5. THE FINITE DIFFERENCE METHOD

5.1 INTRODUCTION

Historically, the finite difference method was one of the first numerical procedures used to obtain approximate solutions to the partial differential equations of interest to engineers. The method is relatively easy to describe, and beginning students can readily program the algorithm for elementary problems. Nevertheless, in many engineering applications, especially solid mechanics, the finite element method has become the method of choice rather than the finite difference method. There are several reasons for this development of which the most important is the ease with which the finite element method can be used in comparison with the finite difference method for problems of geometrical complexity in two and three dimensions. Also, the existence of nonlinear material properties can be easily accommodated with the finite element method. It is for these reasons that the emphasis in this book is on the foundations of the finite element method even though the differences between the two approaches are not that profound in one dimension.

Nevertheless, basic concepts associated with the finite difference method are important and of value to those whose primary interest is in the use of finite elements. For example, it is not unusual to see a time integrator based on a finite difference algorithm used in conjunction with a spatial discretization based on finite elements. Among the important concepts associated with a finite difference algorithm are those of error and convergence that are used as the starting point for this chapter. To show convergence, an expression for the local truncation error must be obtained and consistency shown in addition to stability. The approach adopted here is somewhat unusual in that these terms are defined first and, then, the requirement of consistency is used to derive finite difference algorithms rather than the other way around of showing consistency after an algorithm has been defined. Most of the chapter is devoted to basic algorithms for some of the model problems discussed previously and to the analysis required to obtain the rate of convergence. Issues concerning symmetry of the governing matrix, variable coefficients and discontinuous coefficients are addressed. Algorithms are given for both Dirichlet and Neumann boundary conditions.

Consider a differential equation defined over the domain $0 < x < 1$ with boundary conditions prescribed at $x = 0$ and $x = 1$. On the domain, define an ordered set of n nodes including the boundary points: $x_1 = 0$ and $x_n = 1$ so that $x_1 < x_2 < x_3 < \cdots < x_{n-1} < x_n$. Let h_i denote the spacing between the nodes with the definition

$$h_i = x_{i+1} - x_i \qquad i = 1, \cdots, n \tag{5.1-1}$$

This procedure is called **spatial discretization** and the resulting mesh is shown in Fig. 5.2-1. For the sketch shown, x_1 and x_n are **boundary nodes**; the remainder are **interior nodes**. If

$$h = \frac{1}{n-1} \tag{5.1-2}$$

and $h_i = h$ for all i, then the mesh is said to be uniform.

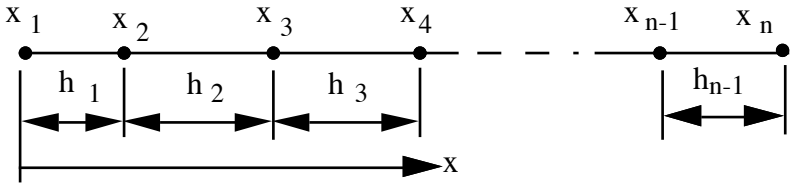

Fig. 5.2-1. One-dimensional mesh showing nodes and spacing.

Consider one-dimensional model problems of the form:

$$D(\phi) + f(x) = 0 \qquad 0 < x < 1 \tag{5.1-3}$$

in which $D(\ )$ is a **linear differential operator** with the boundary conditions unspecified for the moment. Let the analytical or "exact" solution be ϕ^e which is defined for all x in [0, 1]. The smoothness of the solution depends on f(x) and the coefficient functions which appear in $D(\ )$. For now assume ϕ^e is of class C^∞. The exact solution evaluated at each node is

$$\phi_i^e = \phi^e(x_i) \quad i = 1, \cdots, n \tag{5.1-4}$$

These values are often placed in a vector, $\{\phi^e\}$, for comparison with approximate solutions.

A finite difference algorithm provides a numerical, or approximate, solution, ϕ_i, at each node and the solution for all nodes is combined in the vector $\{\phi\}$. To obtain a solution, n equations must be provided, one for each node. Typically, the structure of the equation for each interior node is similar to that for any other interior node. Each equation normally involves the dependent variable for that node and the dependent variables for surrounding nodes. The specific arrangement for a given algorithm is called a **stencil** and is reflected by the nonzero coefficients, A_{ij}, in the equation for node i:

$$\sum_{j=1}^{n} A_{ij}\phi_j = f_i \qquad i = 1,...,n \tag{5.1-5}$$

where $f_i = f(x_i)$. This equation is also called the **approximating equation** to the differential equation.

Since the differential operator considered here is linear, the class of discretizations will be restricted to those that yield linear algebraic equations. There are examples of discretizations which result in nonlinear algebraic equations. This will occur, for example, when A_{ij} depends on $\{\phi\}$.

One example of a linear discretization is a **three-point stencil** in which case a typical equation for node i may be of the form

$$A_{i,i-1}\phi_{i-1} + A_{i,i}\phi_i + A_{i,i+1}\phi_{i+1} = f_i \tag{5.1-6}$$

with $A_{ij} = 0$ for $j \neq i-1, i$ or $i+1$. In general, specific values for A_{ij} depend on the particular finite difference algorithm and differential equation, of which examples will be given later, and will involve the mesh spacing and model parameters. If f_i is a component of the vector $\{f\}$, and if special boundary stencils are introduced for the first and last nodes, the result is a matrix equation

$$[A]\{\phi\} = \{f\} \tag{5.1-7}$$

from which $\{\phi\}$ is obtained.

In the finite difference literature, it is more common to introduce a vector $\{\phi^I\}$ with components consisting only of the desired function ϕ_i at the interior points i = 2, ..., n - 1. Boundary values appear through appropriate modifications to the stencil for interior points close to the boundary and to the forcing vector which becomes $\{f^I\}$. The resulting matrix $[A^I]$ is n - 2 x n - 2 instead of n x n, the size of [A], and the matrix equation becomes

$$[A^I]\{\phi^I\} = \{f^I\} \tag{5.1-8}$$

For an easier comparison with the later development of the finite element method, the form of (5.1-7) will be used. However, on occasion, the notation of (5.1-8) will be used whenever it is more convenient to do so.

5.2 STABILITY

The data for the problem associated with the forcing function are contained in {f}. Recall that the solution is said to be stable if small changes in the data provide small changes in the solution. The same idea carries over to numerical solutions. If the algorithm is **stable**, small changes in {f} should provide small changes in $\{\phi\}$. To show that the requirement of stability places a restriction on [A], let $\{f\}_1$ and $\{f\}_2$ be two force vectors. From (5.1-7), the approximate solutions $\{\phi\}_1$ and $\{\phi\}_2$ satisfy

$$[A]\{\phi\}_1 = \{f\}_1 , \qquad [A]\{\phi\}_2 = \{f\}_2 \tag{5.2-1}$$

Then the difference must satisfy

$$[A]\{\{\phi\}_1 - \{\phi\}_2\} = \{\{f\}_1 - \{f\}_2\} \tag{5.2-2}$$

or

$$\{\{\phi\}_1 - \{\phi\}_2\} = [A]^{-1}\{\{f\}_1 - \{f\}_2\} \tag{5.2-3}$$

If property (e) for a compatible matrix norm is used (Sect. 4.4), then

$$\left\|\{\{\phi\}_1 - \{\phi\}_2\}\right\| \leq \left\|[A]^{-1}\right\| \left\|\{\{f\}_1 - \{f\}_2\}\right\| \tag{5.2-4}$$

If $\left\|\{\{f\}_1 - \{f\}_2\}\right\|$ is small, it follows that $\left\|\{\{\phi\}_1 - \{\phi\}_2\}\right\|$ is also small, i.e., stability holds provided $\left\|[A]^{-1}\right\|$ is finite. For example, if the two norm is used, the smallest eigenvalue

of $[A]^T[A]$ must be greater than zero, i.e., for stability [A] cannot be singular. For numerical evaluations, the problem is said to be stable if the lowest eigenvalue is not close to zero.

A related issue is that of the condition number of [A], which provides a measure of the ratio of relative changes in {f} to relative changes in {ϕ}. For the two norm, the condition number is the ratio of the largest to smallest eigenvalue of [A] if [A] is symmetric, positive definite (Sect. 4.6).

5.3 ERROR

Suppose e(x) is a given function of x defined for $0 \le x \le 1$. In analogy with the norm of a vector, the p-norm of the function, e(x), is defined to be

$$\|e(x)\|_p \equiv \left[\int_0^1 |e(x)|^p \, dx \right]^{\frac{1}{p}} \qquad p \ge 1 \tag{5.3-1}$$

We note that the properties associated with vector norms (Sect. 4.5) hold as well for this norm. The most typically used norms are the one-norm, the two-norm and the infinity norm:

$$\|e(x)\|_1 \equiv \left[\int_0^1 |e(x)| dx \right], \quad \|e(x)\|_2 \equiv \left[\int_0^1 |e(x)|^2 \, dx \right]^{\frac{1}{2}}, \quad \|e(x)\|_\infty \equiv \max_x |e(x)| \tag{5.3-2}$$

Now suppose e(x) is interpreted to be the error function between an exact solution, $\phi^e(x)$, and an approximate solution, $\phi(x)$, to a boundary value problem. Let ε_p be the p-norm of this error function:

$$e(x) = \phi^e(x) - \phi(x) , \qquad \varepsilon_p = \|e(x)\|_p \tag{5.3-3}$$

The p-norm is a convenient measure of error because it is a single positive parameter, which is zero only if the exact and approximate solutions agree almost everywhere.

As will be shown, the finite difference approach provides only approximate solutions at discrete points so it is necessary to consider a measure of error for the finite difference solution. Suppose the approximate solution at node i is ϕ_i. The error at node i is then

$$e_i = \phi_i^e - \phi_i \tag{5.3-4}$$

and the error vector is

$$\{e\} = \{\phi^e\} - \{\phi\} \tag{5.3-5}$$

Define a single scalar measure of the nodal error as a p-norm

$$e_p = \|\{e\}\|_p = \left[\sum_{i=1}^{n} |e_i|^p \right]^{1/p} \tag{5.3-6}$$

in which any value $p \geq 1$ can be used. Again, the most commonly used norms are p = 1, 2, and ∞.

To compare ε_p and e_p, suppose the approximate solution is obtained with the finite difference method. Since the approximate solution and, hence, the error function is known only at the nodes, select a numerical quadrature rule that approximates the integral (5.3-1) by using the nodal values of the error function (Appendix A.4). The Riemann sum and Newton-Cotes formulas fall in this category. For simplicity, consider the Riemann sum as an approximation for the integral required to obtain the norm:

$$\int_0^1 |e(x)|^p \, dx = \sum_{i=1}^{n-1} h_i |e_i|^p + O(\hat{h}^2) \tag{5.3-7}$$

in which $\hat{h}$ is the maximum value of the increments in grid-point spacing, h_i. In the limit of small $\hat{h}$, (5.3-7) becomes

$$\varepsilon_p = \left[\sum_{i=1}^{n-1} h_i |e_i|^p \right]^{1/p} \tag{5.3-8}$$

Since $h_i \leq \hat{h}$ for all i (by definition), it follows that

$$\left[\sum_{i=1}^{n-1} h_i |e_i|^p \right]^{1/p} \leq \hat{h}^{1/p} \left[\sum_{i=1}^{n-1} |e_i|^p \right]^{1/p} \leq \hat{h}^{1/p} \left[\sum_{i=1}^{n} |e_i|^p \right]^{1/p}$$

or

$$\varepsilon_p \leq \hat{h}^{1/p} \, e_p \tag{5.3-9}$$

in which the norm of the **nodal error vector** given in (5.3-6) has been used. Because of the factor involving $\hat{h}$, it is important to emphasize that the norm of the error function, ε_p, is not the same as the norm of the error vector of nodal values, e_p,

If the spacing is uniform, then $h_i = h$ for all i and (5.3-9) yields the inequality

$$\varepsilon_p \leq h^{1/p} \, e_p \tag{5.3-10}$$

For large n, $h \approx 1/n$. It follows that (5.3-10) becomes

$$\varepsilon_p \leq e_{sp} \qquad e_{sp} = \|\{e\}\|_{sp} \equiv \frac{1}{n^{1/p}} \|\{e\}\|_p = \frac{1}{n^{1/p}} e_p \tag{5.3-11}$$

where e_{sp} is the scaled vector norm introduced in Sect. 4.5. Because the scaled norm automatically takes the term $h^{1/p}$ into account, the scaled norm will be used extensively in the remainder of this chapter.

Another argument for using the scaled norm is suggested by the situation where all components of the error vector are the same constant. The use of the p-norm yields a norm of the error vector that increases with n, a result that might be interpreted to mean that error increased with mesh refinement. The use of the scaled norm removes this anomalous feature.

To show convergence, it is necessary to show that the error, ε_p, goes to zero as h $\rightarrow 0$. This is usually most easily accomplished by showing **stability** and **consistency** of the algorithm. A demonstration of consistency requires first determining the **local truncation error** which is obtained from the residual when the exact solution is substituted into the approximating equation. The result of spatial discretization, as indicated in (5.1-5), is the equation

$$\sum_{j=1}^{n} A_{ij}\phi_j = f_i \qquad i = 1,...,n \tag{5.3-12}$$

The local truncation error for node i, which we denote as τ_i, is

$$\tau_i = \sum_{j=1}^{n} A_{ij}\phi_j^e - f_i \tag{5.3-13}$$

The approximating equation is said to be **consistent** with the differential equation if $\tau_i \rightarrow 0$ as $\hat{h} \rightarrow 0$. The vector with components consisting of the local truncation error for all nodes is denoted by $\{\tau\}$. In order to relate the error vector to the local truncation vector, we rewrite (5.3-12) and (5.3-13) in matrix forms:

$$\left.\begin{aligned} \{0\} &= [A]\{\phi\} - \{f\} \\ \{\tau\} &= [A]\{\phi^e\} - \{f\} \end{aligned}\right\} \tag{5.3-14}$$

The result of subtracting corresponding terms is

$$\{\tau\} = [A]\{e\} \tag{5.3-15}$$

with the error vector denoting the difference between the exact and approximate solutions given in (5.3-5), i.e., $\{e\} = \{\phi^e\} - \{\phi\}$. The solution to (5.3-15) is

$$\{e\} = [A]^{-1}\{\tau\} \tag{5.3-16}$$

If a compatible matrix norm is used, the theory of Sect. 4.4 yields

$$\|\{e\}\|_{sp} \le \|[A]^{-1}\|_p \|\{\tau\}\|_{sp} \qquad \text{or} \qquad e_{sp} \le \|[A]^{-1}\|_p \tau_{sp} \tag{5.3-17}$$

in which τ_{sp} denotes the scaled p-norm of $\{\tau\}$. If the procedure is stable, then the p-norm of $[A]^{-1}$ is bounded above by a finite number, A_p, or

$$\|[A]^{-1}\|_p \le A_p < \infty \tag{5.3-18}$$

As an example, suppose a 2-norm is used and [A] is symmetric, positive definite. This is a situation that holds for a large number of problems. Then the 2-norm is the maximum eigenvalue of $[A]^{-1}$, which is the inverse of the minimum eigenvalue of [A]:

$$\|[A]^{-1}\|_2 = \lambda_{max}\left([A]^{-1}\right) = \frac{1}{\lambda_{min}([A])} \equiv \frac{1}{\lambda_1} \tag{5.3-19}$$

and the stability condition is simply $\lambda_1 > 0$. Here, it is assumed that λ_1 is independent of the mesh spacing.

For a uniform mesh as h goes to zero, each component of $\{\tau\}$, and hence the vector, will generally be of the form

$$\tau_i = c_i^\tau h^r , \qquad \{\tau\} = \{c^\tau\} h^r \tag{5.3-20}$$

in which each component, c_i^τ, depends on the data for the problem but is independent of h. If the exponent of h satisfies the inequality $r > 0$, the method is **consistent**. In this context, r is called the **order of the consistency error** or the **order of the local truncation error** or just the **order of the method**. The scaled p-norm of $\{\tau\}$ is

$$\|\{\tau\}\|_{sp} = \frac{1}{n^{1/p}} \left[\sum_{i=1}^{n} |\tau_i|^p \right]^{1/p} = h^r \|\{c^\tau\}\|_{sp} \tag{5.3-21}$$

Suppose that the boundary condition at is satisfied exactly so that $c_n^\tau = 0$. Suppose further that c_i^τ is bounded by C^τ, i.e., $0 \le |c_i^\tau| \le C^\tau$ for all nodes. Then,

$$\|\{\tau\}\|_{sp} \le \frac{h^r}{(n-1)^{1/p}} \left[\sum_{i=1}^{n-1} (C^\tau)^p \right]^{1/p} = C^\tau h^r \tag{5.3-22}$$

i.e., the use of the scaled norm eliminates the factor $(n-1)^{1/p} = h^{-1/p}$. With the use of the inequality of (5.3-17), the stability condition of (5.3-18) and the consistency condition of (5.3-20) as reflected in (5.3-22), it follows that

$$e_{sp} \le C^e h^r , \qquad C^e = A_p C^\tau \tag{5.3-23}$$

If A_p is independent of h, a property which usually holds but is not always easy to show, then C^e is a constant that depends on the data but is independent of h. With the use of (5.3-11), the norm of the error function, which is the measure of primary interest, also satisfies the inequality

$$\varepsilon_p \le C^e h^r \tag{5.3-24}$$

a result which does not depend on the particular p-norm used.

If the error approaches zero as spacing between nodes is reduced, i.e., if $\varepsilon_p \to 0$ as $h \to 0$, then the approximate solution is said to **converge** to the exact solution with mesh refinement. This property is required to hold for all proper algorithms. From (5.3-24), the algorithm will be convergent if $r > 0$, and then r is also called the **rate of convergence**. The simplest acceptable algorithms typically result in $r = 1$; algorithms with higher rates of convergence are usually more complex and require more algebra to derive. For a given number of nodes, and if the solution is sufficiently smooth, numerical solutions from a higher-order algorithm are normally more accurate than results obtained from a low-order algorithm. However, this conclusion may not hold for any one given value of h because there is the possibility that the coefficient C^e may be large for the higher-order algorithm.

In effect, we have proven **Lax's Equivalence Theorem**: stability and consistency imply convergence. Convergence with mesh refinement is the property that is desired. Therefore, stability and consistency are the properties that must be established for an algorithm.

Because the various norms and measures of error may seem confusing, we provide the following summary:

$$\begin{aligned} \tau_i &= c_i^{\tau} h^r , & \|\{\tau\}\|_{sp} &\le C^{\tau} h^r \\ e_{sp} &= \|\{e\}\|_{sp} \le C^e h^r , & \varepsilon_p &= \|e(x)\|_p \le C^e h^r \end{aligned} \tag{5.3-25}$$

in which τ_i denotes the local truncation error at a node. The scaled p-norm of the vector formed from these nodal values is $\|\{\tau\}\|_{sp}$. The scaled p-norm of the vector of nodal errors is e_{sp} and the corresponding p-norm of the error function is ε_p. The conventional vector norms, τ_p and e_p are generally not used because the exponents of the mesh spacing, h, depend on the particular norm used (a term 1/p) whereas the exponent of h does not depend on p for the scaled p-norm and for ε_p. Note that the exponent, r, of h that appears in the typical component of the local truncation error is identical to the exponent for the norm of the error function, an observation that is extensively used for developing consistent finite difference algorithms.

In summary, most algorithms are evaluated through the **local truncation error** which, in turn, is used with **stability** to prove **convergence**. Since **consistency** ($r > 0$) and the **rate of convergence**, r, are features that must be determined anyway, the next section shows how elementary finite difference algorithms are obtained directly from expressions for the local truncation error, a procedure that involves extensive use of a Taylor-series expansion.

5.4 FINITE DIFFERENCE ALGORITHMS FOR ODE'S WITH CONSTANT COEFFICIENTS

5.4.1 A First-Order Differential Equation

Preliminary Comments

The primary task of this subsection is to illustrate the basic procedure for developing finite difference algorithms in the simplest possible context; namely a first-order differential equation. The procedure is to propose a stencil, and then use a Taylor expansion in the expression for the residual to ensure that consistency is met. Then we show that by simply adjusting the force vector, higher-order algorithms can be achieved. Finally we introduce symbolic operators to represent common finite difference algorithms for first derivatives.

Creating a Finite Difference Algorithm

To illustrate the procedure for obtaining a finite difference algorithm that meets the consistency requirement, we begin with a first-order differential equation with one boundary condition:

$$\phi_{,x} = g(x) , \qquad 0 < x \le 1, \qquad \phi(0) = \phi_0 \tag{5.4-1}$$

This might be an equation that governs the distribution of flux when one flux boundary condition is given. We seek an algorithm that will provide a linear algebraic set of equations, $[A]\{\phi\} = \{b\}$, allowing us to obtain nodal variables ϕ_i denoting the

approximation of $\phi(x)$ at x_i. With the mesh of Fig. 5.2-1, we set $\phi_1 = \phi_0$, and propose an equation for the i'th node that involves the unknown nodal values at node i, and node i - 1:

$$\alpha_{i-1}\phi_{i-1} + \alpha_i\phi_i = b_i \qquad \text{for} \qquad i \geq 2 \tag{5.4-2}$$

This form is called a **two-point stencil**. The stencil indicates that the entries in row i of the coefficient matrix [A] are zero except for $A_{i,i-1} = \alpha_{I-1}$ and $A_{i,i} = \alpha_i$. The parameters α_{i-1}, α_i and b_i are determined by enforcing consistency. To this end we derive the local truncation error, which is the residual when the exact solution, ϕ^e, is substituted in (5.4-2):

$$\tau_i = \alpha_{i-1}\phi^e_{i-1} + \alpha_i\phi^e_i - b_i \tag{5.4-3}$$

With the notation

$$\phi^e_{i,x} = \phi^e{}_{,x}(x_i)\,, \qquad \phi^e_{i,xx} = \phi^e{}_{,xx}(x_i)\,, \qquad \text{etc.} \tag{5.4-4}$$

the use of the Taylor expansion about x_i for the first term in (5.4-3) yields

$$\tau_i = \alpha_{i-1}(\phi^e_i - h_{i-1}\phi^e_{i,x} + \frac{h^2_{i-1}}{2}\phi^e_{i,xx} - \cdots) + \alpha_i\phi^e_i - b_i \tag{5.4-5}$$

with higher order terms represented symbolically as $\cdots$. The presence of the derivatives is based on the assumption that they exist, i.e., the solution ϕ^e is assumed to be sufficiently smooth. Recall that the exact solution satisfies the differential equation at each node, i.e.,

$$\phi^e_{i,x} = g_i \tag{5.4-6}$$

The result of substituting (5.4-6) and (5.4-5) is

$$\tau_i = \phi^e_i(\alpha_{i-1} + \alpha_i) - \alpha_{i-1}h_{i-1}g_i + \alpha_{i-1}\frac{h^2_{i-1}}{2}\phi^e_{i,xx} - b_i + \cdots \tag{5.4-7}$$

The zero'th and first order terms in h_{i-1} are set to zero as follows:

$$\alpha_{i-1} + \alpha_i = 0, \qquad b_i = -\alpha_{i-1}h_{i-1}g_i \tag{5.4-8}$$

In order for the local truncation error to be of the same order as the order of the solution error, it is necessary that b_i be proportional to the forcing term, g_i. If the constant of proportionality is chosen to be unity, then

$$\alpha_{i-1} = -\alpha_i = \frac{1}{h_{i-1}}\,, \qquad b_i = g_i \tag{5.4-9}$$

and from (5.4-7), it follows that in the limit as the maximum value of the mesh spacing goes to zero ($\hat{h} \to 0$),

$$\tau_i = c^\tau_i h_{i-1}\,, \qquad c^\tau_i = -\frac{1}{2}\phi^e_{i,xx} = -\frac{1}{2}g_{i,x} \tag{5.4-10}$$

where the coefficient c^τ_i depends only on the data for the problem and

$$\tau_i = O(\hat{h}) \tag{5.4-11}$$

Therefore, the algorithm is consistent with rate of convergence one (or first-order accurate).

With the boundary condition included at the first node, the form of the linear algebraic equation, $[A]\{\phi\} = \{b\}$, becomes

$$\begin{bmatrix} 1 & 0 & 0 & \cdots \\ -\frac{1}{h_1} & \frac{1}{h_1} & 0 & \cdots \\ 0 & -\frac{1}{h_2} & \frac{1}{h_2} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{Bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \vdots \end{Bmatrix} = \begin{Bmatrix} \phi_0 \\ g_2 \\ g_3 \\ \vdots \end{Bmatrix} \tag{5.4-12}$$

Here, the equations can be solved sequentially without the use of a matrix solver, but this is merely a consequence of the choice of problem and algorithm.

Altering the Force Vector

Next, we show that the rate of convergence can be increased simply by altering the term, b_I, to include contributions from the forcing function at the two nodes used in the stencil. As an example, suppose the second part of (5.4-9) is replaced with

$$b_i = (1-\eta_i)g_{i-1} + \eta_i g_i \qquad \text{for} \qquad i \geq 2 \tag{5.4-13}$$

with η_i as an additional parameter. With the use of a Taylor expansion and the differential equation we obtain

$$\begin{aligned} b_i &= (1-\eta_i)(g_i - h_{i-1}g_{i,x} + \frac{h_i^2}{2}g_{i,xx} + \cdots) + \eta_i g_i \\ &= g_i - (1-\eta_i)(h_{i-1}\phi^e_{i,xx} - \frac{h_i^2}{2}\phi^e_{i,xxx} + \cdots) \end{aligned} \tag{5.4-14}$$

The local truncation error of (5.4-7) becomes

$$\begin{aligned} \tau_i = {} & \phi^e_i(\alpha_{i-1} + \alpha_i) - \alpha_{i-1}h_{i-1}g_i + \alpha_{i-1}\frac{h_{i-1}^2}{2}\phi^e_{i,xx} - \alpha_{i-1}\frac{h_{i-1}^3}{6}\phi^e_{i,xxx} + \cdots \\ & - g_i + (1-\eta_i)(h_{i-1}\phi^e_{i,xx} - \frac{h_i^2}{2}\phi^e_{i,xxx} + \cdots) \end{aligned} \tag{5.4-15}$$

With the use of (5.4-9), the algorithm remains first-order accurate for any choice of η_i. However, if we set

$$\eta_i = 1/2 \tag{5.4-16}$$

then the local truncation error reduces to

$$\begin{aligned} \tau_i &= -\alpha_{i-1}\frac{h_{i-1}^3}{6}\phi^e_{i,xxx} - (1-\eta_i)\frac{h_{i-1}^2}{2}\phi^e_{i,xxx} + \cdots \\ &= -\frac{h_i^2}{12}\phi^e_{i,xxx} + \cdots \end{aligned} \tag{5.4-17}$$

and the algorithm is second order.

Alternative Finite Difference Algorithms for First Derivatives

In summary, for an interior node, the two-point stencil of (5.4-2) yields an algorithm, called a **backward difference**, that is first-order accurate and is given as follows:

$$\frac{1}{h_{i-1}}(\phi_i - \phi_{i-1}) = g_i \tag{5.4-18}$$

An alternative way of viewing the process for developing suitable finite difference algorithms is to let $\delta\phi_i/\delta x$ denote the algorithm used for approximating the derivative $d\phi/dx$ at the point x_i. For example, the algorithm given in (5.4-18) is represented as follows:

$$\frac{d\phi_i}{dx} \rightarrow \left[\frac{\delta\phi_i}{\delta x}\right]_{bd1} = \frac{1}{h_{i-1}}(\phi_i - \phi_{i-1}) \tag{5.4-19}$$

The subscript "bd1" indicates that the formula is a backward difference algorithm that is first-order accurate. The local truncation error can be interpreted as the difference of the two terms

$$\tau_i = \left[\frac{\delta\phi_i^e}{\delta x}\right]_{bdl} - \frac{d\phi_i^e}{dx} = \frac{1}{2}\phi_{i,xx}^e h_{i-1} \tag{5.4-20}$$

The approach can be extended to the differential operator associated with a given differential equation. The result is the same as the approach outlined above involving the differential equation itself so the choice of how one interprets the local truncation error is a matter of taste.

Now consider a three-point stencil involving the two adjacent points to the left of the point of interest:

$$\alpha_{i-2}\phi_{i-2} + \alpha_{i-1}\phi_{i-1} + \alpha_i\phi_i = b_i \qquad \text{for} \qquad i \geq 3 \tag{5.4-21}$$

As before, choose $b_i = g_i$. An analysis similar to that performed for the two-point stencil leads to

$$\tau_i = c_0\phi_i^e + c_1\phi_{i,x}^e + c_2\phi_{i,xx}^e + c_3\phi_{i,xxx}^e + \cdots \tag{5.4-22}$$

where

$$\left.\begin{aligned} c_0 &= \alpha_{i-2} + \alpha_{i-1} + \alpha_i \\ c_1 &= -(h_{i-2} + h_{i-1})\alpha_{i-2} - h_{i-1}\alpha_{i-1} - 1 \\ c_2 &= \tfrac{1}{2}(h_{i-2} + h_{i-1})^2\alpha_{i-2} + \tfrac{1}{2}h_{i-1}^2\alpha_{i-1} \\ c_3 &= -\tfrac{1}{6}(h_{i-2} + h_{i-1})^3\alpha_{i-2} - \tfrac{1}{6}h_{i-1}^3\alpha_{i-1} \end{aligned}\right\} \tag{5.4-23}$$

The result of setting the first three terms to zero is

$$\alpha_{i-2} = \frac{h_{i-1}^2}{D}\ , \qquad D = h_{i-2}h_{i-1}(h_{i-2} + h_{i-1})$$
$$\alpha_{i-1} = -\frac{(h_{i-2} + h_{i-1})^2}{D}\ , \qquad \alpha_i = \frac{h_{i-2}(h_{i-2} + 2h_{i-1})}{D} \tag{5.4-24}$$

with a local truncation error of

$$\tau_i = c_3\phi_{i,xxx} = \tfrac{1}{6}\phi_{i,xxx}\, h_{i-1}(h_{i-2} + h_{i-1}) \tag{5.4-25}$$

In summary, the three-point stencil results in a backward-difference algorithm that is second-order accurate:

$$\left[\frac{\delta\phi_i}{\delta x}\right]_{bd2} \equiv \alpha_{i-2}\phi_{i-2} + \alpha_{i-1}\phi_{i-1} + \alpha_i\phi_i\ , \qquad \left[\frac{\delta\phi_i}{\delta x}\right]_{bd2} = g_i \tag{5.4-26}$$

For a uniform mesh with $h_i = h$ for all i, this operator reduces to

$$\left[\frac{\delta\phi_i}{\delta x}\right]_{bd2} = \frac{1}{2h}[\phi_{i-2} - 4\phi_{i-1} + 3\phi_i] \qquad \text{(uniform mesh)} \tag{5.4-27}$$

A similar approach for two- and three-point stencils involving one and two points to the right of the node of interest, respectively, result in forward-difference algorithms that are first-order and second-order accurate:

$$\left[\frac{\delta\phi_i}{\delta x}\right]_{fd1} = \frac{1}{h_i}(\phi_{i+1} - \phi_i)$$
$$\left[\frac{\delta\phi_i}{\delta x}\right]_{fd2} = \frac{1}{D}\left[-h_{i+1}(2h_i + h_{i+1})\phi_i + (h_i + h_{i+1})^2\phi_{i+1} - h_i^2\phi_{i+2})\right] \tag{5.4-28}$$
$$D = h_i h_{i+1}(h_i + h_{i+1})$$

For a uniform mesh, these formulas reduce to

$$\left.\begin{aligned}\left[\frac{\delta\phi_i}{\delta x}\right]_{fd1} &= \frac{1}{h}(\phi_{i+1} - \phi_i) \\ \left[\frac{\delta\phi_i}{\delta x}\right]_{fd2} &= \frac{1}{2h}\left[-3\phi_i + 4\phi_{i+1} - \phi_{i+2})\right]\end{aligned}\right\} \quad \text{(uniform mesh)} \tag{5.4-29}$$

Now consider a **three-point stencil** involving points on either side of the node of interest:

$$\alpha_{i-1}\phi_{i-1} + \alpha_i\phi_i + \alpha_{i+1}\phi_{i+1} = g_i \tag{5.4-30}$$

The local truncation error analysis results in expressions for the three coefficients in (5.4-24) and in the limit as the mesh spacing goes to zero, the local truncation error becomes

$$\tau_i = C_i^\tau h_{i-1}h_i\ , \qquad C_i^\tau = -\frac{1}{6}\phi_{i,xxx}^e \tag{5.4-31}$$

which indicates the algorithm is second-order accurate. The algorithm for the three-point stencil and a nonuniform mesh is

$$\left[\frac{\delta\phi_i}{\delta x}\right]_{cd2} \equiv \frac{1}{h_{i-1}h_i(h_i+h_{i-1})}[-h_i^2\phi_{i-1} + (h_i^2 - h_{i-1}^2)\phi_i + h_{i-1}^2\phi_{i+1}] = g_i \tag{5.4-32}$$

which reduces to the following for a uniform mesh:

$$\frac{1}{2h}[\phi_{i+1} - \phi_{i-1}] = g_i \tag{5.4-33}$$

The latter is often called the **central difference algorithm**.

Closing Comments

In this subsection, we have introduced a systematic scheme for deriving finite difference algorithms. For a representative node, a stencil is assumed involving the product of unknown parameters and the values of the required function at a set of adjacent nodes. Based on the convergence requirement that the algorithm must be consistent with the differential operator, a set of equations is obtained for determining the unknown coefficients and the forcing term. The order of the algorithm can be increased by using more nodal points in the stencil or by including the values of the forcing function at all nodes in the stencil within an expression for b_i. The latter is an exceedingly easy way to improve accuracy and, surprisingly, is not generally discussed in books on finite differences.

5.4.2 A Second-Order Differential Equation

A Finite Difference Algorithm

Now we consider the model problem

$$k\phi_{,xx} + f(x) = 0\,, \qquad 0 < x < 1 \qquad (k = \text{constant}) \tag{5.4-34}$$

for which an approximate solution is to be obtained using a three-point stencil of the form

$$\alpha_{i-1}\phi_{i-1} + \alpha_i\phi_i + \alpha_{i+1}\phi_{i+1} = b_i \tag{5.4-35}$$

with $f_i = f(x_i)$ and $A_{i,i-1} = \alpha_{I-1}$, etc. For this problem, a two-point stencil will not yield a consistent algorithm. As before, the parameters α_{i-1}, α_i and α_{i+1} are determined by considering the local truncation error which is the residual when the exact solution is substituted in (5.4-35):

$$\tau_i = \alpha_{i-1}\phi^e_{i-1} + \alpha_i\phi^e_i + \alpha_{i+1}\phi^e_{i+1} - b_i \tag{5.4-36}$$

For our initial algorithm choose

$$b_i = f_i = -k\phi^e_{i,xx} \tag{5.4-37}$$

Introduce the Taylor expansions

$$\begin{aligned}
\phi^e_{i-1} &= \phi^e_i - h_{i-1}\phi^e_{i,x} + \frac{h_{i-1}^2}{2}\phi^e_{i,xx} - \frac{h_{i-1}^3}{6}\phi^e_{i,xxx} + \frac{h_{i-1}^4}{24}\phi^e_{i,xxxx} - \cdots \\
\phi^e_{i+1} &= \phi^e_i + h_i\phi^e_{i,x} + \frac{h_i^2}{2}\phi^e_{i,xx} + \frac{h_i^3}{6}\phi^e_{i,xxx} + \frac{h_i^4}{24}\phi^e_{i,xxxx} + \cdots
\end{aligned} \tag{5.4-38}$$

with higher order terms represented symbolically as +···. Then

$$\tau_i = c_0\phi_i^e + c_1\phi_{i,x}^e + c_2\phi_{i,xx}^e + c_3\phi_{i,xxx}^e + c_4\phi_{i,xxxx}^e + \cdots \tag{5.4-39}$$

where

$$\left.\begin{aligned} c_0 &= \alpha_{i-1} + \alpha_i + \alpha_{i+1}\,, & c_1 &= -h_{i-1}\alpha_{i-1} + h_i\alpha_{i+1} \\ c_2 &= \tfrac{1}{2}(h_{i-1}^2\alpha_{i-1} + h_i^2\alpha_{i+1}) + k & & \\ c_3 &= -\tfrac{1}{6}(h_{i-1}^3\alpha_{i-1} - h_i^3\alpha_{i+1})\,, & c_4 &= \tfrac{1}{24}(h_{i-1}^4\alpha_{i-1} + h_i^4\alpha_{i+1}) \end{aligned}\right\} \tag{5.4-40}$$

Setting $c_0 = 0$, $c_1 = 0$ and $c_2 = 0$ results in three algebraic equations for determining α_i, β_i and γ_i and is the minimum requirement for ensuring that the local truncation error is proportional to h with a positive exponent. Then c_3 is used to determine the rate of convergence unless $c_3 = 0$, in which case c_4 determines the rate of convergence. If $c_4 = 0$, c_5 must be determined and so on. The consequence of the set of equations $c_0 = 0$, $c_1 = 0$ and $c_2 = 0$ is

$$\alpha_{i-1} = \frac{-2k}{h_{i-1}(h_{i-1} + h_i)}, \qquad \alpha_i = \frac{2k}{h_{i-1}h_i}\,, \qquad \alpha_{i+1} = \frac{-2k}{h_i(h_{i-1} + h_i)} \tag{5.4-41}$$

and then

$$c_3 = -\frac{k}{3}(h_i - h_{i-1})\,, \qquad c_4 = -\frac{k}{12}(h_{i-1}^2 - h_{i-1}h_i + h_i^2) \tag{5.4-42}$$

For unequal spacing of the nodes, $c_3 \neq 0$. For small $\hat{h}$, the maximum value of h_i, the lowest order terms dominate so that in the limit as $\hat{h} \to 0$

$$\tau_i = C_i^\tau(h_i - h_{i-1})\,, \qquad C_i^\tau = \tfrac{k}{3}\phi_{i,xxx}^e \tag{5.4-43}$$

It follows that

$$\tau_i = O(\hat{h}) \tag{5.4-44}$$

and the algorithm is consistent with rate of convergence one (or first-order accurate). However, for uniform spacing where $h = h_{i-1} = h_i$, then $c_3 = 0$, and as h approaches zero

$$\tau_i = C_i^\tau h^2\,, \qquad C_i^\tau = \frac{k}{12}\phi_{i,xxxx}^e \tag{5.4-45}$$

and the algorithm is second-order accurate. Since the exact solution depends on the data for the problem, so does the local truncation error.

In summary, for an interior node, the three-point stencil yields an algorithm for nonuniform spacing that is first-order accurate and is given as follows:

$$\frac{2k}{h_{i-1}h_i(h_{i-1} + h_i)}[-h_i\phi_{i-1} + (h_{i-1} + h_i)\phi_i - h_{i-1}\phi_{i+1}] = f_i \tag{5.4-46}$$

For uniform spacing the same finite difference algorithm is second-order accurate and reduces to

$$\frac{k}{h^2}(-\phi_{i-1} + 2\phi_i - \phi_{i+1}) = f_i \tag{5.4-47}$$

Symmetry of [A]

Suppose the index in (5.4-46) is increased by one to obtain the next equation:

$$\frac{2k}{h_i h_{i+1}(h_i + h_{i+1})}[-h_{i+1}\phi_i + (h_i + h_{i+1})\phi_{i+1} - h_i\phi_{i+2}] = f_{i+1} \tag{5.4-48}$$

Then the off-diagonal terms in the governing matrix [A] identified as $A_{i,i+1}$ and $A_{i+1,i}$ follow from (5.4-46) and (5.4-48):

$$A_{i,i+1} = \frac{-2k}{h_i(h_{i-1} + h_i)}\ , \qquad A_{i+1,i} = \frac{-2k}{h_i(h_i + h_{i+1})} \tag{5.4-49}$$

which indicates the matrix [A] is not symmetric for unequal mesh spacing. However, if all terms in (5.4-46) are multiplied by $(h_{i-1} + h_i)/2$ the result is the following equivalent form:

$$\frac{k}{h_{i-1}h_i}[-h_i\phi_{i-1} + (h_{i-1} + h_i)\phi_i - h_{i-1}\phi_{i+1}] = \frac{(h_{i-1} + h_i)}{2} f_i \tag{5.4-50}$$

Similarly, if the corresponding change is made to (5.4-48)

$$\frac{k}{h_i h_{i+1}}[-h_{i+1}\phi_i + (h_i + h_{i+1})\phi_{i+1} - h_i\phi_{i+2}] = \frac{(h_i + h_{i+1})}{2} f_{i+1} \tag{5.4-51}$$

then the coefficient of ϕ_{i+1} in (5.4-50) is identical to the coefficient of ϕ_i in (5.4-51) and the resulting system matrix is symmetric. If this form is adopted, then it makes sense to ensure that boundary conditions do not destroy symmetry. In the form given by (5.4-50) the term on the right side has an engineering interpretation of simply being the value of the forcing function at the node multiplied by the contributive area of one-half the mesh spacing on either side of the node. As we shall see, this form arises naturally with the finite element method.

Alteration of the Forcing Term

In analogy with the approach suggested in the previous subsection, suppose we replace (5.4-37) with

$$b_i = \eta_{i-1}f_{i-1} + (1 - \eta_{i-1} - \eta_{i+1})f_i + \eta_{i+1}f_{i+1} \tag{5.4-52}$$

in which η_{i-1} and η_{i+1} are parameters to be determined from the consistency equation. With the use of Taylor expansions and the differential equation, we obtain

$$\begin{aligned} b_i = -k\phi_{,xx} &- \eta_{i-1}k(-h_{i-1}\phi_{i,xxx} + \tfrac{1}{2}h_{i-1}^2\phi_{i,xxxx} - \cdots) \\ &- \eta_{i+1}k(h_i\phi_{i,xxx} + \tfrac{1}{2}h_i^2\phi_{i,xxxx} + \cdots) \end{aligned} \tag{5.4-53}$$

The expressions for c_0, c_1 and c_2 in (5.4-39) remain unchanged from that given in (5.4-40) and consequently, the solutions for α_{i-1}, α_i and α_{i+1} also remain unchanged from (5.4-41). Instead of (5.4-42), we now have

$$c_3 = -\frac{k}{3}(h_i - h_{i-1}) - \eta_{i-1}kh_{i-1} + \eta_{i+1}kh_i$$
$$c_4 = -\frac{k}{12}(h_{i-1}^2 - h_{i-1}h_i + h_i^2) + \eta_{i-1}k\frac{h_{i-1}^2}{2} + \eta_{i+1}k\frac{h_i^2}{2} \tag{5.4-54}$$

The result of setting $c_3 = 0$ and $c_4 = 0$ is

$$\eta_{i-1} = \frac{1}{3}\frac{(2h_{i-1}^2 - h_{i-1}h_i + h_i^2)}{h_{i-1}(h_{i-1} + h_i)}, \qquad \eta_{i+1} = \frac{1}{3}\frac{(2h_i^2 - h_{i-1}h_i + h_{i-1}^2)}{h_i(h_{i-1} + h_i)} \tag{5.4-55}$$

for a nonuniform mesh and

$$\eta_{i-1} = \frac{1}{3}, \qquad \eta_{i+1} = \frac{1}{3} \qquad \text{(uniform mesh)} \tag{5.4-56}$$

In summary, by simply altering b_i to the form of (5.4-52), the local truncation error is at least of third order, even for a nonuniform mesh.

5.4.3 A More Complex Model Problem

Now consider the model problem in which both an advection term and the function itself are present:

$$k\phi_{,xx} - a\phi_{,x} - b\phi + f(x) = 0, \qquad 0 < x < 1 \tag{5.4-57}$$

The coefficients, k, *a* and *b* are constant. Again suppose the three-point stencil of (5.4-35) is proposed as a potential candidate for a finite-difference algorithm. The local truncation error remains of the form given previously in (5.4-39) and (5.4-40) but with the coefficients

$$\left.\begin{aligned} c_0 &= \alpha_{i-1} + \alpha_i + \alpha_{i+1} - b \\ c_1 &= -h_{i-1}\alpha_{i-1} + h_i\alpha_{i+1} - a \\ c_2 &= \tfrac{1}{2}(h_{i-1}^2\alpha_{i-1} + h_i^2\alpha_{i+1}) + k \\ c_3 &= -\tfrac{1}{6}(h_{i-1}^3\alpha_{i-1} - h_i^3\alpha_{i+1}) \\ c_4 &= \tfrac{1}{24}(h_{i-1}^4\alpha_{i-1} + h_i^4\alpha_{i+1}) \end{aligned}\right\} \tag{5.4-58}$$

in which the following expression for the forcing term, b_i, has been assumed:

$$b_i = f_i = -k\phi_{i,xx}^e + a\phi_{i,x}^e + b\phi_i^e \tag{5.4-59}$$

The result of setting $c_0 = 0$, $c_1 = 0$ and $c_2 = 0$ is

$$\left.\begin{aligned}\alpha_{i-1} &= \frac{1}{h_{i-1}(h_{i-1}+h_i)}(-2k - ah_i)\\ \alpha_i &= \frac{1}{h_{i-1}h_i}[2k + a(h_{i-1}-h_i) + bh_{i-1}h_i]\\ \alpha_{i+1} &= \frac{1}{h_i(h_{i-1}+h_i)}(-2k + ah_{i-1})\end{aligned}\right\} \tag{5.4-60}$$

The resulting algorithm for a nonuniform mesh is the following:

$$\begin{aligned}\frac{1}{h_{i-1}h_i(h_{i-1}+h_i)}\Big\{&2k\left[-h_i\phi_{i-1} + (h_{i-1}+h_i)\phi_i - h_{i-1}\phi_{i+1}\right]\\ &+ a\left[-h_i^2\phi_{i-1} + (h_i^2 - h_{i-1}^2)\phi_i + h_{i-1}^2\phi_{i+1}\right]\Big] \;+\; b\phi_i \;=\; f_i\end{aligned} \tag{5.4-61}$$

For a uniform mesh, the algorithm reduces to

$$\frac{k}{h^2}(-\phi_{i-1} + 2\phi_i - \phi_{i+1}) + \frac{a}{2h}(-\phi_{i-1} + \phi_{i+1}) + b\phi_i = f_i \tag{5.4-62}$$

Note that the presence of the term with the coefficient a has the consequence that the coefficient matrix becomes nonsymmetric whether or not the mesh is uniform.

For a nonuniform mesh the coefficients for the remaining terms in the local truncation error of (5.4-39) are

$$\left.\begin{aligned}c_3 &= \tfrac{1}{6}[2k(h_{i-1}-h_i) + ah_{i-1}h_i]\\ c_4 &= \tfrac{1}{24}[-2k(h_{i-1}^2 - h_{i-1}h_i + h_i^2) - ah_{i-1}h_i(h_{i-1}-h_i)]\end{aligned}\right\} \tag{5.4-63}$$

and, for a uniform mesh, these expressions reduce to

$$c_3 = \frac{ah^2}{6}, \qquad c_4 = -\frac{kh^2}{12} \tag{5.4-64}$$

Now, as long as $a \neq 0$ the algorithm is first-order accurate whether or not the mesh is uniform. However, if $a = 0$ and $b \neq 0$ the algorithm is second-order accurate for a uniform mesh. Again, a higher-order algorithm can be obtained by choosing a form for b_i that includes contributions of the forcing function evaluated at nodes x_{I-1}, x_i and x_{I+1}.

5.5 BOUNDARY CONDITIONS

With the algorithm of the previous sections for interior points, the governing matrix equation assumes the form

$$[A]\{\phi\} = \{b\} \tag{5.5-1}$$

in which [A] and {b} are

$$[A]=\begin{bmatrix} ? & ? & ? & ? & \cdots \\ A_{21} & A_{22} & A_{23} & 0 & \cdots \\ 0 & A_{32} & A_{33} & A_{34} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \\ ? & ? & ? & ? & \cdots \end{bmatrix}, \qquad \{b\}=\begin{Bmatrix} ? \\ f_2 \\ f_3 \\ \vdots \\ ? \end{Bmatrix} \tag{5.5-2}$$

The entries of [A] are considered to be specified except for the first and last rows (denoted by ?). The first and last entries of {b} are also not known. These unknown terms must come from the boundary conditions, which are presumed to be associated with the first and last nodes, or boundary nodes, of the mesh. For convenience, consideration will be given only to the first node with the understanding that entries for boundary conditions at the last node are obtained in a similar manner.

Function Specification on the Boundary

First, consider the case when the function evaluated at x = 0 is specified, $\phi^*(0)$. Then the first row of [A] and the first component of {b} are chosen to be:

$$\langle A \rangle_1 = \langle 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad \cdots \rangle, \qquad b_1 = \phi^*(0) \tag{5.5-3}$$

With an appropriate modification for the other boundary condition, the algebraic solution for the values of ϕ at the nodes will automatically satisfy this boundary condition.

Although straightforward to implement, (5.5-3) destroys the symmetry of [A], which can be an important factor if the remaining part of the matrix is symmetric and a solver based on symmetry is being used. One method for retaining symmetry is to recognize that the set of equations involves the known product of $\phi^*(0)$ with the first column of [A] on the left side of the equality. This known vector can be taken to the right side and the first column of [A] adjusted to maintain symmetry. A similar operation is required when the function $\phi^*(1)$ is also prescribed.

The procedure when the function is prescribed at both boundary points is summarized as follows:

(i) Define a new force vector to be

$$\{b^*\} = \{b\} - \phi^*(0)\{A\}_1 - \phi^*(1)\{A\}_n \tag{5.5-4}$$

(ii) Override the first and last components of {b*} by setting

$$b_1^* = \phi^*(0), \qquad b_n^* = \phi^*(1) \tag{5.5-5}$$

(iii) Define a new matrix [A*] as the matrix [A] in which the first row and column of [A] are replaced with the unit vectors $\langle i \rangle_1 = \langle 1, 0, 0, \ldots \rangle$ and $\{i\}_1$, respectively; and the last rows and columns are similarly replaced with the unit vectors $\langle i \rangle_n = \langle \ldots, 0, 0, 1 \rangle$ and $\{i\}_n$.

For emphasis the governing matrix assumes the following form:

$$[A^*] = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & A_{22} & A_{23} & \cdots & 0 \\ 0 & A_{32} & A_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.5-6}$$

Another approach is to use the reduced form of (5.1-8), which involves only interior nodes

$$[A^I]\{\phi^I\} = \{b^I\} \tag{5.5-7}$$

in which the first row of $[A^I]$ remains unchanged and the first component of the modified force vector $\{b^I{}^*\}$ reflects the prescribed value of ϕ:

$$[A^I] = \begin{bmatrix} A_{22} & A_{23} & 0 & \cdots \\ A_{32} & A_{33} & A_{34} & \cdots \\ 0 & A_{43} & A_{44} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}_{(n-2)x(n-2)}, \qquad \{b^I{}^*\} = \begin{Bmatrix} f_2 - A_{21}\phi^*(0) \\ f_3 \\ f_4 \\ \vdots \end{Bmatrix}_{n-2} \tag{5.5-8}$$

Similarly, the last component of $\{b^I{}^*\}$ becomes $b^I_{n-2} = f_{n-1} - A_{n-1,n}\phi^*(1)$. The reduction to a smaller system might be more efficient in higher dimensions but many prefer to stay with an nxn matrix when there are n nodes.

Specification of Flux on the Boundary

Now suppose the boundary condition at x = 0 is a flux condition which is transformed to a prescribed value of the derivative, $\phi'^*(0)$. The approach for constructing a suitable finite difference algorithm is identical to that used for an interior point described in Subsection 5.4.1. If a forward difference algorithm is used, then the derivative is approximated with

$$\frac{1}{h_1}(\phi_2 - \phi_1) = \phi'^*(0) \tag{5.5-9}$$

and the local truncation error is first-order accurate. To maintain symmetry of [A] when the boundary condition is applied, multiply (5.5-9) by $h_1 A_{21}$. The analogue to (5.5-5) is

$$[A]=\begin{bmatrix} -A_{21} & A_{21} & 0 & 0 & \cdots \\ A_{21} & A_{22} & A_{23} & 0 & \cdots \\ 0 & A_{32} & A_{33} & A_{34} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \cdots \\ 0 & ? & ? & ? & \ddots \end{bmatrix}, \qquad \{b^*\}=\begin{Bmatrix} h_1 A_{21}\phi'^*(0) \\ f_2 \\ f_3 \\ \vdots \\ ? \end{Bmatrix} \tag{5.5-10}$$

For the approach using $[A^I]$, the variable ϕ_1 must be eliminated from the equation for node 2 by using the finite difference form of the boundary condition, (5.5-9). The result is to replace f_1^I with $f_2 + A_{21}h_1\phi_{,x}^*(0)$ and A_{22} with $A_{21}+A_{22}$. The analogue to (5.5-7) is the reduced set of equations

$$[A^I]=\begin{bmatrix} A_{21}+A_{22} & A_{23} & 0 & \cdots \\ A_{32} & A_{33} & A_{34} & \cdots \\ 0 & A_{43} & A_{44} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \qquad \{b^I\}=\begin{Bmatrix} f_2 + A_{21}h_1\phi_{,x}^*(0) \\ f_3 \\ f_4 \\ \vdots \end{Bmatrix} \tag{5.5-11}$$

A Higher-Order Algorithm for a Mixed Boundary Condition

Consider a boundary condition of the Robin form

$$\left[k\phi_{,x} + a^*\phi\right]_{x=0} = g^* \tag{5.5-12}$$

with k, a* and g* known. Let $f_1 = g^*$ and consider an algorithm of the form

$$\alpha_1\phi_1 + \alpha_2\phi_2 + \alpha_3\phi_3 = f_1 \tag{5.5-13}$$

Following the standard format outlined above, the local truncation error is

$$\tau_1 = c_0\phi_1^e + c_1\phi_{1,x}^e + c_2\phi_{1,xx}^e + c_3\phi_{1,xxx}^e \tag{5.5-14}$$

where

$$\left.\begin{aligned} c_0 &= \alpha_1 + \alpha_2 + \alpha_3 - a^* \\ c_1 &= h_1\alpha_2 + (h_1+h_2)\alpha_3 - k \\ c_2 &= \tfrac{1}{2}[h_1^2\alpha_2 + (h_1+h_2)^2\alpha_3] \\ c_3 &= \tfrac{1}{6}[h_1^3\alpha_2 + (h_1+h_2)^3\alpha_3] \end{aligned}\right\} \tag{5.5-15}$$

The result of setting $c_0 = 0$, $c_1 = 0$ and $c_2 = 0$ is

$$\alpha_1 = a^* - k\frac{(2h_1+h_2)}{h_1(h_1+h_2)}, \qquad \alpha_2 = k\frac{(h_1+h_2)}{h_1h_2}, \qquad \alpha_3 = -k\frac{h_1}{h_2(h_1+h_2)} \tag{5.5-16}$$

and as the mesh spacing approaches zero

$$\tau_1 = C_1^\tau h_1(h_1+h_2), \qquad C_1^\tau = -\frac{1}{6}k\phi_{0,xxx}^e \tag{5.5-17}$$

so the algorithm is second-order accurate. For uniform spacing, the corresponding results are

$$\alpha_1 = a^* - k\frac{3}{2h}, \qquad \alpha_2 = k\frac{2}{h}, \qquad \alpha_3 = -k\frac{1}{2h} \tag{5.5-18}$$

and the finite difference algorithm for the boundary node at x = 0 is

$$\frac{k}{2h}(-3\phi_1 + 4\phi_2 - \phi_3) + a^*\phi_1 = g^* \tag{5.5-19}$$

which can be checked with the forward difference algorithm given in (5.4-29). Because this equation provides coefficients in the first, second and third columns of the first row of [A], a more elaborate manipulation involving the first three rows is needed to preserve symmetry. A general procedure is provided in Chapter 15.

5.6 THE ADVECTION-DIFFUSION EQUATION

Consider the differential equation

$$k\phi_{,xx} - a\phi_{,x} = 0, \qquad 0 < x < 1 \tag{5.6-1}$$

which is a special case of the differential equation considered previously in Subsection 5.4.3. The term involving k corresponds to the physical process of **diffusion**; the term involving *a* is the **advection**, and can be interpreted as a contribution due to material moving with a velocity (wind) proportional to *a*. For simplicity, make the restriction of a uniform mesh in which case the second-order algorithm of (5.4-50) is

$$\frac{k}{h^2}(-\phi_{i-1} + 2\phi_i - \phi_{i+1}) + \frac{a}{2h}(-\phi_{i-1} + \phi_{i+1}) = 0 \tag{5.6-2}$$

Consider the specific boundary conditions $\phi(0) = 1$ and $\phi(1) = 0$. Introduce a "mesh Peclet number" as follows:

$$P_h = \frac{ah}{2k} \tag{5.6-3}$$

which is the coefficient of the second term after multiplying through by h^2/k. If the algorithm of (5.6-2) is used to obtain a numerical solution, it is observed that if the mesh is sufficiently coarse so that $P_h > 1$, then the numerical solution oscillates about the exact solution, and for large mesh Peclet numbers, the numerical solution can be very inaccurate. As the mesh is refined to the point where $P_h < 1$, convergence is displayed and the numerical solution is accurate. The elimination of these oscillations is an important and practical concern in the development of general purpose programs for application to large problems because, in two and three dimensions, the coarsest mesh consistent with the desired accuracy is the one normally desired in order to reduce computational costs. For diffusion-advection problems, the desired mesh is often one with a large mesh Peclet number. Even if the mesh is refined in the region of most

interest, oscillations that arise from an outlying coarse mesh can pollute the numerical solution everywhere. The research questions then are "What is the source of the problem?" and "What can be done to rectify the algorithm so the oscillations do not appear?". Here, some insight into the matter is given together with a possible solution that has been used extensively, namely, **upwind differencing**.

Recall that a matrix [A] is diagonally dominant if

$$|A_{ii}| \geq \sum_{j \neq i}^{n} |A_{ij}| \tag{5.6-4}$$

with strict inequality satisfied for at least one row. The typical terms in a row provided by (5.6-2) with $a = 0$ are

$$A_{i,i-1} = -\frac{k}{h^2}\ , \qquad A_{ii} = \frac{2k}{h^2}\ , \qquad A_{i,i+1} = -\frac{k}{h^2} \tag{5.6-5}$$

so that (5.6-4) is barely met with an equality sign for every interior node. Therefore, the inequality must be provided by a boundary equation which is the case if the function is prescribed, for then the diagonal entry of the matrix is nonzero; all other entries in the row are zero. The inequality is also met by the equation for the node adjacent to the boundary if the approach involving $[A^I]$ is used.

Now consider the case when $a \neq 0$. Then

$$A_{i,i-1} = -\frac{k}{h^2} - \frac{a}{2h}\ , \qquad A_{ii} = \frac{2k}{h^2}\ , \qquad A_{i,i+1} = -\frac{k}{h^2} + \frac{a}{2h} \tag{5.6-6}$$

Suppose $a > 0$ and $\dfrac{a}{2h} < \dfrac{k}{h^2}$ ($P_h < 1$). Then

$$|A_{i,i-1}| + |A_{i,i+1}| = \frac{k}{h^2} + \frac{a}{2h} + \frac{k}{h^2} - \frac{a}{2h} = \frac{2k}{h^2} \tag{5.6-7}$$

and the diagonal dominance condition is satisfied. However suppose $\dfrac{a}{2h} > \dfrac{k}{h^2}$ ($P_h > 1$). Then

$$|A_{i,i-1}| + |A_{i,i+1}| = \frac{k}{h^2} + \frac{a}{2h} - \frac{k}{h^2} + \frac{a}{2h} = \frac{a}{h} \tag{5.6-8}$$

which is greater than the diagonal term so diagonal dominance does not hold, and it is possible that stability is lost. The same results hold if a is negative. An alternative, possibly equivalent, explanation for the appearance of oscillations is that with the nonsymmetric terms in the matrix, as indicated by (5.6-2), there is the likelihood that the eigenvalues will become complex. To show this possibility, consider a four-node uniform mesh with ϕ prescribed at each end. The governing matrix becomes

$$[A]=\frac{k}{h^2}\begin{bmatrix} 1 & 0 & 0 & 0 \\ -1-P_h & 2 & -1+P_h & 0 \\ 0 & -1-P_h & 2 & -1+P_h \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.6-9}$$

for which two of the eigenvalues are k/h^2 and the other two are

$$\lambda_{3,4}=\frac{k}{h^2}\left[\,2\pm\sqrt{1-P_h^2}\,\right] \tag{5.7-10}$$

It is seen that, indeed, the eigenvalues are complex if $P_h > 1$.

An approach to rectify the problem is to use a first-order backward difference algorithm for the advection term, in which case (5.6-2) is replaced with the following:

$$\frac{k}{h^2}(-\phi_{i-1}+2\phi_i-\phi_{i+1})+\frac{a}{h}(-\phi_{i-1}+\phi_i)=0 \tag{5.6-11}$$

For the algorithm of (5.6-11) there follows

$$A_{i,i-1}=-\frac{k}{h^2}-\frac{a}{2h}\,,\qquad A_{ii}=\frac{2k}{h^2}+\frac{a}{h}\,,\qquad A_{i,i+1}=-\frac{k}{h^2} \tag{5.6-12}$$

and for positive *a*

$$|A_{ii}|=\frac{2k}{h^2}+\frac{a}{h}\,,\qquad |A_{i,i-1}|+|A_{i,i+1}|=\frac{2k}{h^2}+\frac{a}{2h} \tag{5.6-13}$$

so the diagonal dominance criterion is always met. Analogous to the model problem reflected in (5.6-9), the governing matrix is

$$[A]=\frac{k}{h^2}\begin{bmatrix} 1 & 0 & 0 & 0 \\ -1-2P_h & 2(1+P_h) & -1 & 0 \\ 0 & -1-2P_h & 2(1+P_h) & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.6-14}$$

for which the two interesting eigenvalues are

$$\lambda_{3,4}=\frac{k}{h^2}\left[\,2(1+P_h)\pm\sqrt{1+2P_h}\,\right] \tag{5.6-15}$$

We see that even though the matrix is not symmetric the eigenvalues are always positive and real no matter what value the mesh Peclet number assumes.

Since the term, *a*, is associated with a velocity in the direction of positive x, the involvement of a contribution from node x_{i-1} to the equation for node i implies that information "upwind" is being used and, consequently, the term upwind differencing is applied in this context. Unfortunately, if *a* is a variable that depends on ϕ, which is the case for fluid flow, the differencing scheme must account for the sign of ϕ (and the direction of ϕ in two and three dimensions). This complication, together with the fact that the method is first-order accurate even on a uniform mesh, provides one reason why

better schemes are being sought. Nevertheless, for the simple model problem given above, the change of the algorithm to a lower order one removes the oscillations.

5.7 VARIABLE COEFFICIENTS

Preliminary Comments

The majority of problems of technical interest involve cases where the coefficient functions, k, *a* and *b* and the forcing function, f, vary with x either smoothly, or with jumps. In addition, there is the possibility of point forces. Here, we provide an introductory development to show how some of these various cases can be handled. All possibilities are not considered but it is hoped that the basic concepts can be extended by the reader when required.

Smoothly Varying Coefficient

The case of variable f has already been considered. Here we generalize slightly by allowing k to also depend smoothly on x but, for the sake of simplicity, we set $a = 0$ and $b = 0$. The model problem becomes

$$[k(x)\phi_{,x}]_{,x} + f(x) = 0 , \qquad 0 < x < 1 \tag{5.7-1}$$

in which k(x) is assumed to be a function of class C^1. Replace (5.7-1) with the following equivalent form:

$$k\phi_{,xx} + k_{,x}\phi_{,x} + f(x) = 0 , \qquad 0 < x < 1 \tag{5.7-2}$$

First, we suppose that k(x) is varying sufficiently slowly so that the term involving $k_{,x}$ can be ignored. Then the differential equation, when applied at each node, yields

$$-f_i = k_i\phi_{i,xx} \tag{5.7-3}$$

If the three-point stencil of (5.4-35) is used, and the procedure of Section 5.4 followed with $b_i = f_i$, the result is the set of two finite difference algorithms

$$\frac{2k_i}{h_{i-1}h_i(h_{i-1}+h_i)}[-h_i\phi_{i-1} + (h_{i-1}+h_i)\phi_i - h_{i-1}\phi_{i+1}] = f_i \tag{5.7-4}$$

$$\frac{k_i}{h^2}(-\phi_{i-1} + 2\phi_i - \phi_{i+1}) = f_i \tag{5.7-5}$$

for nonuniform and uniform spacing, respectively. The resulting matrix will not be symmetric for either case unless (5.7-4) and (5.7-5) are replaced with the following:

$$\frac{1}{h_{i-1}h_i}[-h_i\phi_{i-1} + (h_{i-1}+h_i)\phi_i - h_{i-1}\phi_{i+1}] = \frac{(h_{i-1}+h_i)}{2k_i}f_i \tag{5.7-6}$$

$$\frac{1}{h}(-\phi_{i-1} + 2\phi_i - \phi_{i+1}) = \frac{h}{k_i}f_i \tag{5.7-7}$$

The approach of ignoring $k_{,x}$ is equivalent to the assumption that the function k can be replaced with a piecewise constant function as shown in Fig. 5.7-1. For many engineering problems, this approach is quite adequate. However, the method has not been shown to be consistent so that convergence with mesh refinement, although highly probable, has not been proven.

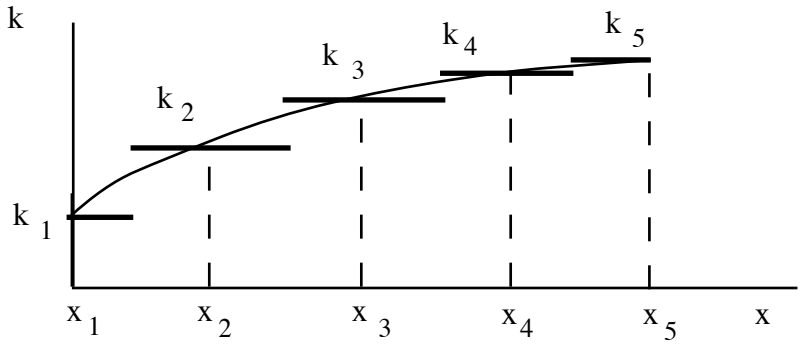

Fig. 5.7-1. The replacement of k(x) with a piecewise constant function.

If k(x) is rapidly varying, then the derivative must be retained and the differential equation assumes a form analogous to that of (5.4-57) with $a \neq 0$ and $b = 0$. For each node the differential equation yields

$$-f_i = k_i \phi_{i,xx} + k_{i,x} \phi_{i,x} \tag{5.7-8}$$

in which $k_{i,x} = k_{,x}(x_i)$. The resulting three-point stencil is given by (5.4-61) with k and a replaced by k_i and $k_{i,x}$, respectively, and with $b = 0$.

An alternative approach that eliminates the need for the derivative of k at x_i is to use a finite difference algorithm for $k_{,x}$ with care taken to ensure that the order of accuracy is not lost. For example, suppose the second-order, central-difference algorithm for the first derivative of (5.4-32) is used for $k_{,x}$ at x_i. The result is

$$k_{i,x} = \frac{1}{h_{i-1} h_i (h_{i-1} + h_i)} [-h_i^2 k_{i-1} - (h_{i-1}^2 - h_i^2) k_i + h_{i-1}^2 k_{i+1}] \tag{5.7-9}$$

The three-point stencil of (5.4-27) then yields the following algorithm:

$$\begin{aligned} &\frac{1}{h_{i-1}^2 h_i^2 (h_i + h_{i-1})^2} \Big[\{h_i^2 k_{i-1} - (h_{i-1} + h_i)^2 k_i - h_{i-1}^2 k_{i+1}\} h_i^2 \phi_{i-1} \\ &- \{h_i^2 (h_{i-1} - h_i) k_{i-1} - (h_{i-1} + h_i)(4 h_{i-1} h_i - h_{i-1}^2 - h_i^2) k_i \\ &\qquad - h_{i-1}^2 (h_{i-1} - h_i) k_{i+1}\} (h_{i-1} + h_i) \phi_i \\ &\qquad - \{h_i^2 k_{i-1} + (h_{i-1} + h_i)^2 k_i - h_{i-1}^2 k_{i+1}\} h_{i-1}^2 \phi_{i+1} \Big] = f_i \end{aligned} \tag{5.7-10}$$

For constant k, we recover (5.4-46). For a uniform mesh, (5.7-10) reduces to

$$\frac{1}{4h^2} [(k_{i-1} - 4k_i - k_{i+1}) \phi_{i-1} + 8 k_i \phi_i - (k_{i-1} + 4k_i - k_{i+1}) \phi_{i+1}] = f_i \tag{5.7-11}$$

which becomes (5.4-47) for constant k.

A Symmetric Coefficient Matrix

The coefficient matrix for the algorithm of (5.7-11) is not symmetric even though the spacing is uniform. Here, an approach is given that provides a symmetric matrix. Additional nodes are introduced at locations halfway between the existing ones as shown in Fig. 5.7-2. Suppose these half-way nodes are used for the central-difference algorithm as applied to the derivative of $k\phi_{,x}$.

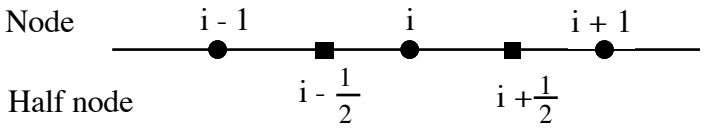

Fig. 5.7-2. The use of intermediate nodes.

Because the new set of nodes are a distance h/2 apart, the denominator of 2h in (5.4-22) becomes h and we obtain the following result:

$$\frac{d}{dx}(k\frac{d\phi}{dx}) \rightarrow \frac{\delta}{\delta x}(k\frac{d\phi}{dx})_i = \frac{1}{h}[k_{i+\frac{1}{2}}\frac{d}{dx}(\phi_{i+\frac{1}{2}}) - k_{i-\frac{1}{2}}\frac{d}{dx}(\phi_{i-\frac{1}{2}})] \tag{5.7-12}$$

Suppose the same central-difference algorithm is used for the derivatives of ϕ:

$$\left.\begin{aligned} \frac{d}{dx}\phi_{i+\frac{1}{2}} &\rightarrow \frac{\delta}{\delta x}\phi_{i+\frac{1}{2}} = \frac{1}{h}[\phi_{i+1} - \phi_i] \\ \frac{d}{dx}\phi_{i-\frac{1}{2}} &\rightarrow \frac{\delta}{\delta x}\phi_{i-\frac{1}{2}} = \frac{1}{h}[\phi_i - \phi_{i-1}] \end{aligned}\right\} \tag{5.7-13}$$

The substitution of (5.7-13) in (5.7-12) yields the following algorithm for the equation $(k\phi_{,x})_{,x} + f = 0$:

$$\frac{1}{h^2}[-k_{i-\frac{1}{2}}\phi_{i-1} + (k_{i-\frac{1}{2}} + k_{i+\frac{1}{2}})\phi_i - k_{i+\frac{1}{2}}\phi_{i+1}] = f_i \tag{5.7-14}$$

By considering entries in the coefficient matrix for two consecutive equations, it can be shown that the resulting coefficient matrix is symmetric.

Discontinuous Functions and Point Forces

Now we combine several features within one algorithm again under the assumption that $a = 0$ and $b = 0$. Suppose k is piecewise constant but the discontinuity is now displayed at the node x_i. In addition, suppose that f(x) also exhibits a discontinuity and that a point force, P_i, is applied at the same point. Then from the analysis of Subsection 1.6.2 it is known that the function ϕ is continuous and the discontinuity in flux must equal the point source, or

$$\left.\begin{aligned} \phi_i^- &= \phi_i^+ \\ k_i^- \phi_{i,x}^- - k_i^+ \phi_{i,x}^+ &= P_i \end{aligned}\right\} \qquad \text{at } x = x_i \tag{5.7-15}$$

in which the subscripts "-" and "+" denote values just to the left and right, respectively, of x_d, as indicated in Fig. 5.7-3. In addition, we invoke the differential equation as follows:

$$\left.\begin{aligned} k_i^- \phi_{i,xx}^- + f_i^- &= 0 \\ k_i^+ \phi_{i,xx}^+ + f_i^+ &= 0_i \end{aligned}\right\} \qquad \text{at } x = x_i \tag{5.7-16}$$

We propose a three-point stencil. With $\phi_i = \phi_i^- = \phi_i^+$ the local truncation error becomes

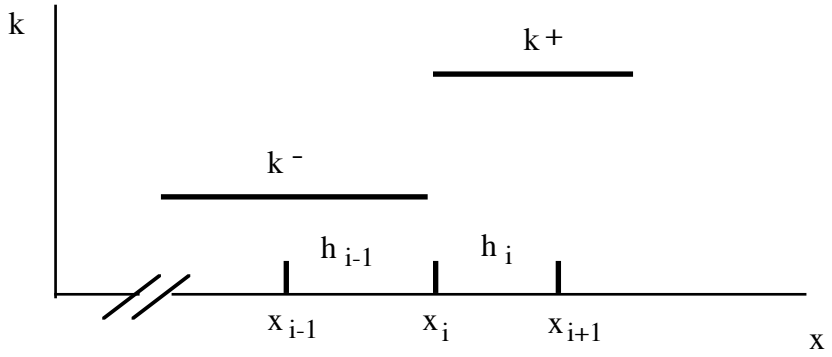

Fig. 5.7-3. Notation for a problem with a jump in k.

$$\begin{aligned} \tau_i &= \alpha_{i-1}\phi_{i-1} + \alpha_i\phi_i + \alpha_{i+1}\phi_{i+1} - b_i \\ &= \alpha_{i-1}(\phi_i - h_{i-1}\phi_{i,x}^- + \tfrac{1}{2}h_{i-1}^2\phi_{i,xx}^- - \tfrac{1}{6}h_{i-1}^3\phi_{i,xxx}^- + \cdots) \\ &\quad + \alpha_i\phi_i - b_i \\ &\quad + \alpha_{i+1}(\phi_i + h_i\phi_{i,x}^+ + \tfrac{1}{2}h_i^2\phi_{i,xx}^+ + \tfrac{1}{6}h_i^3\phi_{i,xxx}^+ + \cdots) \end{aligned} \tag{5.7-17}$$

With the substitution of the second of (5.7-15) and (5.7-16) in (5.7-17), we obtain

$$\begin{aligned} \tau_i &= \phi_i(\alpha_{i-1} + \alpha_i + \alpha_{i+1}) - b_i \\ &\quad - \frac{1}{k_i^-}(\alpha_{i-1}h_{i-1}k_i^+ + \alpha_{i+1}k_i^- h_i)\phi_{i,x}^+ \\ &\quad - \alpha_{i-1}h_{i-1}\frac{P_i}{k_i^-} - \alpha_{i-1}h_{i-1}^2\frac{f_i^-}{2k_i^-} - \alpha_{i+1}h_i^2\frac{f_i^+}{2k_i^+} \\ &\quad - \alpha_{i-1}\tfrac{1}{6}h_{i-1}^3\phi_{i,xxx}^- + \alpha_{i+1}\tfrac{1}{6}h_i^3\phi_{i,xxx}^+ + \cdots \end{aligned} \tag{5.7-18}$$

By setting the coefficients of ϕ and its derivative equal to zero, and by choosing b_i so as to cancel the loading terms, we obtain

$$\begin{aligned}
&\alpha_{i-1} + \alpha_i + \alpha_{i+1} = 0 \\
&\alpha_{i-1} h_{i-1} k_i^+ + \alpha_{i+1} k_i^- h_i = 0 \\
&b_i = \alpha_{i-1} h_{i-1} \frac{P_i}{k_i^-} + \alpha_{i-1} h_{i-1}^2 \frac{f_i^-}{2k_i^-} + \alpha_{i+1} h_i^2 \frac{f_i^+}{2k_i^+}
\end{aligned} \tag{5.7-19}$$

If there is no point force, and if the forcing function is continuous, we want $b_i = f_i$. This suggests that as a third equation for determining the set α_{i-1}, α_i and α_{i+1}, we choose

$$\alpha_{i-1} h_{i-1}^2 \frac{1}{2k_i^-} + \alpha_{i+1} h_i^2 \frac{1}{2k_i^+} = 1 \tag{5.7-20}$$

Now, (5.7-19) and (5.7-20) yield

$$\begin{aligned}
&\alpha_{i-1} = \frac{2k_i^-}{h_{i-1}(h_{i-1} + h_i)}, \qquad \alpha_{i+1} = \frac{2k_i^+}{h_i (h_{i-1} + h_i)} \\
&\alpha_i = -\frac{2}{(h_{i-1} + h_i)} \left(\frac{k_i^-}{h_{i-1}} + \frac{k_i^+}{h_i} \right) \\
&b_i = \frac{2P_i}{(h_{i-1} + h_i)} + \frac{h_{i-1} f_i^-}{(h_{i-1} + h_i)} + \frac{h_i f_i^+}{(h_{i-1} + h_i)}
\end{aligned} \tag{5.7-21}$$

The contribution of the point force can be considered equivalent to a distributed force if the point force is uniformly distributed over half the mesh on either side of the node x_i. Note the similarity of these parameters with those obtained for the continuous case given in (5.4-46). For uniform spacing, these expressions become

$$\begin{aligned}
&\alpha_{i-1} = \frac{k_i^-}{h^2}, \qquad \alpha_i = -\frac{(k_i^- + k_i^+)}{h^2} \\
&\alpha_{i+1} = \frac{k_i^+}{h^2}, \qquad b_i = \frac{P_i}{h} + \frac{1}{2}(f_i^- + f_i^+)
\end{aligned} \tag{5.7-21}$$

and represent a plausible generalization of (5.4-47).

The coefficients of $\phi_i^-{}_{,xxx}$ and $\phi_i^+{}_{,xxx}$ in (5.7-18) become

$$-\alpha_{i-1} \tfrac{1}{6} h_{i-1}^3 = -\frac{k_i^- h_{i-1}^2}{3(h_{i-1} + h_i)}, \qquad \alpha_{i+1} \tfrac{1}{6} h_i^3 = \frac{k_i^+ h_i^2}{3(h_{i-1} + h_i)} \tag{5.7-22}$$

so the algorithm is first-order accurate whether or not the mesh is uniform.

An Alternative Derivation

Here, we retain the terms involving *a* and *b* in the differential equation but assume they are constant. The model problem becomes

$$(k\phi_{,x})_{,x} - a\phi_{,x} - b\phi + f(x) = 0\ , \qquad 0 < x < 1 \tag{5.7-23}$$

Integrate each term in the differential equation over a segment involving half the distance to the adjoining nodes about the node x_i, with beginning and end coordinates of $x_b = x_i - h_{i-1}/2$ and $x_e = x_i + h_i/2$, respectively:

$$\int_{x_b}^{x_e} \left[(k\phi_{,x})_{,x} - a\phi_{,x} - b\phi + f^r(x) + P_i\delta[x - x_i]\right] dx = 0 \tag{5.7-24}$$

in which $f^r(x)$ is the remaining part of the forcing function after the point contribution, P_i, is separated out of f(x). Let $\hat{f}_i$ be the integral of $f^r(x)$ so that

$$\int_{x_b}^{x_e} \left[f^r(x) + P_i\delta[x - x_i]\right] dx = \hat{f}_i + P_i \tag{5.7-25}$$

For the case when $f^r(x) = f_i$, a constant over the interval, then

$$\hat{f}_i = \frac{(h_{i-1} + h_i)}{2} f_i \tag{5.7-26}$$

The integral of the first term in (5.7-23) becomes

$$\begin{aligned} \int_{x_b}^{x_e} (k\phi_{,x})_{,x}\, dx &= k\phi_{,x}\Big|_{x_b}^{x_e} \\ &= k^+\phi_{i,x}^+ + k^+\frac{h_i}{2}\phi_{i,xx}^+ + \cdots - k^-\phi_{i,x}^- + k^-\frac{h_{i-1}}{2}\phi_{i,xx}^- - \cdots \end{aligned} \tag{5.7-27}$$

in which Taylor series to the right and left have been used to evaluate terms on the respective sides of the discontinuity. Similarly, the next two terms on the left side of (5.7-23) yield

$$\begin{aligned} \int_{x_b}^{x_e} a\phi_{,x}\, dx &= a\phi\Big|_{x_b}^{x_e} = a\left[\phi_i + \frac{h_i}{2}\phi_{i,x}^+ + \cdots - \{\phi_i - \frac{h_{i-1}}{2}\phi_{i,x}^- + \cdots\}\right] \\ \int_{x_b}^{x_e} b\phi\, dx &= b\int_{x_b}^{x_i} \left[\phi_i + (x - x_i)\phi_{i,x}^- + \cdots\right] dx + b\int_{x_i}^{x_e} \left[\phi_i + (x - x\_)\phi_{i,x}^+ + \cdots\right] dx \\ &= b\left[\frac{(h_{i-1} + h_i)}{2}\phi_i + \frac{h_i^2}{8}\phi_{i,x}^+ - \frac{h_{i-1}^2}{8}\phi_{i,x}^- + \cdots\right] \end{aligned} \tag{5.7-28}$$

in which higher-order terms have not been included. Now we propose an algorithm based on a three-point stencil of the form

$$\alpha_{i-1}\phi_{i-1} + \alpha_i\phi_i + \alpha_{i+1}\phi_{i+1} = \hat{f}_i + P_i \tag{5.7-29}$$

The use of (5.7-25), and the requirement that each of the coefficients of ϕ_i, $\phi_i^+{}_{,x}$ and $\phi_i^-{}_{,x}$ be zero, result in three equations for determining α_{i-1}, α_i and α_{i+1}. In this derivation, it turns out that only the first term in the coefficient of *b* in (5.7-22) is required to obtain expressions with a consistent exponent of mesh spacing. The result is the algorithm

$$\begin{aligned} -\left(\frac{k^L}{h_{i-1}} + \frac{a}{2}\right)\phi_{i-1} + \left[\frac{k^L}{h_{i-1}} + \frac{k^R}{h_i} + \frac{b}{2}(h_{i-1} + h_i)\right]\phi_i \\ -\left(\frac{k^R}{h_i} - \frac{a}{2}\right)\phi_{i+1} = \hat{f}_i + P_s \end{aligned} \tag{5.7-30}$$

If each term in (5.7-22) is multiplied by $(h_{I-1} + h_i)$, the resulting algorithm is identical to 5.7-30) with the exception that (5.7-30) includes terms involving *a* and *b*. This derivation merely serves to illustrate the point that there may be several ways to interpret the basis for a given algorithm.

5.8 NUMERICAL DETERMINATION OF RATE OF CONVERGENCE

Suppose a differential equation with a corresponding finite difference algorithm is available and it is desired to determine the rate of convergence. Frequently the algorithm is too complicated to obtain an analytical expression for the error so a numerical approach is used instead. Based on analyses of simpler algorithms, as given previously in this chapter, an upper bound to the norm of the error function, ε_p^u, is assumed to be of the form

$$\varepsilon_p \le \varepsilon_p^u = C^\varepsilon h^r \tag{5.8-1}$$

Suppose it is desired to determine the rate of convergence, r. The procedure is to first choose a function, $\phi^e(x)$, to be an exact solution to the boundary value problem. Then, obtain the forcing function so that the differential equation is satisfied. Next, select boundary conditions consistent with the assumed solution. Finally, obtain an approximate solution based on the finite difference method, and determine the error, ε_p. Consider ε_p to be a reasonable estimate of ε_p^u. Repeat the procedure for several values of h. Plot the results in log-log form and fit a straight line through the plotted points, as indicated in Fig. 5.8-1. Based on (5.8-1)

$$\ln(\varepsilon_p^u) = \ln(C^\varepsilon) + r \ln(h) \tag{5.8-2}$$

which is the equation of a line. The intercept of the line with the ordinate provides an estimate for $\ln(C^\varepsilon)$ and the slope of the line is an approximation for r, a numerical estimate of the rate of convergence.

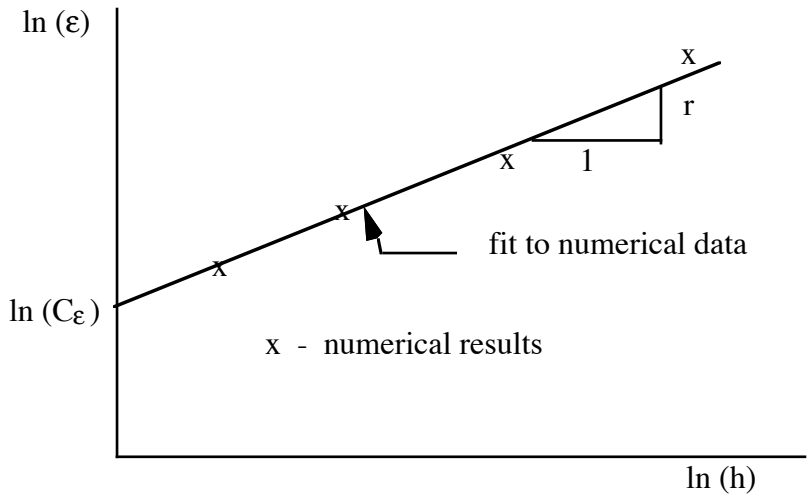

Fig. 5.8-1. Numerical determination of rate of convergence.

5.9 LAX'S EQUIVALENCE THEOREM

In Section 5.3, an outline of the steps were given for **Lax's Equivalence Theorem**: stability and consistency imply convergence. Because this theorem is essential to the establishment of finite difference algorithms, we go over the proof again but now in an abbreviated manner because many of the terms and the notation should now be familiar. In the process, more care is given to an aspect related to an assumption that was invoked previously

The result of spatial discretization is the following matrix equation:

$$[A]\{\phi\} - \{f\} = \{0\} \tag{5.9-1}$$

The local truncation error is the residual obtained by substituting the analytical solution in the approximating equation of (5.9-1):

$$[A]\{\phi^e\} - \{f\} = \{\tau\} \tag{5.9-2}$$

The result of subtracting corresponding terms in (5.9-1) and (5.9-2) is

$$[A]\{e\} = \{\tau\} \tag{5.9-3}$$

with the error vector denoting the difference between the exact and approximate solutions:

$$\{e\} = \{\phi^e\} - \{\phi\} \tag{5.9-4}$$

The solution to (5.9-3) is

$$\{e\} = [A]^{-1}\{\tau\} \tag{5.9-5}$$

If a compatible matrix norm is used, the result of Section 4.4 yields

$$e_p \le \|[A]^{-1}\|_p \|\{\tau\}\|_p \tag{5.9-6}$$

Suppose a 2-norm is used. If [A] is positive definite, a situation which holds for a large number of problems, then the 2-norm of a matrix is simply the maximum eigenvalue:

$$A_2 \equiv \|[A]^{-1}\|_2 = \lambda_{max}([A]^{-1}) = 1/\lambda_{min}([A]) \equiv 1/\lambda_1 \tag{5.9-7}$$

with the stability condition $\lambda_1 > 0$ assumed to hold. Then, an alternative form for (5.9-6) is

$$e_p \le \frac{1}{\lambda_1}\|\{\tau\}\|_p \tag{5.9-8}$$

From consistency with a uniform mesh, each component of $\{\tau\}$ is of the form $\tau_i = c_i^\tau h^r$. If $|c_i^\tau| \le C^\tau$ it follows from (5.3-11) that the 2-norm of the local truncation error and of the error function satisfy the following inequalities:

$$\|\{\tau\}\|_2 \le C^\tau h^{r-(1/2)}, \qquad \varepsilon_2 \le C^\tau \lambda_1 h^r \tag{5.9-9}$$

The error will approach zero with decreasing h provided λ_1 does not depend on h, the assumption made in Section 5.3.

EOP

To indicate that the spatial independence of λ_1 is plausible, at least for small h, consider the formulation in which only internal nodes are considered and $[A^I]$ is of the following tridiagonal form:

$$[A^I] = \frac{k}{h^2}[A^{I*}], \qquad [A^{I*}] = \begin{bmatrix} \beta & \alpha & 0 & 0 & 0 & 0 & \cdots \\ \alpha & \beta & \alpha & 0 & 0 & 0 & \cdots \\ 0 & \alpha & \beta & \alpha & 0 & 0 & \cdots \\ 0 & 0 & \alpha & \beta & \alpha & 0 & \cdots \\ 0 & 0 & 0 & \alpha & \beta & \alpha & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \alpha & \beta \end{bmatrix}_{n-2,n-2} \tag{5.9-10}$$

From Smith [1985; page 154], or see Exercise 6, the eigenvalues of the tridiagonal matrix [A*] are given by the expression

$$\lambda_j^* = \beta + 2\alpha\cos\left(\frac{j\pi}{n-1}\right), \qquad j = 1,..., n-2 \tag{5.9-11}$$

For example, the second-order-accurate algorithm of (5.4-36) yields the values $\alpha = -1$, $\beta = 2$ and the result from (5.9-11) is

$$\lambda_j^* = 2\left(1 - \cos\frac{j\pi}{n-1}\right), \qquad j = 1,..., n-2 \tag{5.9-12}$$

The minimum eigenvalue is

$$\lambda_1^* = 2(1-\cos\frac{\pi}{n-1}) = 2[1-\{1-\frac{\pi^2}{2(n-1)^2}+\cdots\}] \cong \frac{\pi^2}{(n-1)^2} \qquad (5.9\text{-}13)$$

where the approximation holds for large n. The mesh size is h = 1/(n-1) so that

$$\lambda_1^* \cong \pi^2 h^2 \qquad \text{and} \qquad \lambda_1 = \frac{k}{h^2}\lambda_1^* \cong k\pi^2 \qquad (5.9\text{-}14)$$

which indicates that the lowest eigenvalue of $[A^I]$ is independent of h. Therefore, the rate of convergence for the error is the exponent of the mesh size that appears in (5.9-9).

Similarly, (5.9-12) indicates that the largest eigenvalue is $\lambda_{n-2}^* \cong 4$ because $(n-2)/(n-1) \cong 1$ for large n. Recall from (5.9-10) that the matrix of interest involves a factor k/h^2 so that $\lambda_{max} \cong 4k/h^2$. The condition number is then

$$c[A^I] = \frac{\lambda_{max}}{\lambda_{min}} \cong \frac{4}{\pi^2 h^2} \qquad (5.9\text{-}15)$$

which increases quadratically in 1/h as h decreases. This result provides some substance for the statement that the condition number generally becomes larger with mesh refinement.

5.10 USE OF A SCALED NORM

Recall that the norms of the local truncation error, the vector of nodal errors, and the error function are related as follows:

$$\|\{\tau\}\|_p \le \|\{c^\tau\}\|_p h^r \le C^\tau h^{r-(1/p)}, \qquad \varepsilon_p \le h^{1/p} e_p, \qquad e_p = \Big[\sum_{i=1}^{n} |e_i|^p\Big]^{1/p} \qquad (5.10\text{-}1)$$

A mesh refinement study results in solution vectors with different numbers of components. Consequently, different vector spaces are being defined but the vector error norms on these spaces are implicitly being compared because the error measure of fundamental interest is ε_p. The result is the annoying factor, $h^{1/p}$, which appears in various terms in (5.10-1) with the consequence that it becomes quite possible that one might infer an incorrect rate of convergence if the norm of the wrong vector is inadvertently used. Here, it is suggested that the use of the scaled p-norm is a useful procedure for taking into account the different dimensions of the vector spaces. It is the norm that arises naturally in the development of the formulas for numerical determination of error.

Define a **scaled p-norm** of a vector, {v}, to be

$$\|\{v\}\|_{sp} = \frac{1}{n^{1/p}}\|\{v\}\|_p = \frac{1}{n^{1/p}}\Big[\sum_{i=1}^{n} |v_i|^p\Big]^{1/p}, \qquad p \ge 1 \qquad (5.10\text{-}2)$$

The scaled p-norm satisfies the rules for a norm given in Subsection 4.4. Special cases are the scaled one, two and infinity norms:

$$\left.\begin{aligned} \|\{v\}\|_{s1} &= \frac{1}{n}\sum_{i=1}^{n}|v_i| = \frac{1}{n}\|\{v\}\|_1 \\ \|\{v\}\|_{s2} &= \frac{1}{\sqrt{n}}\Big[\sum_{i=1}^{n}|v_i|^2\Big]^{1/2} = \frac{1}{\sqrt{n}}\|\{v\}\|_2 \\ \|\{v\}\|_{s\infty} &= \max_i|v_i| = \|\{v\}\|_\infty \end{aligned}\right\} \qquad (5.10\text{-}3)$$

One nice property of the scaled p-norm follows from (4.5-8); namely, the norm increases with p so that, for example,

$$\|\{v\}\|_{s1} \le \|\{v\}\|_{s2} \le \|\{v\}\|_{s\infty} \qquad (5.10\text{-}4)$$

Recall that a compatible matrix norm is defined such that

$$\|[A]\| = \sup_{\{v\}\neq 0}\frac{\|[A]\{v\}\|}{\{v\}} \qquad (5.10\text{-}5)$$

The scaling factor drops out so the p-norm of a matrix is also compatible with the scaled p-norm of a vector.

For applications to error analysis based on the finite difference method, note that

$$h \cong \frac{1}{n} \qquad \text{for large } n \qquad (5.10\text{-}6)$$

Then the scaled norm of the vector of nodal errors becomes

$$e_{sp} = (\tfrac{1}{n})^{1/p}\Big[\sum_{i=1}^{n}|e_i|^p\Big]^{1/p} = h^{1/p}\Big[\sum_{i=1}^{n}|e_i|^p\Big]^{1/p} \qquad (5.10\text{-}7)$$

and (5.10-1) becomes

$$\varepsilon_p \le e_{sp} \qquad (5.10\text{-}8)$$

i.e., the scaled p-norm of the error vector is an upper bound to the p-norm of the error function. Similarly, if each component of the local truncation error is of order h^r, then so is τ_{sp}, the scaled p-norm of the vector of local truncation error.

Recall from (5.3-25) that with the conventional p-norm the components of the local truncation error and various inequalities involving measures of error are the following:

$$\left.\begin{aligned} \tau_i &= c_i^\tau h^r, & \|\{\tau\}\|_p &\le C^\tau h^{r-(1/p)} \\ e_p &\le C^e h^{r-(1/p)}, & \varepsilon_p &\le C^e h^r \end{aligned}\right\} \qquad (5.10\text{-}9)$$

With the use of a scaled p-norm, the corresponding quantities become

$$\begin{aligned} \tau_i &= c_i^{\tau} h^r , \qquad & \left\| \{\tau\} \right\|_{sp} &\le C^{\tau} h^r \\ e_{sp} &\le C^{e} h^r , \qquad & \varepsilon_p &\le C^{e} h^r \end{aligned} \tag{5.10-10}$$

The result is that the scaled p-norm of the local truncation error, τ_{sp}, and of the error, e_{sp}, now display the same exponent for h as the exponent that appears in the p-norm of the error function, ε_p. Therefore, no matter which one of these parameters is determined analytically or numerically, the exponent of h is the rate of convergence.

5.11 CONCLUDING REMARKS

In this chapter the fundamental concepts of error, local truncation error and Taylor series are used as the basis for deriving finite difference algorithms. The minimum requirement is that stability and consistency be satisfied. If additional restrictions are met, then a higher rate of convergence can be achieved. The possibilities of a nonuniform mesh and variable coefficients are included as is the condition in which the coefficients are discontinuous. The advection-diffusion problem and the need for a modification to a conventional finite difference algorithm, such as that provided by upwind differencing, is introduced. For idealized conditions, the connection between the local truncation error and convergence are made. Finally, the approach for determining numerically the rate of convergence is given.

Although the model problem considered is only one dimensional, the concepts introduced in this chapter are fundamental and applicable to problems of any dimension. Therefore, it is worthwhile to use this opportunity to perform the exercises as a means for solidly implanting these concepts prior to the study of more complex problems. Ciarlet (1988) and Strikwerda (1989) are good sources for a rigorous treatment of many of the concepts at a similar level of presentation.

5.12 EXERCISES

1. What is meant by each of the following terms: (i) spatial discretization, (ii) stencil, (iii) approximating equation, (iv) error, (v) convergence, (vi) rate of convergence, (vii) local truncation error, (viii) upwind differencing.

2. Derive Equation (5.3-18).

3. Write a finite difference program for solving the model problem with k, *a* and b as prescribed constants and an arbitrary function f(x) for Dirichlet and Neumann boundary conditions. Verify your program works by comparing analytical and numerical solutions for a set of problems of your choice.

4. For one of the problems chosen in Exercise 3, determine numerically the rate of convergence.

5. For the advection-diffusion problem, show that the algorithm of (5.6-2) yields an oscillatory solution for a coarse mesh but yet displays convergence with mesh refinement. Show that the change to the upwind scheme of (5.6-7) removes the oscillations.

6. The objective of this exercise is to verify the expression for the eigenvalues of a symmetric tridiagonal matrix used in Section 5.9.

   (i) Suppose [A] is a square matrix with eigenvalues λ_j and eigenvectors $\{e\}_j$. Let c and d be scalars. Show that the eigenvalues and eigenvectors of c[A] + d[I] are $c\lambda_j + d$ and $\{e\}_j$, respectively.

   (ii) Let [A] be an nxn symmetric tridiagonal matrix with zeros on the diagonal and each nonzero offdiagonal term is one. Prove that the j'th eigenvalue is $\lambda_j = 2\cos(\frac{j\pi}{n+1})$ and the i'th component of the j'th eigenvector is $(x_{(i)})_j = \sin(\frac{ij\pi}{n+1})$ with i, j = 1, ..., n.

   (iii) Use (i) and (ii) to prove that if [A] is a tridiagonal matrix with each diagonal term equal to β and each nonzero offdiagonal term equal to α, then the eigenvalues of [A] are

$$\lambda_j = \beta + 2\alpha\cos(\frac{j\pi}{n+1}), \qquad j = 1, ..., n$$