxation Method (LSOR)

The idea of relaxation may also be applied to the line Gauss-Seidel method,

$$\omega u_{i-1,j}^{k+1} - 2(1+\beta^2)u_{i,j}^{k+1} + \omega u_{i+1,j}^{k+1} = -(1-\omega)\lfloor 2(1+\beta^2)\rfloor u_{i,j}^{k} - \omega\beta^2\left(u_{i,j+1}^{k} + u_{i,j-1}^{k+1}\right) \tag{4.1.13}$$

where an optimum relaxation parameter ω can be determined experimentally, or by (4.1.11).

Alternating Direction Implicit (ADI) Method

In this method, a tridiagonal system is solved for rows first and then followed by columns, or vice versa. Toward this end, we recast (4.1.8) into two parts:

$$u_{i-1,j}^{k+\frac{1}{2}} - 2(1+\beta^2)u_{i,j}^{k+\frac{1}{2}} + u_{i+1,j}^{k+\frac{1}{2}} = -\beta^2\left(u_{i,j+1}^{k} + u_{i,j-1}^{k+\frac{1}{2}}\right) \tag{4.1.14a}$$

and

$$\beta^2 u_{i,j-1}^{k+1} - 2(1+\beta^2)u_{i,j}^{k+1} + \beta^2 u_{i,j+1}^{k+1} = -\left(u_{i+1,j}^{k+\frac{1}{2}} + u_{i-1,j}^{k+1}\right) \tag{4.1.14b}$$

Here (4.1.14a) and (4.1.14b) are solved implicitly in the x-direction and y-direction, respectively. The relaxation parameter ω may be introduced to accelerate the convergence,

$$\omega u_{i-1,j}^{k+\frac{1}{2}} - 2(1+\beta^2)u_{i,j}^{k+\frac{1}{2}} + \omega u_{i+1,j}^{k+\frac{1}{2}} = -(1-\omega)[2(1+\beta^2)]u_{i,j}^{k} - \omega\beta^2\left(u_{i,j+1}^{k} + u_{i,j-1}^{k+\frac{1}{2}}\right) \tag{4.1.15a}$$

and

$$\omega\beta^2 u_{i,j-1}^{k+1} - 2(1+\beta^2)u_{i,j}^{k+1} + \omega\beta^2 u_{i,j+1}^{k+1} = -(1-\omega)[2(1+\beta^2)]u_{i,j}^{k+\frac{1}{2}} - \omega\left(u_{i+1,j}^{k+\frac{1}{2}} + u_{i-1,j}^{k+1}\right) \tag{4.1.15b}$$

with the optimum ω being determined experimentally as appropriate for different physical problems.

4.1.3 DIRECT METHOD WITH GAUSSIAN ELIMINATION

Consider the simultaneous equations resulting from the finite difference approximation of (4.1.2) in the form

$$\begin{aligned} &k_{11}u_1 + k_{12}u_2 + \cdots = g_1 \\ &k_{21}u_1 + k_{22}u_2 + \cdots = g_2 \\ &\vdots \\ &k_{n1}u_n \cdot \qquad\quad \cdots = g_n \end{aligned} \tag{4.1.16}$$

Here, our objective is to transform the system into an upper triangular array. To this end, we choose the first row as the "pivot" equation and eliminate the u_1 term from each equation below it. To eliminate u_1 from the second equation, we multiply the first equation by k_{21}/k_{11} and subtract it from the second equation. We continue similarly until u_1 is eliminated from all equations. We then eliminate $u_2, u_3, \ldots$ in the same manner until we achieve the upper triangular form,

$$\begin{bmatrix} k_{11} & k_{12} & \cdot & \cdot \\ & k'_{22} & \cdot & \cdot \\ & & \cdot & \cdot \\ & & & k'_{nn} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \cdot \\ u_n \end{bmatrix} = \begin{bmatrix} g_1 \\ g'_2 \\ \cdot \\ g'_n \end{bmatrix} \tag{4.1.17}$$

It is seen that backsubstitution will determine all unknowns.

An example for the solution of a typical elliptical equation is shown in Section 4.7.1.

4.2 PARABOLIC EQUATIONS

The governing equations for some problems in fluid dynamics, such as unsteady heat conduction or boundary layer flows, are parabolic. The finite difference representation