CHAPTER

2

DISCRETIZATION OF PDES

Partial Differential Equations (PDEs) constitute by far the biggest source of sparse matrix problems. The typical way to solve such equations is to *discretize* them, i.e., to approximate them by equations that involve a finite number of unknowns. The matrix problems that arise from these discretizations are generally large and sparse, i.e., they have very few nonzero entries. There are several different ways to discretize a Partial Differential Equation. The simplest method uses *finite difference* approximations for the partial differential operators. The *Finite Element Method* replaces the original function by a function which has some degree of smoothness over the global domain, but which is piecewise polynomial on simple cells, such as small triangles or rectangles. This method is probably the most general and well understood discretization technique available. In between these two methods, there are a few conservative schemes called *Finite Volume Methods*, which attempt to emulate continuous *conservation laws* of physics. This chapter introduces these three different discretization methods.

2.1 PARTIAL DIFFERENTIAL EQUATIONS

Physical phenomena are often modeled by equations that relate several partial derivatives of physical quantities, such as forces, momentums, velocities, energy, temperature, etc. These equations rarely have a *closed-form* (explicit) solution. In this chapter, a few types of Partial Differential Equations are introduced, which will serve as models throughout the book. Only one- or two-dimensional problems are considered, and the space variables are denoted by x in the case of one-dimensional problems or x_1 and x_2 for two-dimensional problems. In two dimensions, x denotes the "vector" of components (x_1, x_2).

2.1.1 ELLIPTIC OPERATORS

One of the most common Partial Differential Equations encountered in various areas of engineering is Poisson's equation:

$$\frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2} = f, \text{ for } x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \text{ in } \Omega \tag{2.1}$$

where Ω is a bounded, open domain in $\mathbb{R}^2$. Here, x_1, x_2 are the two space variables.

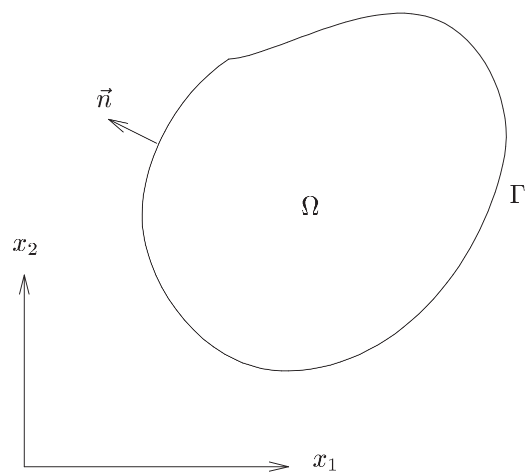

Figure 2.1 *Domain Ω for Poisson's equation.*

The above equation is to be satisfied only for points that are located at the interior of the domain Ω. Equally important are the conditions that must be satisfied on the *boundary* Γ of Ω. These are termed *boundary conditions*, and they come in three common types:

Dirichlet condition	$u(x)$	$=$	$\phi(x)$
Neumann condition	$\frac{\partial u}{\partial \vec{n}}(x)$	$=$	0
Cauchy condition	$\frac{\partial u}{\partial \vec{n}}(x) + \alpha(x)u(x)$	$=$	$\gamma(x)$

The vector $\vec{n}$ usually refers to a unit vector that is normal to Γ and directed outwards. Note that the Neumann boundary conditions are a particular case of the Cauchy conditions with $\gamma = \alpha = 0$. For a given unit vector, $\vec{v}$ with components v_1 and v_2, the directional derivative $\partial u / \partial \vec{v}$ is defined by

$$\begin{aligned} \frac{\partial u}{\partial \vec{v}}(x) &= \lim_{h \to 0} \frac{u(x + h\vec{v}) - u(x)}{h} \\ &= \frac{\partial u}{\partial x_1}(x) v_1 + \frac{\partial u}{\partial x_2}(x) v_2 &\quad (2.2) \\ &= \nabla u . \vec{v} &\quad (2.3) \end{aligned}$$

where ∇u is the gradient of u,

$$\nabla u = \begin{pmatrix} \frac{\partial u}{\partial x_1} \\ \frac{\partial u}{\partial x_2} \end{pmatrix}, \tag{2.4}$$

and the dot in (2.3) indicates a dot product of two vectors in $\mathbb{R}^2$.

In reality, Poisson's equation is often a limit case of a time-dependent problem. It can, for example, represent the steady-state temperature distribution in a region Ω when there is a heat source f that is constant with respect to time. The boundary conditions should then model heat loss across the boundary Γ.

The particular case where $f(x) = 0$, i.e., the equation

$$\Delta u = 0,$$

to which boundary conditions must be added, is called the *Laplace equation* and its solutions are called *harmonic functions*.

Many problems in physics have boundary conditions of *mixed type*, e.g., of Dirichlet type in one part of the boundary and of Cauchy type in another. Another observation is that the Neumann conditions do not define the solution uniquely. Indeed, if u is a solution, then so is $u + c$ for any constant c.

The operator

$$\Delta = \frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2}$$

is called the *Laplacean operator* and appears in many models of physical and mechanical phenomena. These models often lead to more general elliptic operators of the form

$$\begin{aligned} L &= \frac{\partial}{\partial x_1}\left(a\frac{\partial}{\partial x_1}\right) + \frac{\partial}{\partial x_2}\left(a\frac{\partial}{\partial x_2}\right) \\ &= \nabla.\ (a\nabla) \end{aligned} \tag{2.5}$$

where the scalar function a depends on the coordinate and may represent some specific parameter of the medium, such as density, porosity, etc. At this point it may be useful to recall some notation which is widely used in physics and mechanics. The ∇ operator can be considered as a vector consisting of the components $\frac{\partial}{\partial x_1}$ and $\frac{\partial}{\partial x_2}$. When applied to a scalar function u, this operator is nothing but the *gradient* operator, since it yields a vector with the components $\frac{\partial u}{\partial x_1}$ and $\frac{\partial u}{\partial x_2}$ as is shown in (2.4). The dot notation allows dot products of vectors in $\mathbb{R}^2$ to be defined. These vectors can include partial differential operators. For example, the dot product $\nabla.u$ of ∇ with $u = \binom{u_1}{u_2}$ yields the scalar quantity,

$$\frac{\partial u_1}{\partial x_1} + \frac{\partial u_2}{\partial x_2},$$

which is called the *divergence* of the vector function $\vec{u} = \binom{u_1}{u_2}$. Applying this *divergence operator* to $u = a\nabla$, where a is a scalar function, yields the L operator in (2.5). The divergence of the vector function $\vec{v}$ is often denoted by div $\vec{v}$ or $\nabla.\vec{v}$. Thus,

$$\text{div}\ \vec{v} = \nabla.\vec{v} = \frac{\partial v_1}{\partial x_1} + \frac{\partial v_2}{\partial x_2}.$$

The closely related operator

$$L = \frac{\partial}{\partial x_1}\left(a_1 \frac{\partial}{\partial x_1}\right) + \frac{\partial}{\partial x_2}\left(a_2 \frac{\partial}{\partial x_2}\right)$$
$$= \nabla \ \ (\vec{a}.\ \nabla) \tag{2.6}$$

is a further generalization of the Laplacean operator Δ in the case where the medium is *anisotropic* and *inhomogeneous.* The coefficients a_1, a_2 depend on the space variable x and reflect the position as well as the directional dependence of the material properties, such as porosity in the case of fluid flow or dielectric constants in electrostatics. In fact, the above operator can be viewed as a particular case of $L = \nabla.(A\nabla)$, where A is a 2×2 matrix which acts on the two components of ∇.

2.1.2 THE CONVECTION DIFFUSION EQUATION

Many physical problems involve a combination of "diffusion" and "convection" phenomena. Such phenomena are modeled by the convection-diffusion equation

$$\frac{\partial u}{\partial t} + b_1 \frac{\partial u}{\partial x_1} + b_2 \frac{\partial u}{\partial x_2} = \nabla.(a\nabla)u + f$$

or

$$\frac{\partial u}{\partial t} + \vec{b}.\nabla u = \nabla.(a\nabla)u + f$$

the steady-state version of which can be written as

$$-\nabla.(a\nabla)u + \vec{b}.\nabla u = f. \tag{2.7}$$

Problems of this type are often used as model problems because they represent the simplest form of conservation of mass in fluid mechanics. Note that the vector $\vec{b}$ is sometimes quite large, which may cause some difficulties either to the discretization schemes or to the iterative solution techniques.

2.2 FINITE DIFFERENCE METHODS

The *finite difference* method is based on local approximations of the partial derivatives in a Partial Differential Equation, which are derived by low order Taylor series expansions. The method is quite simple to define and rather easy to implement. Also, it is particularly appealing for simple regions, such as rectangles, and when uniform meshes are used. The matrices that result from these discretizations are often well structured, which means that they typically consist of a few nonzero diagonals. Another advantage is that there are a number of "fast solvers" for constant coefficient problems, which can deliver the solution in logarithmic time per grid point. This means the total number of operations is of the order of $n \log(n)$ where n is the total number of discretization points. This section gives an overview of finite difference discretization techniques.

2.2.1 BASIC APPROXIMATIONS

The simplest way to approximate the first derivative of a function u at the point x is via the formula

$$\left(\frac{du}{dx}\right)(x) \approx \frac{u(x+h)-u(x)}{h}. \tag{2.8}$$

When u is differentiable at x, then the limit of the above ratio when h tends to zero is the derivative of u at x. For a function that is C^4 in the neighborhood of x, we have by Taylor's formula

$$u(x+h) = u(x) + h\frac{du}{dx} + \frac{h^2}{2}\frac{d^2u}{dx^2} + \frac{h^3}{6}\frac{d^3u}{dx^3} + \frac{h^4}{24}\frac{d^4u}{dx^4}(\xi_+), \tag{2.9}$$

for some ξ_+ in the interval $(x, x+h)$. Therefore, the above approximation (2.8) satisfies

$$\frac{du}{dx} = \frac{u(x+h)-u(x)}{h} - \frac{h}{2}\frac{d^2u(x)}{dx^2} + O(h^2). \tag{2.10}$$

The formula (2.9) can be rewritten with h replaced by $-h$ to obtain

$$u(x-h) = u(x) - h\frac{du}{dx} + \frac{h^2}{2}\frac{d^2u}{dx^2} - \frac{h^3}{6}\frac{d^3u}{dx^3} + \frac{h^4}{24}\frac{d^4u(\xi_-)}{dx^4}, \tag{2.11}$$

in which ξ_- belongs to the interval $(x-h, x)$. Adding (2.9) and (2.11), dividing through by h^2, and using the mean value theorem for the fourth order derivatives results in the following approximation of the second derivative

$$\frac{d^2u(x)}{dx^2} = \frac{u(x+h)-2u(x)+u(x-h)}{h^2} - \frac{h^2}{12}\frac{d^4u(\xi)}{dx^4}, \tag{2.12}$$

where $\xi_- \le \xi \le \xi_+$. The above formula is called a *centered difference approximation* of the second derivative since the point at which the derivative is being approximated is the center of the points used for the approximation. The dependence of this derivative on the values of u at the points involved in the approximation is often represented by a "stencil" or "molecule," shown in Figure 2.2.

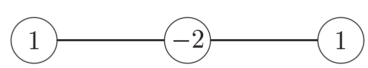

Figure 2.2 *The three-point stencil for the centered difference approximation to the second order derivative.*

The approximation (2.8) for the first derivative is *forward* rather than centered. Also, a *backward* formula can be used which consists of replacing h with $-h$ in (2.8). The two formulas can also be averaged to obtain the *centered difference* formula:

$$\frac{du(x)}{dx} \approx \frac{u(x+h)-u(x-h)}{2\,h}. \tag{2.13}$$

It is easy to show that the above centered difference formula is of the second order, while (2.8) is only first order accurate. Denoted by δ^+ and δ^-, the forward and backward difference operators are defined by

$$\delta^+ u(x) = u(x+h) - u(x) \tag{2.14}$$

$$\delta^- u(x) = u(x) - u(x-h). \tag{2.15}$$

All previous approximations can be rewritten using these operators.

In addition to standard first order and second order derivatives, it is sometimes necessary to approximate the second order operator

$$\frac{d}{dx}\left[a(x)\,\frac{d}{dx}\right].$$

A centered difference formula for this, which has second order accuracy, is given by

$$\begin{aligned}\frac{d}{dx}\left[a(x)\,\frac{du}{dx}\right] &= \frac{1}{h^2}\delta^+\left(a_{i-1/2}\,\delta^- u\right) + O(h^2) \qquad (2.16)\\ &\approx \frac{a_{i+1/2}(u_{i+1}-u_i) - a_{i-1/2}(u_i - u_{i-1})}{h^2}.\end{aligned}$$

2.2.2 DIFFERENCE SCHEMES FOR THE LAPLACEAN OPERATOR

If the approximation (2.12) is used for both the $\frac{\partial^2}{\partial x_1^2}$ and $\frac{\partial^2}{\partial x_2^2}$ terms in the Laplacean operator, using a mesh size of h_1 for the x_1 variable and h_2 for the x_2 variable, the following second order accurate approximation results:

$$\begin{aligned}\Delta u(x) \approx {} & \frac{u(x_1+h_1,x_2) - 2u(x_1,x_2) + u(x-h_1,x_2)}{h_1^2} + \\ & \frac{u(x_1,x_2+h_2) - 2u(x_1,x_2) + u(x_1,x_2-h_2)}{h_2^2}.\end{aligned}$$

In the particular case where the mesh sizes h_1 and h_2 are the same and equal to a mesh size h, the approximation becomes

$$\begin{aligned}\Delta u(x) \approx \frac{1}{h^2}&\left[u(x_1+h,x_2) + u(x_1-h,x_2) + u(x_1,x_2+h)\right.\\ &\left. + u(x_1,x_2-h) - 4u(x_1,x_2)\right],\end{aligned} \tag{2.17}$$

which is called the five-point centered approximation to the Laplacean. The stencil of this finite difference approximation is illustrated in (a) of Figure 2.3.

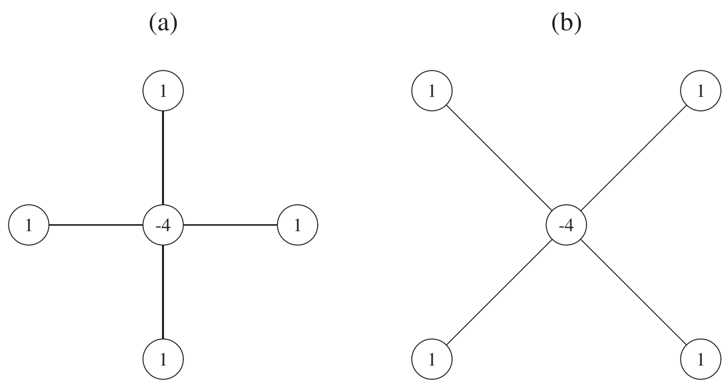

Figure 2.3 *Five-point stencils for the centered difference approximation to the Laplacean operator: (a) the standard stencil, (b) the skewed stencil.*

Another approximation may be obtained by exploiting the four points $u(x_1 \pm h, x_2 \pm h)$ located on the two diagonal lines from $u(x_1, x_2)$. These points can be used in the same manner as in the previous approximation except that the mesh size has changed. The corresponding stencil is illustrated in (b) of Figure 2.3.

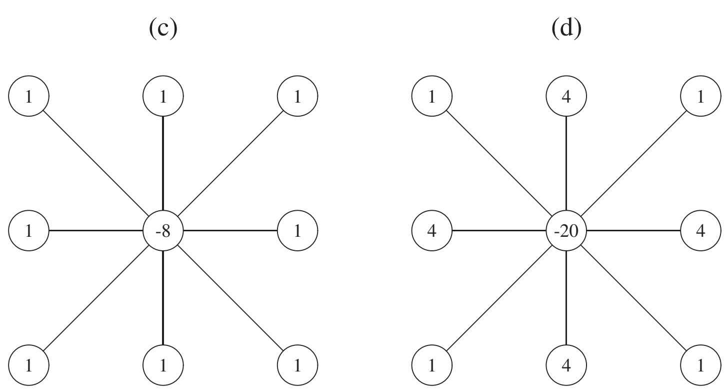

Figure 2.4 *Two nine-point centered difference stencils for the Laplacean operator.*

The approximation (2.17) is second order accurate and the error takes the form

$$\frac{h^2}{12}\left(\frac{\partial^4 u}{\partial^4 x_1} + \frac{\partial^4 u}{\partial^4 x_2}\right) + O(h^3).$$

There are other schemes that utilize nine-point formulas as opposed to five-point formulas. Two such schemes obtained by combining the standard and skewed stencils described above are shown in Figure 2.4. Both approximations (c) and (d) are second order accurate. However, (d) is sixth order for harmonic functions, i.e., functions whose Laplacean is zero.

2.2.3 FINITE DIFFERENCES FOR 1-D PROBLEMS

Consider the one-dimensional equation,

$$-u''(x) = f(x) \text{ for } x \in (0,1) \tag{2.18}$$
$$u(0) = u(1) = 0. \tag{2.19}$$

The interval [0,1] can be discretized uniformly by taking the $n + 2$ points

$$x_i = i \times h, \; i = 0, \ldots, n+1$$

where $h = 1/(n+1)$. Because of the Dirichlet boundary conditions, the values $u(x_0)$ and $u(x_{n+1})$ are known. At every other point, an approximation u_i is sought for the exact solution $u(x_i)$.

If the centered difference approximation (2.12) is used, then by the equation (2.18) expressed at the point x_i, the unknowns u_i, u_{i-1}, u_{i+1} satisfy the relation

$$-u_{i-1} + 2u_i - u_{i+1} = h^2 \, f_i,$$

in which $f_i \equiv f(x_i)$. Notice that for $i = 1$ and $i = n$, the equation will involve u_0 and u_{n+1} which are known quantities, both equal to zero in this case. Thus, for $n = 6$, the linear system obtained is of the form

$$Ax = f$$

where

$$A = \frac{1}{h^2} \begin{pmatrix} 2 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & -1 & 2 & -1 & & \\ & & -1 & 2 & -1 & \\ & & & -1 & 2 & -1 \\ & & & & -1 & 2 \end{pmatrix}.$$

2.2.4 UPWIND SCHEMES

Consider now the one-dimensional version of the convection-diffusion equation (2.7) in which the coefficients a and b are constant, and $f = 0$, using Dirichlet boundary conditions,

$$\begin{cases} -a\,u'' + b\,u' & = & 0, \quad 0 < x < L = 1 \\ u(0) = 0, \; u(L) & = & 1. \end{cases} \tag{2.20}$$

In this particular case, it is easy to verify that the exact solution to the above equation is given by

$$u(x) = \frac{1 - e^{Rx}}{1 - e^{R}}$$

where R is the so-called Péclet number defined by $R = bL/a$. Now consider the approximate solution provided by using the centered difference schemes seen above, for both the first- and second order derivatives. The equation for unknown number i becomes

$$b\frac{u_{i+1}-u_{i-1}}{2h} - a\frac{u_{i+1}-2u_i+u_{i-1}}{h^2} = 0,$$

or, defining $c = Rh/2$,

$$-(1-c)u_{i+1} + 2u_i - (1+c)u_{i-1} = 0. \tag{2.21}$$

This is a second order homogeneous linear difference equation and the usual way to solve it is to seek a general solution in the form $u_j = r^j$. Substituting in (2.21), r must satisfy

$$(1-c)r^2 - 2r + (c+1) = 0.$$

Therefore, $r_1 = 1$ is a root and the second root is $r_2 = (1+c)/(1-c)$. The general solution of the above difference equation is now sought as a linear combination of the two solutions corresponding to these two roots,

$$u_i = \alpha r_1^i + \beta r_2^i = \alpha + \beta \left(\frac{1+c}{1-c}\right)^i .$$

Because of the boundary condition $u_0 = 0$, it is necessary that $\beta = -\alpha$. Likewise, the boundary condition $u_{n+1} = 1$ yields

$$\alpha = \frac{1}{1-\sigma^{n+1}} \quad \text{with} \quad \sigma \equiv \frac{1+c}{1-c}.$$

Thus, the solution is

$$u_i = \frac{1-\sigma^i}{1-\sigma^{n+1}}.$$

When $h > 2/R$ the factor σ becomes negative and the above approximations will oscillate around zero. In contrast, the exact solution is positive and monotone in the range $[0, 1]$. In this situation the solution is very inaccurate regardless of the arithmetic. In other words, the scheme itself creates the oscillations. To avoid this, a small enough mesh h can be taken to ensure that $c < 1$. The resulting approximation is in much better agreement with the exact solution. Unfortunately, this condition can limit the mesh size too drastically for large values of b.

Note that when $b < 0$, the oscillations disappear since $\sigma < 1$. In fact, a linear algebra interpretation of the oscillations comes from comparing the tridiagonal matrices obtained from the discretization. Again, for the case $n = 6$, the tridiagonal matrix resulting from discretizing the equation (2.7) takes the form

$$A = \frac{1}{h^2}\begin{pmatrix} 2 & -1+c & & & & \\ -1-c & 2 & -1+c & & & \\ & -1-c & 2 & -1+c & & \\ & & -1-c & 2 & -1+c & \\ & & & -1-c & 2 & -1+c \\ & & & & -1-c & 2 \end{pmatrix}.$$

The above matrix is no longer a diagonally dominant M-matrix. Observe that if the backward difference formula for the first order derivative is used, we obtain

$$b\,\frac{u_i-u_{i-1}}{h} - a\frac{u_{i-1}-2u_i+u_{i+1}}{h^2} = 0.$$

Then (weak) diagonal dominance is preserved if $b > 0$. This is because the new matrix obtained for the above backward scheme is

$$A = \frac{1}{h^2}\begin{pmatrix} 2+c & -1 & & & & \\ -1-c & 2+c & -1 & & & \\ & -1-c & 2+c & -1 & & \\ & & -1-c & 2+c & -1 & \\ & & & -1-c & 2+c & -1 \\ & & & & -1-c & 2+c \end{pmatrix}$$

where c is now defined by $c = Rh$. Each diagonal term a_{ii} gets reinforced by the positive term c while each subdiagonal term $a_{i,i-1}$ increases by the same amount in absolute value. In the case where $b < 0$, the forward difference formula

$$b\frac{u_{i+1} - u_i}{h} - a\frac{u_{i-1} - 2u_i + u_{i+1}}{h^2} = 0$$

can be used to achieve the same effect. Generally speaking, if b depends on the space variable x, the effect of weak-diagonal dominance can be achieved by simply adopting the following discretization known as an "upwind scheme":

$$b\,\frac{\delta_i^* u_i}{h} - a\,\frac{u_{i-1} - 2u_i + u_{i+1}}{h^2} = 0$$

where

$$\delta_i^* = \begin{cases} \delta_i^- & \text{if} \quad b > 0 \\ \delta_i^+ & \text{if} \quad b < 0. \end{cases}$$

The above difference scheme can be rewritten by introducing the sign function $\mathrm{sign}\,(b) = |b|/b$. The approximation to u' at x_i is then defined by

$$u'(x_i) \approx \frac{1}{2}(1 - \mathrm{sign}(b))\frac{\delta^+ u_i}{h} + \frac{1}{2}(1 + \mathrm{sign}(b))\frac{\delta^- u_i}{h}.$$

Making use of the notation

$$(x)^+ = \frac{1}{2}(x + |x|), \quad (x)^- = \frac{1}{2}(x - |x|), \tag{2.22}$$

a slightly more elegant formula can be obtained by expressing the approximation of the product $b(x_i)u'(x_i)$,

$$\begin{aligned} b(x_i)u'(x_i) &\approx \frac{1}{2}(b_i - |b_i|)\frac{\delta^+ u_i}{h} + \frac{1}{2}(b_i + |b_i|)\frac{\delta^- u_i}{h} \\ &\approx \frac{1}{h}\left[-b_i^+ u_{i-1} + |b_i| u_i + b_i^- u_{i+1}\right], \end{aligned} \tag{2.23}$$

where b_i stands for $b(x_i)$. The diagonal term in the resulting tridiagonal matrix is nonnegative, the offdiagonal terms are nonpositive, and the diagonal term is the negative sum of the offdiagonal terms. This property characterizes upwind schemes.

A notable disadvantage of upwind schemes is the low order of approximation which they yield. An advantage is that upwind schemes yield linear systems that are easier to solve by iterative methods.

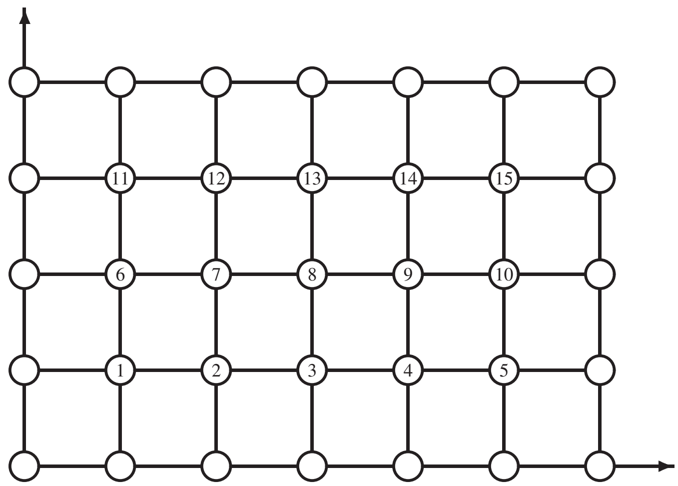

Figure 2.5 *Natural ordering of the unknowns for a 7×5 two-dimensional grid.*

2.2.5 FINITE DIFFERENCES FOR 2-D PROBLEMS

Similar to the previous case, consider this simple problem,

$$-\left(\frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2}\right) = f \quad \text{in } \Omega \tag{2.24}$$

$$u = 0 \quad \text{on } \Gamma \tag{2.25}$$

where Ω is now the rectangle $(0, l_1) \times (0, l_2)$ and Γ its boundary. Both intervals can be discretized uniformly by taking $n_1 + 2$ points in the x_1 direction and $n_2 + 2$ points in the x_2 directions:

$$x_{1,i} = i \times h_1, i = 0, \ldots, n_1 + 1 \quad x_{2,j} = j \times h_2, j = 0, \ldots, n_2 + 1$$

where

$$h_1 = \frac{l_1}{n_1 + 1} \quad h_2 = \frac{l_2}{n_2 + 1}.$$

Since the values at the boundaries are known, we number only the interior points, i.e., the points $(x_{1,i}, x_{2,j})$ with $0 < i < n_1$ and $0 < j < n_2$. The points are labeled from the bottom up, one horizontal line at a time. This labeling is called *natural ordering* and is shown in Figure 2.5 for the very simple case when $n_1 = 7$ and $n_2 = 5$. The pattern of the matrix corresponding to the above equations appears in Figure 2.6.

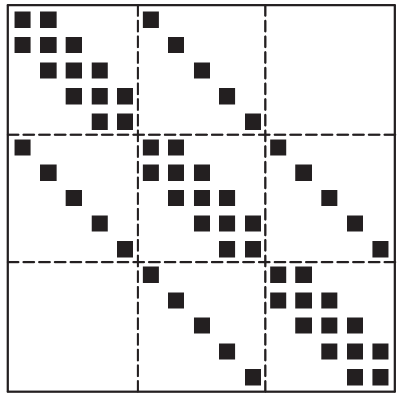

Figure 2.6 *Pattern of matrix associated with the 7×5 finite difference mesh of Figure 2.5.*

To be more accurate, the matrix has the following block structure:

$$A = \frac{1}{h^2}\begin{pmatrix} B & -I & \\ -I & B & -I \\ & -I & B \end{pmatrix} \quad \text{with} \quad B = \begin{pmatrix} 4 & -1 & & \\ -1 & 4 & -1 & \\ & -1 & 4 & -1 \\ & & -1 & 4 \end{pmatrix}.$$

2.3 THE FINITE ELEMENT METHOD

The finite element method is best illustrated with the solution of a simple elliptic Partial Differential Equation in a two-dimensional space. Consider again Poisson's equation (2.24) with the Dirichlet boundary condition (2.25), where Ω is a bounded open domain in $\mathbb{R}^2$ and Γ its boundary. The Laplacean operator

$$\Delta = \frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2}$$

appears in many models of physical and mechanical phenomena. Equations involving the more general elliptic operators (2.5) and (2.6) can be treated in the same way as Poisson's equation (2.24) and (2.25), at least from the viewpoint of the numerical solutions techniques.

An essential ingredient for understanding the finite element method is *Green's formula*. The setting for this formula is an open set Ω whose boundary consists of a closed and smooth curve Γ as illustrated in Figure 2.1. A vector-valued function $\vec{v} = \binom{v_1}{v_2}$, which is continuously differentiable in Ω, is given. The *divergence theorem* in two-dimensional spaces states that

$$\int_\Omega \text{div}\vec{v}\, dx = \int_\Gamma \vec{v}.\vec{n}\ \ ds. \tag{2.26}$$

The dot in the right-hand side represents a dot product of two vectors in $\mathbb{R}^2$. In this case it is between the vector $\vec{v}$ and the unit vector $\vec{n}$ which is normal to Γ at the point of consideration and oriented outward. To derive Green's formula, consider a scalar function v and a vector function $\vec{w} = \binom{w_1}{w_2}$. By standard differentiation,

$$\nabla.(v\vec{w}) = (\nabla v).\vec{w} + v\nabla.\vec{w},$$

which expresses $\nabla v.\vec{w}$ as

$$\nabla v.\vec{w} = -v\nabla.\vec{w} + \nabla.(v\vec{w}). \tag{2.27}$$

Integrating the above equality over Ω and using the divergence theorem, we obtain

$$\begin{aligned} \int_\Omega \nabla v.\vec{w}\ \ dx &= -\int_\Omega v\nabla.\vec{w}\ \ dx + \int_\Omega \nabla.(v\vec{w})\, dx \\ &= -\int_\Omega v\nabla.\vec{w}\ \ dx + \int_\Gamma v\vec{w}.\vec{n}\, ds. \end{aligned} \tag{2.28}$$

The above equality can be viewed as a generalization of the standard integration by part formula in calculus. Green's formula results from (2.28) by simply taking a vector $\vec{w}$ which is itself a gradient of a scalar function u, namely, $\vec{w} = \nabla u$,

$$\int_\Omega \nabla v.\nabla u\ \ dx = -\int_\Omega v\nabla.\nabla u\ \ dx + \int_\Gamma v\nabla u.\vec{n}\, ds.$$

Observe that $\nabla.\nabla u = \Delta u$. Also the function $\nabla u.\vec{n}$ is called the *normal derivative* and is denoted by

$$\nabla u.\vec{n} = \frac{\partial u}{\partial \vec{n}}.$$

With this, we obtain Green's formula

$$\int_\Omega \nabla v.\nabla u\ \ dx = -\int_\Omega v\Delta u\ \ dx + \int_\Gamma v\frac{\partial u}{\partial \vec{n}}\, ds. \tag{2.29}$$

We now return to the initial problem (2.24-2.25). To solve this problem approximately, it is necessary to (1) take approximations to the unknown function u, and (2) translate the equations into a system which can be solved numerically. The options for approximating u are numerous. However, the primary requirement is that these approximations should be in a (small) finite dimensional space. There are also some additional desirable numerical properties. For example, it is difficult to approximate high degree polynomials numerically. To extract systems of equations which yield the solution, it is common to use the *weak formulation* of the problem. Let us define

$$\begin{aligned} a(u,v) &\equiv \int_\Omega \nabla u.\nabla v\ \ dx = \int_\Omega \left(\frac{\partial u}{\partial x_1}\frac{\partial v}{\partial x_1} + \frac{\partial u}{\partial x_2}\frac{\partial v}{\partial x_2} \right) dx, \\ (f,v) &\equiv \int_\Omega fv\, dx. \end{aligned}$$

An immediate property of the functional a is that it is *bilinear.* That means that it is linear with respect to u and v, namely,

$$\begin{aligned} a(\mu_1 u_1 + \mu_2 u_2, v) &= \mu_1 a(u_1, v) + \mu_2 a(u_2, v), \quad \forall \mu_1, \mu_2 \in \mathbb{R}, \\ a(u, \lambda_1 v_1 + \lambda_2 v_2) &= \lambda_1 a(u, v_1) + \lambda_2 a(u, v_2), \quad \forall \lambda_1, \lambda_2 \in \mathbb{R}. \end{aligned}$$

Notice that (u, v) denotes the L_2-inner product of u and v in Ω, i.e.,

$$(u, v) = \int_\Omega u(x)v(x)dx,$$

then, for functions satisfying the Dirichlet boundary conditions, which are at least twice differentiable, Green's formula (2.29) shows that

$$a(u, v) = -(\Delta u, v).$$

The weak formulation of the initial problem (2.24-2.25) consists of selecting a subspace of reference V of L^2 and then defining the following problem:

$$\text{Find} \quad u \in V \quad \text{such that} \quad a(u, v) = (f, v), \quad \forall\, v \in V. \tag{2.30}$$

In order to understand the usual choices for the space V, note that the definition of the weak problem only requires the dot products of the gradients of u and v and the functions f and v to be L_2–integrable. The most general V under these conditions is the space of all functions whose derivatives up to the first order are in L_2. This is known as $H^1(\Omega)$. However, this space does not take into account the boundary conditions. The functions in V must be restricted to have zero values on Γ. The resulting space is called $H_0^1(\Omega)$.

The finite element method consists of approximating the weak problem by a finite-dimensional problem obtained by replacing V with a subspace of functions that are defined as low-degree polynomials on small pieces (elements) of the original domain.

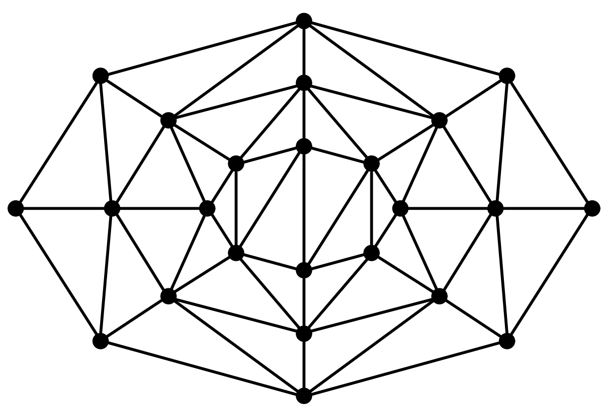

Figure 2.7 *Finite element triangulation of a domain.*

Consider a region Ω in the plane which is triangulated as shown in Figure 2.7. In this example, the domain is simply an ellipse but the external enclosing curve is not shown. The original domain is thus approximated by the union Ω_h of m triangles K_i,

$$\Omega_h = \bigcup_{i=1}^{m} K_i.$$

For the triangulation to be valid, these triangles must have no vertex that lies on the edge of any other triangle. The *mesh size* h is defined by

$$h = \max_{i=1,\dots,m} \text{diam}(K_i)$$

where diam(K), the diameter of a triangle K, is the length of its longest side.

Then the finite dimensional space V_h is defined as the space of all functions which are piecewise linear and continuous on the polygonal region Ω_h, and which vanish on the boundary Γ. More specifically,

$$V_h = \{\phi \mid \phi_{|\Omega_h} \text{continuous}, \phi_{|\Gamma_h} = 0,\ \phi_{|K_j} \text{linear} \,\forall\, j\}.$$

Here, $\phi_{|X}$ represents the restriction of the function ϕ to the subset X. If $x_j, j = 1, \dots, n$ are the nodes of the triangulation, then a function ϕ_j in V_h can be associated with each node x_j, so that the family of functions ϕ_j's satisfies the following conditions:

$$\phi_j(x_i) = \delta_{ij} = \begin{cases} 1 & \text{if} \quad x_i = x_j \\ 0 & \text{if} \quad x_i \neq x_j \end{cases}. \tag{2.31}$$

These conditions define $\phi_i, i = 1, \dots, n$ uniquely. In addition, the ϕ_i's form a basis of the space V_h.

Each function of V_h can be expressed as

$$\phi(x) = \sum_{i=1}^{n} \xi_i \phi_i(x).$$

The finite element approximation consists of writing the Galerkin condition (2.30) for functions in V_h. This defines the approximate problem:

$$\text{Find} \quad u \in V_h \quad \text{such that} \quad a(u, v) = (f, v), \;\; \forall \;\, v \in V_h. \tag{2.32}$$

Since u is in V_h, there are n degrees of freedom. By the linearity of a with respect to v, it is only necessary to impose the condition $a(u, \phi_i) = (f, \phi_i)$ for $i = 1, \dots, n$. This results in n constraints.

Writing the desired solution u in the basis $\{\phi_i\}$ as

$$u = \sum_{i=1}^{n} \xi_i \phi_i(x)$$

and substituting in (2.32) gives the linear problem

$$\sum_{j=1}^{n} \alpha_{ij} \xi_i = \beta_i \tag{2.33}$$

where

$$\alpha_{ij} = a(\phi_i, \phi_j), \quad \beta_i = (f, \phi_i).$$

The above equations form a linear system of equations

$$Ax = b,$$

in which the coefficients of A are the α_{ij}'s; those of b are the β_j's. In addition, A is a *Symmetric Positive Definite* matrix. Indeed, it is clear that

$$\int_\Omega \nabla\phi_i \nabla\phi_j \; dx = \int_\Omega \nabla\phi_j \nabla\phi_i \; dx,$$

which means that $\alpha_{ij} = \alpha_{ji}$. To see that A is positive definite, first note that $a(u, u) \geq 0$ for any function u. If $a(\phi, \phi) = 0$ for a function in V_h, then it must be true that $\nabla \phi = 0$ *almost everywhere* in Ω_h. Since ϕ is linear in each triangle and continuous, then it is clear that it must be constant on all Ω. Since, in addition, it vanishes on the boundary, then it must be equal to zero on all of Ω. The result follows by exploiting the relation

$$(A\xi, \xi) = a(\phi, \phi) \quad \text{with} \quad \phi = \sum_{i=1}^{n} \xi_i \phi_i,$$

which is valid for any vector $\{\xi_i\}_{i=1,\ldots,n}$.

Another important observation is that the matrix A is also sparse. Indeed, α_{ij} is nonzero only when the two basis functions ϕ_i and ϕ_j have common support triangles, or equivalently when the nodes i and j are the vertices of a common triangle. Specifically, for a given node i, the coefficient α_{ij} will be nonzero only when the node j is one of the nodes of a triangle that is adjacent to node i.

In practice, the matrix is built by summing up the contributions of all triangles by applying the formula

$$a(\phi_i, \phi_j) = \sum_K a_K(\phi_i, \phi_j)$$

in which the sum is over all the triangles K and

$$a_K(\phi_i, \phi_j) = \int_K \nabla \phi_i \ \nabla \phi_j \ dx.$$

Note that $a_K(\phi_i, \phi_j)$ is zero unless the nodes i and j are both vertices of K. Thus, a triangle contributes nonzero values to its three vertices from the above formula. The 3×3 matrix

$$A_K = \begin{pmatrix} a_K(\phi_i, \phi_i) & a_K(\phi_i, \phi_j) & a_K(\phi_i, \phi_k) \\ a_K(\phi_j, \phi_i) & a_K(\phi_j, \phi_j) & a_K(\phi_j, \phi_k) \\ a_K(\phi_k, \phi_i) & a_K(\phi_k, \phi_j) & a_K(\phi_k, \phi_k) \end{pmatrix}$$

associated with the triangle $K(i, j, k)$ with vertices i, j, k is called an *element stiffness matrix*. In order to form the matrix A, it is necessary to sum up all the contributions $a_K(\phi_k, \phi_m)$ to the position k, m of the matrix. This process is called an *assembly* process. In the assembly, the matrix is computed as

$$A = \sum_{e=1}^{nel} A^{[e]}, \tag{2.34}$$

in which nel is the number of elements. Each of the matrices $A^{[e]}$ is of the form

$$A^{[e]} = P_e A_{K_e} P_e^T$$

where A_{K_e} is the element matrix for the element K_e as defined above. Also P_e is an $n \times 3$ Boolean connectivity matrix which maps the coordinates of the 3×3 matrix A_{K_e} into the coordinates of the full matrix A.

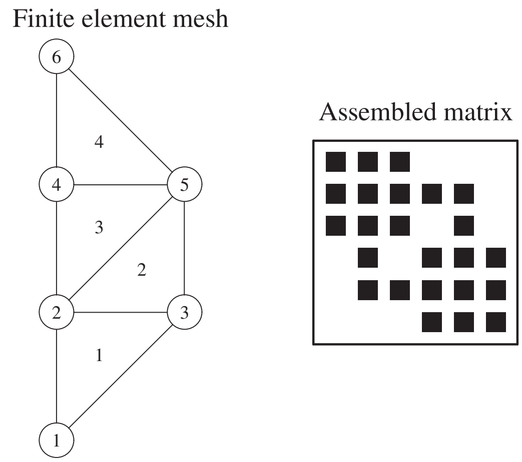

Figure 2.8 *A simple finite element mesh and the pattern of the corresponding assembled matrix.*

Example 2.1 The assembly process can be illustrated with a very simple example. Consider the finite element mesh shown in Figure 2.8. The four elements are numbered from bottom to top as indicated by the labels located at their centers. There are six nodes in this mesh and their labeling is indicated in the circled numbers. The four matrices $A^{[e]}$ associated with these elements are shown in Figure 2.9. Thus, the first element will contribute to the nodes $1, 2, 3$, the second to nodes $2, 3, 5$, the third to nodes $2, 4, 5$, and the fourth to nodes $4, 5, 6$.

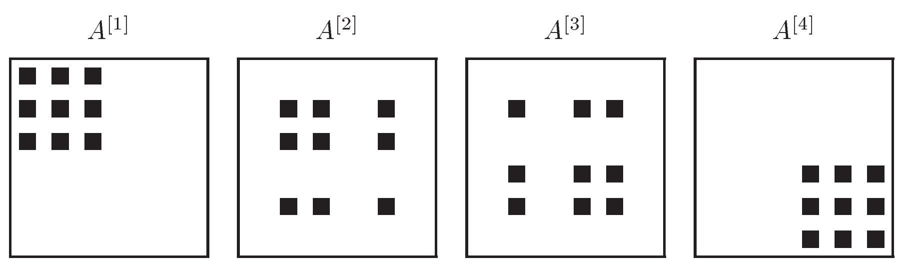

Figure 2.9 *The element matrices $A^{[e]}$, $e = 1, \ldots, 4$ for the finite element mesh shown in Figure 2.8.*

In fact there are two different ways to represent and use the matrix A. We can form all the element matrices one by one and then we can store them, e.g., in an $nel \times 3 \times 3$ rectangular array. This representation is often called the *unassembled form* of A. Then the matrix A may be assembled if it is needed. However, element stiffness matrices can also be used in different ways without having to assemble the matrix. For example, *frontal techniques* are direct solution methods that take the linear system in unassembled form and compute the solution by a form of Gaussian elimination. There are also iterative solution techniques which work directly with unassembled matrices. One of the main operations required in many iterative methods is to compute $y = Ax$, the product of the matrix A by an arbitrary vector x. In unassembled form, this can be achieved as follows:

$$y = Ax = \sum_{e=1}^{nel} A^{[e]}x \; = \; \sum_{e=1}^{nel} P_e A_{K_e}(P_e^T x). \tag{2.35}$$

Thus, the product $P_e^T x$ gathers the x data associated with the e-element into a 3-vector consistent with the ordering of the matrix A_{K_e}. After this is done, this vector must be multiplied by A_{K_e}. Finally, the result is added to the current y vector in appropriate locations determined by the P_e array. This sequence of operations must be done for each of the nel elements.

A more common and somewhat more appealing technique is to perform the assembly of the matrix. All the elements are scanned one by one and the nine associated contributions $a_K(\phi_k, \phi_m),\ k, m \ \in \ \{i, j, k\}$ added to the corresponding positions in the global "stiffness" matrix. The assembled matrix must now be stored but the element matrices may be discarded. The structure of the assembled matrix depends on the ordering of the nodes. To facilitate the computations, a widely used strategy transforms all triangles into a reference triangle with vertices $(0, 0), (0, 1), (1, 0)$. The area of the triangle is then simply the determinant of the Jacobian of the transformation that allows passage from one set of axes to the other.

Simple boundary conditions such as Neumann or Dirichlet do not cause any difficulty. The simplest way to handle Dirichlet conditions is to include boundary values as unknowns and modify the assembled system to incorporate the boundary values. Thus, each equation associated with the boundary point in the assembled system is replaced by the equation $u_i = f_i$. This yields a small identity block hidden within the linear system. For Neumann conditions, Green's formula will give rise to the equations

$$\int_\Omega \nabla u.\nabla \phi_j \;\; dx = \int_\Omega f\phi_j dx + \int_\Gamma \phi_j \frac{\partial u}{\partial \vec{n}}\, ds, \tag{2.36}$$

which will involve the Neumann data $\frac{\partial u}{\partial \vec{n}}$ over the boundary. Since the Neumann data is typically given at some points only (the boundary nodes), linear interpolation (trapezoidal rule) or the mid-line value (midpoint rule) can be used to approximate the integral. Note that (2.36) can be viewed as the j-th equation of the linear system. Another important point is that if the boundary conditions are only of Neumann type, then the resulting system is singular. An equation must be removed, or the linear system must be solved by taking this singularity into account.

2.4 MESH GENERATION AND REFINEMENT

Generating a finite element triangulation can be done quite easily by exploiting some initial grid and then refining the mesh a few times either uniformly or in specific areas. The simplest refinement technique consists of taking the three midpoints of a triangle, thus creating four smaller triangles from a larger triangle and losing one triangle, namely, the original one. A systematic use of one level of this strategy is illustrated for the mesh in Figure 2.8, and is shown in Figure 2.10.

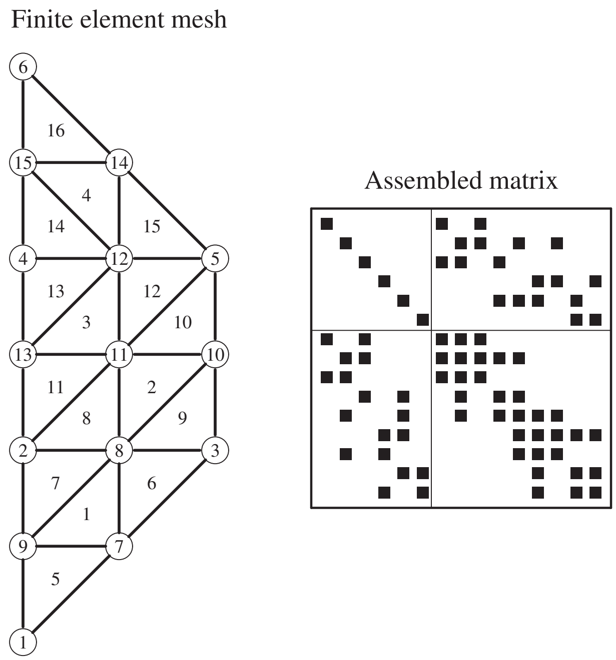

Figure 2.10 *The simple finite element mesh of Figure 2.8 after one level of refinement and the corresponding matrix.*

One advantage of this approach is that it preserves the angles of the original triangulation. This is an important property since the angles on a good quality triangulation must satisfy certain bounds. On the other hand, the indiscriminate use of the uniform refinement strategy may lead to some inefficiencies. Indeed, it is desirable to introduce more triangles in areas where the solution is likely to have large variations. In terms of vertices, midpoints should be introduced only where needed. To obtain standard finite element triangles, the points that have been created on the edges of a triangle must be linked to existing vertices in the triangle. This is because no vertex of a triangle is allowed to lie on the edge of another triangle.

Figure 2.11 shows three possible cases that can arise. The original triangle is (a). In (b), only one new vertex (numbered 4) has appeared on one edge of the triangle and it is joined to the vertex opposite to it. In (c), two new vertices appear inside the original triangle. There is no alternative but to join vertices (4) and (5). However, after this is done, either vertices (4) and (3) or vertices (1) and (5) must be joined. If angles are desired that will not become too small with further refinements, the second choice is clearly better in this case. In fact, various strategies for improving the quality of the triangles have been devised. The final case (d) corresponds to the "uniform refinement" case where all edges have been split in two. There are three new vertices and four new elements, and the larger initial element is removed.

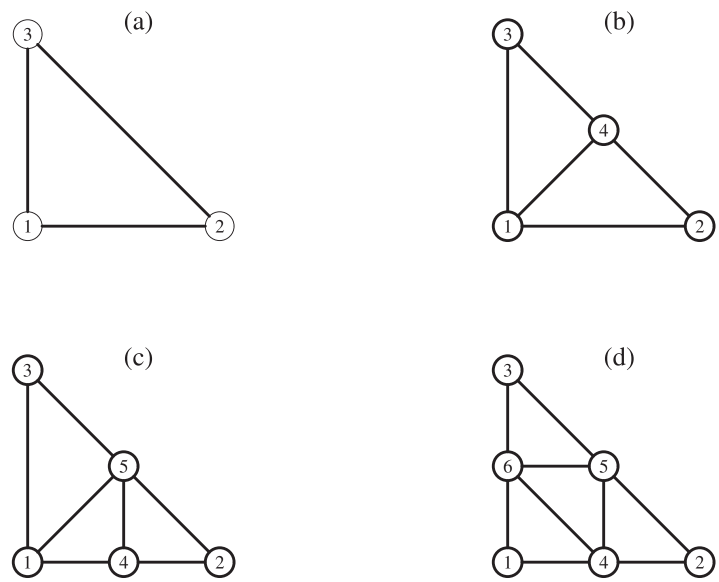

Figure 2.11 *Original triangle (a) and three possible refinement scenarios.*

2.5 FINITE VOLUME METHOD

The finite volume method is geared toward the solution of conservation laws of the form:

$$\frac{\partial u}{\partial t} + \nabla.\vec{F} = Q. \tag{2.37}$$

In the above equation, $\vec{F}(u,t)$ is a certain vector function of u and time, possibly nonlinear. This is called the "flux vector." The *source term* Q is a function of space and time. We now apply the principle used in the weak formulation, described before. Multiply both sides by a test function w, and take the integral

$$\int_\Omega w\, \frac{\partial u}{\partial t}\, dx + \int_\Omega w\; \nabla.\vec{F}\; dx = \int_\Omega w\; Q\; dx.$$

Then integrate by part using formula (2.28) for the second term on the left-hand side to obtain

$$\int_\Omega w \frac{\partial u}{\partial t}\, dx - \int_\Omega \nabla w.\vec{F}\; dx + \int_\Gamma w\; \vec{F}.\vec{n}\; ds = \int_\Omega w\; Q\;\; dx.$$

Consider now a *control volume* consisting, for example, of an elementary triangle K_i in the two-dimensional case, such as those used in the finite element method. Take for w a function w_i whose value is one on the triangle and zero elsewhere. The second term in the above equation vanishes and the following relation results:

$$\int_{K_i} \frac{\partial u}{\partial t}\, dx + \int_{\Gamma_i} \vec{F}.\vec{n}\, ds = \int_{K_i} Q \; dx. \tag{2.38}$$

The above relation is at the basis of the finite volume approximation. To go a little further, the assumptions will be simplified slightly by taking a vector function $\vec{F}$ that is linear with respect to u. Specifically, assume

$$\vec{F} = \begin{pmatrix} \lambda_1 u \\ \lambda_2 u \end{pmatrix} \equiv \vec{\lambda} u.$$

Note that, in this case, the term $\nabla.\vec{F}$ in (2.37) becomes $\vec{F}(u) = \vec{\lambda}.\nabla u$. In addition, the right-hand side and the first term in the left-hand side of (2.38) can be approximated as follows:

$$\int_{K_i} \frac{\partial u}{\partial t}\, dx \approx \frac{\partial u_i}{\partial t}|K_i|, \quad \int_{K_i} Q\, dx \approx q_i|K_i|.$$

Here, $|K_i|$ represents the volume [1] of K_i, and q_i is some average value of Q in the cell K_i. These are crude approximations but they serve the purpose of illustrating the scheme.

The finite volume equation (2.38) yields

$$\frac{\partial u_i}{\partial t}|K_i| + \vec{\lambda}. \int_{\Gamma_i} u\, \vec{n}\, ds = q_i|K_i|. \tag{2.39}$$

The contour integral

$$\int_{\Gamma_i} u\, \vec{n}\, ds$$

is the sum of the integrals over all edges of the control volume. Let the value of u on each edge j be approximated by some "average" $\bar{u}_j$. In addition, s_j denotes the length of each edge and a common notation is

$$\vec{s}_j = s_j \vec{n}_j.$$

Then the contour integral is approximated by

$$\vec{\lambda}. \int_{\Gamma_i} u\, \vec{n}\, ds \approx \sum_{edges} \bar{u}_j \vec{\lambda}.\vec{n}_j s_j = \sum_{edges} \bar{u}_j \vec{\lambda}.\vec{s}_j. \tag{2.40}$$

The situation in the case where the control volume is a simple triangle is depicted in Figure 2.12. The unknowns are the approximations u_i of the function u associated with each cell. These can be viewed as approximations of u at the centers of gravity of each cell i. This type of model is called *cell-centered* finite volume approximations. Other techniques based on using approximations on the vertices of the cells are known as *cell-vertex* finite volume techniques.

[1] In two dimensions, "volume" is considered to mean area.

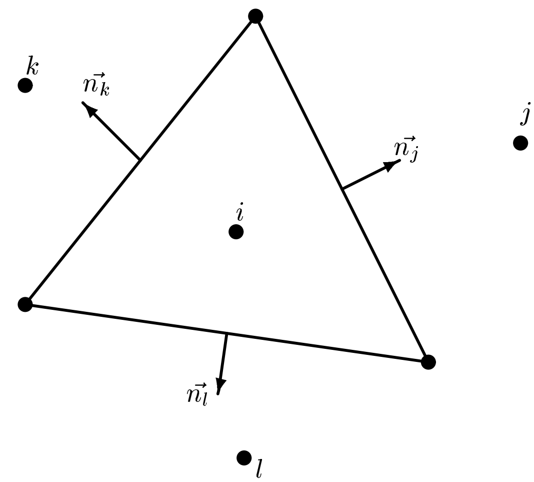

Figure 2.12 *Finite volume cell associated with node i and three neighboring cells.*

The value $\bar{u}_j$ required in (2.40) can be taken simply as the average between the approximation u_i of u in cell i and the approximation u_j in the cell j on the other side of the edge

$$\bar{u}_j = \frac{1}{2}(u_j + u_i). \tag{2.41}$$

This gives

$$\frac{\partial u_i}{\partial t}|K_i| \; + \frac{1}{2}\sum_j (u_i + u_j)\vec{\lambda}.\vec{s}_j = q_i|K_i|.$$

One further simplification takes place by observing that

$$\sum_j \vec{s}_j = 0$$

and therefore

$$\sum_j \; u_i\vec{\lambda}.\vec{s}_j = u_i\vec{\lambda}. \; \sum_j \vec{s}_j = 0.$$

This yields

$$\frac{\partial u_i}{\partial t} \; |K_i| + \frac{1}{2}\sum_j u_j\vec{\lambda}.\vec{s}_j = q_i|K_i|.$$

In the above equation, the summation is over all the neighboring cells j. One problem with such simple approximations is that they do not account for large gradients of u in the components. In finite volume approximations, it is typical to exploit upwind schemes which are more suitable in such cases. By comparing with one-dimensional upwind schemes, it can be easily seen that the suitable modification to (2.41) is as follows:

$$\bar{u}_j = \frac{1}{2}(u_j + u_i) - \frac{1}{2}\ \text{sign}\left(\vec{\lambda}.\vec{s}_j\right)\ (u_j - u_i). \tag{2.42}$$

This gives

$$\frac{\partial u_i}{\partial t}\ |K_i| + \sum_j \vec{\lambda}.\vec{s}_j\ \left(\frac{1}{2}(u_j + u_i) - \frac{1}{2}\ \text{sign}(\vec{\lambda}.\vec{s}_j)(u_j - u_i)\right) = q_i|K_i|.$$

Now write

$$\frac{\partial u_i}{\partial t}\ |K_i| + \sum_j \left(\frac{1}{2}(u_j + u_i)\vec{\lambda}.\vec{s}_j - \frac{1}{2}|\vec{\lambda}.\vec{s}_j|(u_j - u_i)\right) = q_i|K_i|$$

$$\frac{\partial u_i}{\partial t}\ |K_i| + \sum_j \left(u_i(\vec{\lambda}.\vec{s}_j)^+ + u_j(\vec{\lambda}.\vec{s}_j)^-\right) = q_i|K_i|$$

where

$$(z)^{\pm} \equiv \frac{z\ \pm\ |z|}{2}.$$

The equation for cell i takes the form

$$\frac{\partial u_i}{\partial t}\ |K_i| + \beta_i u_i + \sum_j \alpha_{ij} u_j = q_i|K_i|,$$

where

$$\beta_i = \sum_j (\vec{\lambda}.\vec{s}_j)^+\ \geq 0, \tag{2.43}$$

$$\alpha_{ij} = (\vec{\lambda}.\vec{s}_j)^-\ \leq 0. \tag{2.44}$$

Thus, the diagonal elements of the matrix are nonnegative, while its offdiagonal elements are nonpositive. In addition, the row-sum of the elements, i.e., the sum of all elements in the same row, is equal to zero. This is because

$$\beta_i + \sum_j \alpha_{ij} = \sum_j (\vec{\lambda}.\vec{s}_j)^+ + \sum_j (\vec{\lambda}.\vec{s}_j)^- = \sum_j \vec{\lambda}.\vec{s}_j = \vec{\lambda}.\sum_j \vec{s}_j = 0.$$

The matrices obtained have the same desirable property of weak diagonal dominance seen in the one-dimensional case. A disadvantage of upwind schemes, whether in the context of irregular grids or in one-dimensional equations, is the loss of accuracy due to the low order of the schemes.

EXERCISES

1. Derive Forward Difference formulas similar to (2.8), i.e., involving $u(x), u(x+h), u(x+2h), \ldots$, which are of second and third order. Write down the discretization errors explicitly.

2. Derive a Centered Difference formula for the first derivative, similar to (2.13), which is at least of third order.

3. Show that the Upwind Difference scheme described in 2.2.4, when a and $\vec{b}$ are constant, is stable for the model problem (2.7).

4. Develop the two nine-point formulas illustrated in Figure 2.4. Find the corresponding discretization errors. [Hint: Combine $\frac{1}{3}$ of the five-point formula (2.17) plus $\frac{2}{3}$ of the same formula based on the diagonal stencil $\{(x,y),(x+h,y+h)+(x+h,y-h),(x-h,y+h),(x-h,y-h)\}$ to get one formula. Use the reverse combination $\frac{2}{3}$, $\frac{1}{3}$ to get the other formula.]

5. Consider a (two-dimensional) rectangular mesh which is discretized as in the finite difference approximation. Show that the finite volume approximation to $\vec{\lambda}.\nabla u$ yields the same matrix as an upwind scheme applied to the same problem. What would be the mesh of the equivalent upwind finite difference approximation?

6. Show that the right-hand side of equation (2.16) can also be written as

$$\frac{1}{h^2}\delta^-\left(a_{i+\frac{1}{2}}\;\delta^+ u\right).$$

7. Show that the formula (2.16) is indeed second order accurate for functions that are in C^4.

8. Show that the functions ϕ_i's defined by (2.31) form a basis of V_h.

9. Develop the equivalent of Green's formula for the elliptic operator L defined in (2.6).

10. Write a short FORTRAN or C program to perform a matrix-by-vector product when the matrix is stored in unassembled form.

11. Consider the finite element mesh of Example 2.1. Compare the number of operations required to perform a matrix-by-vector product when the matrix is in assembled and in unassembled form. Compare also the storage required in each case. For a general finite element matrix, what can the ratio be between the two in the worst case (consider only linear approximations on triangular elements) for arithmetic? Express the number of operations in terms of the number of nodes and edges of the mesh. You may make the assumption that the maximum number of elements that are adjacent to a given node is p (e.g., $p = 8$).

12. Let K be a polygon in $\mathbb{R}^2$ with m edges, and let $\vec{s}_j = s_j\vec{n}_j$, for $j = 1,\dots,m$, where s_j is the length of the j-th edge and $\vec{n}_j$ is the unit outward normal at the j-th edge. Use the divergence theorem to prove that $\sum_{j=1}^m \vec{s}_j = 0$.

NOTES AND REFERENCES. The material in this chapter is based on several sources. For a basic description of the finite element method, the book by C. Johnson is a good reference [128]. Axelsson and Barker [16] gives a treatment which includes various solution techniques emphasizing iterative techniques. For finite difference and finite volume methods, we recommend C. Hirsch [121], which also gives a good description of the equations and solution methods for fluid flow problems. ■