Chapter 2

Steady States and Boundary Value Problems

We will first consider ordinary differential equations (ODEs) that are posed on some interval $a < x < b$, together with some boundary conditions at each end of the interval. In the next chapter we will extend this to more than one space dimension and will study *elliptic partial differential equations* (PDEs) that are posed in some region of the plane or three-dimensional space and are solved subject to some boundary conditions specifying the solution and/or its derivatives around the boundary of the region. The problems considered in these two chapters are generally *steady-state* problems in which the solution varies only with the spatial coordinates but not with time. (But see Section 2.16 for a case where $[a,b]$ is a time interval rather than an interval in space.)

Steady-state problems are often associated with some time-dependent problem that describes the dynamic behavior, and the 2-point boundary value problem (BVP) or elliptic equation results from considering the special case where the solution is steady in time, and hence the time-derivative terms are equal to zero, simplifying the equations.

2.1 The heat equation

As a specific example, consider the flow of heat in a rod made out of some heat-conducting material, subject to some external heat source along its length and some boundary conditions at each end. If we assume that the material properties, the initial temperature distribution, and the source vary only with x, the distance along the length, and not across any cross section, then we expect the temperature distribution at any time to vary only with x and we can model this with a differential equation in one space dimension. Since the solution might vary with time, we let $u(x,t)$ denote the temperature at point x at time t, where $a < x < b$ along some finite length of the rod. The solution is then governed by the *heat equation*

$$u_t(x,t) = (\kappa(x)u_x(x,t))_x + \psi(x,t), \tag{2.1}$$

where $\kappa(x)$ is the coefficient of heat conduction, which may vary with x, and $\psi(x,t)$ is the heat source (or sink, if $\psi < 0$). See Appendix E for more discussion and a derivation. Equation (2.1) is often called the *diffusion equation* since it models diffusion processes more generally, and the diffusion of heat is just one example. It is assumed that the basic

theory of this equation is familiar to the reader. See standard PDE books such as [55] for a derivation and more introduction. In general it is extremely valuable to understand where the equation one is attempting to solve comes from, since a good understanding of the physics (or biology, etc.) is generally essential in understanding the development and behavior of numerical methods for solving the equation.

2.2 Boundary conditions

If the material is homogeneous, then $\kappa(x) \equiv \kappa$ is independent of x and the heat equation (2.1) reduces to

$$u_t(x,t) = \kappa u_{xx}(x,t) + \psi(x,t). \tag{2.2}$$

Along with the equation, we need initial conditions,

$$u(x,0) = u^0(x),$$

and boundary conditions, for example, the temperature might be specified at each end,

$$u(a,t) = \alpha(t), \qquad u(b,t) = \beta(t). \tag{2.3}$$

Such boundary conditions, where the value of the solution itself is specified, are called *Dirichlet boundary conditions*. Alternatively one end, or both ends, might be insulated, in which case there is zero heat flux at that end, and so $u_x = 0$ at that point. This boundary condition, which is a condition on the derivative of u rather than on u itself, is called a *Neumann boundary condition*. To begin, we will consider the Dirichlet problem for (2.2) with boundary conditions (2.3).

2.3 The steady-state problem

In general we expect the temperature distribution to change with time. However, if $\psi(x,t)$, $\alpha(t)$, and $\beta(t)$ are all time independent, then we might expect the solution to eventually reach a *steady-state* solution $u(x)$, which then remains essentially unchanged at later times. Typically there will be an initial *transient* time, as the initial data $u^0(x)$ approach $u(x)$ (unless $u^0(x) \equiv u(x)$), but if we are interested only in computing the steady-state solution itself, then we can set $u_t = 0$ in (2.2) and obtain an ODE in x to solve for $u(x)$:

$$u''(x) = f(x), \tag{2.4}$$

where we introduce $f(x) = -\psi(x)/\kappa$ to avoid minus signs below. This is a second order ODE, and from basic theory we expect to need two boundary conditions to specify a unique solution. In our case we have the boundary conditions

$$u(a) = \alpha, \qquad u(b) = \beta. \tag{2.5}$$

Remark: Having two boundary conditions does not necessarily guarantee that there exists a unique solution for a general second order equation—see Section 2.13.

The problem (2.4), (2.5) is called a *2-point* (BVP), since one condition is specified at each of the two endpoints of the interval where the solution is desired. If instead two data

values were specified at the same point, say, $u(a) = \alpha, u'(a) = \sigma$, and we want to find the solution for $t \geq a$, then we would have an *initial value problem* (IVP) instead. These problems are discussed in Chapter 5.

One approach to computing a numerical solution to a steady-state problem is to choose some initial data and march forward in time using a numerical method for the time-dependent PDE (2.2), as discussed in Chapter 9 on the solution of parabolic equations. However, this is typically not an efficient way to compute the steady state solution if this is all we want. Instead we can discretize and solve the 2-point BVP given by (2.4) and (2.5) directly. This is the first BVP that we will study in detail, starting in the next section. Later in this chapter we will consider some other BVPs, including more challenging nonlinear equations.

2.4 A simple finite difference method

As a first example of a finite difference method for solving a differential equation, consider the second order ODE discussed above,

$$u''(x) = f(x) \quad \text{for } 0 < x < 1, \tag{2.6}$$

with some given boundary conditions

$$u(0) = \alpha, \qquad u(1) = \beta. \tag{2.7}$$

The function $f(x)$ is specified and we wish to determine $u(x)$ in the interval $0 < x < 1$. This problem is called a *2-point* BVP since boundary conditions are given at two distinct points. This problem is so simple that we can solve it explicitly (integrate $f(x)$ twice and choose the two constants of integration so that the boundary conditions are satisfied), but studying finite difference methods for this simple equation will reveal some of the essential features of all such analysis, particularly the relation of the global error to the local truncation error and the use of stability in making this connection.

We will attempt to compute a grid function consisting of values $U_0, U_1, \ldots, U_m, U_{m+1}$, where U_j is our approximation to the solution $u(x_j)$. Here $x_j = jh$ and $h = 1/(m+1)$ is the *mesh width*, the distance between grid points. From the boundary conditions we know that $U_0 = \alpha$ and $U_{m+1} = \beta$, and so we have m unknown values $U_1, \ldots, U_m$ to compute. If we replace $u''(x)$ in (2.6) by the centered difference approximation

$$D^2 U_j = \frac{1}{h^2}(U_{j-1} - 2U_j + U_{j+1}),$$

then we obtain a set of algebraic equations

$$\frac{1}{h^2}(U_{j-1} - 2U_j + U_{j+1}) = f(x_j) \quad \text{for } j = 1, 2, \ldots, m. \tag{2.8}$$

Note that the first equation ($j = 1$) involves the value $U_0 = \alpha$ and the last equation ($j = m$) involves the value $U_{m+1} = \beta$. We have a linear system of m equations for the m unknowns, which can be written in the form

$$AU = F, \tag{2.9}$$

where U is the vector of unknowns $U = [U_1, U_2, \ldots, U_m]^T$ and

$$A = \frac{1}{h^2}\begin{bmatrix} -2 & 1 & & & & & \\ 1 & -2 & 1 & & & & \\ & 1 & -2 & 1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & 1 & -2 & 1 & \\ & & & & 1 & -2 \end{bmatrix}, \qquad F = \begin{bmatrix} f(x_1) - \alpha/h^2 \\ f(x_2) \\ f(x_3) \\ \vdots \\ f(x_{m-1}) \\ f(x_m) - \beta/h^2 \end{bmatrix}. \quad (2.10)$$

This tridiagonal linear system is nonsingular and can be easily solved for U from any right-hand side F.

How well does U approximate the function $u(x)$? We know that the centered difference approximation D^2, when applied to a known smooth function $u(x)$, gives a second order accurate approximation to $u''(x)$. But here we are doing something more complicated—we know the values of u'' at each point and are computing a whole set of discrete values $U_1, \ldots, U_m$ with the property that applying D^2 to these discrete values gives the desired values $f(x_j)$. While we might hope that this process also gives errors that are $O(h^2)$ (and indeed it does), this is certainly not obvious.

First we must clarify what we mean by the error in the discrete values $U_1, \ldots, U_m$ relative to the true solution $u(x)$, which is a function. Since U_j is supposed to approximate $u(x_j)$, it is natural to use the pointwise errors $U_j - u(x_j)$. If we let $\hat{U}$ be the vector of true values

$$\hat{U} = \begin{bmatrix} u(x_1) \\ u(x_2) \\ \vdots \\ u(x_m) \end{bmatrix}, \quad (2.11)$$

then the error vector E defined by

$$E = U - \hat{U}$$

contains the errors at each grid point.

Our goal is now to obtain a bound on the magnitude of this vector, showing that it is $O(h^2)$ as $h \to 0$. To measure the magnitude of this vector we must use some *norm*, for example, the max-norm

$$\|E\|_\infty = \max_{1 \le j \le m} |E_j| = \max_{1 \le j \le m} |U_j - u(x_j)|.$$

This is just the largest error over the interval. If we can show that $\|E\|_\infty = O(h^2)$, then it follows that each pointwise error must be $O(h^2)$ as well.

Other norms are often used to measure grid functions, either because they are more appropriate for a given problem or simply because they are easier to bound since some mathematical techniques work only with a particular norm. Other norms that are frequently used include the 1-norm

$$\|E\|_1 = h \sum_{j=1}^{m} |E_j|$$

and the 2-norm

$$\|E\|_2 = \left(h \sum_{j=1}^{m} |E_j|^2 \right)^{1/2}.$$

Note the factor of h that appears in these definitions. See Appendix A for a more thorough discussion of grid function norms and how they relate to standard vector norms.

Now let's return to the problem of estimating the error in our finite difference solution to BVP obtained by solving the system (2.9). The technique we will use is absolutely basic to the analysis of finite difference methods in general. It involves two key steps. We first compute the *local truncation error* (LTE) of the method and then use some form of *stability* to show that the *global error* can be bounded in terms of the LTE.

The global error simply refers to the error $U - \hat{U}$ that we are attempting to bound. The LTE refers to the error in our finite difference approximation of derivatives and hence is something that can be easily estimated using Taylor series expansions, as we have seen in Chapter 1. Stability is the magic ingredient that allows us to go from these easily computed bounds on the local error to the estimates we really want for the global error. Let's look at each of these in turn.

2.5 Local truncation error

The LTE is defined by replacing U_j with the true solution $u(x_j)$ in the finite difference formula (2.8). In general the true solution $u(x_j)$ won't satisfy this equation exactly and the discrepancy is the LTE, which we denote by τ_j:

$$\tau_j = \frac{1}{h^2}(u(x_{j-1}) - 2u(x_j) + u(x_{j+1})) - f(x_j) \tag{2.12}$$

for $j = 1, 2, \ldots, m$. Of course in practice we don't know what the true solution $u(x)$ is, but if we assume it is smooth, then by the Taylor series expansions (1.5a) we know that

$$\tau_j = \left[u''(x_j) + \frac{1}{12} h^2 u''''(x_j) + O(h^4) \right] - f(x_j). \tag{2.13}$$

Using our original differential equation (2.6) this becomes

$$\tau_j = \frac{1}{12} h^2 u''''(x_j) + O(h^4).$$

Although u'''' is in general unknown, it is some fixed function independent of h, and so $\tau_j = O(h^2)$ as $h \to 0$.

If we define τ to be the vector with components τ_j, then

$$\tau = A\hat{U} - F,$$

where $\hat{U}$ is the vector of true solution values (2.11), and so

$$A\hat{U} = F + \tau. \tag{2.14}$$

2.6 Global error

To obtain a relation between the local error τ and the global error $E = U - \hat{U}$, we subtract (2.14) from (2.9) that defines U, obtaining

$$AE = -\tau. \tag{2.15}$$

This is simply the matrix form of the system of equations

$$\frac{1}{h^2}(E_{j-1} - 2E_j + E_{j+1}) = -\tau(x_j) \quad \text{for } j = 1, 2, \ldots, m$$

with the boundary conditions

$$E_0 = E_{m+1} = 0$$

since we are using the exact boundary data $U_0 = \alpha$ and $U_{m+1} = \beta$. We see that the global error satisfies a set of finite difference equations that has exactly the same form as our original difference equations for U except that the right-hand side is given by $-\tau$ rather than F.

From this it should be clear why we expect the global error to be roughly the same magnitude as the local error τ. We can interpret the system (2.15) as a discretization of the ODE

$$e''(x) = -\tau(x) \quad \text{for } 0 < x < 1 \tag{2.16}$$

with boundary conditions

$$e(0) = 0, \quad e(1) = 0.$$

Since $\tau(x) \approx \frac{1}{12}h^2 u''''(x)$, integrating twice shows that the global error should be roughly

$$e(x) \approx -\frac{1}{12}h^2 u''(x) + \frac{1}{12}h^2 \left(u''(0) + x(u''(1) - u''(0))\right)$$

and hence the error should be $O(h^2)$.

2.7 Stability

The above argument is not completely convincing because we are relying on the assumption that solving the difference equations gives a decent approximation to the solution of the underlying differential equations (actually the converse now, that the solution to the differential equation (2.16) gives a good indication of the solution to the difference equations (2.15)). Since it is exactly this assumption we are trying to prove, the reasoning is rather circular.

Instead, let's look directly at the discrete system (2.15), which we will rewrite in the form

$$A^h E^h = -\tau^h, \tag{2.17}$$

where the superscript h indicates that we are on a grid with mesh spacing h. This serves as a reminder that these quantities change as we refine the grid. In particular, the matrix A^h is an $m \times m$ matrix with $h = 1/(m+1)$ so that its dimension is growing as $h \to 0$.

Let $(A^h)^{-1}$ be the inverse of this matrix. Then solving the system (2.17) gives

$$E^h = -(A^h)^{-1}\tau^h$$

and taking norms gives

$$\begin{aligned}\|E^h\| &= \|(A^h)^{-1}\tau^h\| \\ &\le \|(A^h)^{-1}\|\,\|\tau^h\|.\end{aligned}$$

We know that $\|\tau^h\| = O(h^2)$ and we are hoping the same will be true of $\|E^h\|$. It is clear what we need for this to be true: we need $\|(A^h)^{-1}\|$ to be bounded by some constant independent of h as $h \to 0$:

$$\|(A^h)^{-1}\| \le C \quad \text{for all } h \text{ sufficiently small.}$$

Then we will have

$$\|E^h\| \le C\|\tau^h\| \tag{2.18}$$

and so $\|E^h\|$ goes to zero at least as fast as $\|\tau^h\|$. This motivates the following definition of *stability* for linear BVPs.

Definition 2.1. *Suppose a finite difference method for a linear BVP gives a sequence of matrix equations of the form $A^hU^h = F^h$, where h is the mesh width. We say that the method is* stable *if $(A^h)^{-1}$ exists for all h sufficiently small (for $h < h_0$, say) and if there is a constant C, independent of h, such that*

$$\|(A^h)^{-1}\| \le C \quad \textit{for all } h < h_0. \tag{2.19}$$

2.8 Consistency

We say that a method is *consistent* with the differential equation and boundary conditions if

$$\|\tau^h\| \to 0 \quad \text{as } h \to 0. \tag{2.20}$$

This simply says that we have a sensible discretization of the problem. Typically $\|\tau^h\| = O(h^p)$ for some integer $p > 0$, and then the method is certainly consistent.

2.9 Convergence

A method is said to be *convergent* if $\|E^h\| \to 0$ as $h \to 0$. Combining the ideas introduced above we arrive at the conclusion that

$$\text{consistency} \;+\; \text{stability} \quad \Longrightarrow \quad \text{convergence}. \tag{2.21}$$

This is easily proved by using (2.19) and (2.20) to obtain the bound

$$\|E^h\| \le \|(A^h)^{-1}\|\,\|\tau^h\| \le C\|\tau^h\| \to 0 \quad \text{as } h \to 0.$$

Although this has been demonstrated only for the linear BVP, in fact most analyses of finite difference methods for differential equations follow this same two-tier approach, and the statement (2.21) is sometimes called the *fundamental theorem of finite difference methods*. In fact, as our above analysis indicates, this can generally be strengthened to say that

$$O(h^p) \text{ local truncation error} \quad + \quad \text{stability} \quad \Longrightarrow \quad O(h^p) \text{ global error.} \tag{2.22}$$

Consistency (and the order of accuracy) is usually the easy part to check. Verifying stability is the hard part. Even for the linear BVP just discussed it is not at all clear how to check the condition (2.19) since these matrices become larger as $h \to 0$. For other problems it may not even be clear how to define stability in an appropriate way. As we will see, there are many definitions of "stability" for different types of problems. The challenge in analyzing finite difference methods for new classes of problems often is to find an appropriate definition of "stability" that allows one to prove convergence using (2.21) while at the same time being sufficiently manageable that we can verify it holds for specific finite difference methods. For nonlinear PDEs this frequently must be tuned to each particular class of problems and relies on existing mathematical theory and techniques of analysis for this class of problems.

Whether or not one has a formal proof of convergence for a given method, it is always good practice to check that the computer program is giving convergent behavior, at the rate expected. Appendix A contains a discussion of how the error in computed results can be estimated.

2.10 Stability in the 2-norm

Returning to the BVP at the start of the chapter, let's see how we can verify stability and hence second order accuracy. The technique used depends on what norm we wish to consider. Here we will consider the 2-norm and see that we can show stability by explicitly computing the eigenvectors and eigenvalues of the matrix A. In Section 2.11 we show stability in the max-norm by different techniques.

Since the matrix A from (2.10) is symmetric, the 2-norm of A is equal to its spectral radius (see Section A.3.2 and Section C.9):

$$\|A\|_2 = \rho(A) = \max_{1 \le p \le m} |\lambda_p|.$$

(Note that λ_p refers to the pth eigenvalue of the matrix. Superscripts are used to index the eigenvalues and eigenvectors, while subscripts on the eigenvector below refer to components of the vector.)

The matrix A^{-1} is also symmetric, and the eigenvalues of A^{-1} are simply the inverses of the eigenvalues of A, so

$$\|A^{-1}\|_2 = \rho(A^{-1}) = \max_{1 \le p \le m} |(\lambda_p)^{-1}| = \left(\min_{1 \le p \le m} |\lambda_p| \right)^{-1}.$$

So all we need to do is compute the eigenvalues of A and show that they are bounded away from zero as $h \to 0$. Of course we have an infinite set of matrices A^h to consider,

as h varies, but since the structure of these matrices is so simple, we can obtain a general expression for the eigenvalues of each A^h. For more complicated problems we might not be able to do this, but it is worth going through in detail for this problem because one often considers model problems for which such an analysis is possible. We will also need to know these eigenvalues for other purposes when we discuss parabolic equations in Chapter 9. (See also Section C.7 for more general expressions for the eigenvalues of related matrices.)

We will now focus on one particular value of $h = 1/(m+1)$ and drop the superscript h to simplify the notation. Then the m eigenvalues of A are given by

$$\lambda_p = \frac{2}{h^2}(\cos(p\pi h) - 1) \quad \text{for } p = 1, 2, \ldots, m. \tag{2.23}$$

The eigenvector u^p corresponding to λ_p has components u_j^p for $j = 1, 2, \ldots, m$ given by

$$u_j^p = \sin(p\pi jh). \tag{2.24}$$

This can be verified by checking that $Au^p = \lambda_p u^p$. The jth component of the vector Au^p is

$$\begin{aligned}
(Au^p)_j &= \frac{1}{h^2}\left(u_{j-1}^p - 2u_j^p + u_{j+1}^p\right) \\
&= \frac{1}{h^2}(\sin(p\pi(j-1)h) - 2\sin(p\pi jh) + \sin(p\pi(j+1)h)) \\
&= \frac{1}{h^2}(\sin(p\pi jh)\cos(p\pi h) - 2\sin(p\pi jh) + \sin(p\pi jh)\cos(p\pi h)) \\
&= \lambda_p u_j^p.
\end{aligned}$$

Note that for $j = 1$ and $j = m$ the jth component of Au^p looks slightly different (the u_{j-1}^p or u_{j+1}^p term is missing) but that the above form and trigonometric manipulations are still valid provided that we define

$$u_0^p = u_{m+1}^p = 0,$$

as is consistent with (2.24). From (2.23) we see that the smallest eigenvalue of A (in magnitude) is

$$\begin{aligned}
\lambda_1 &= \frac{2}{h^2}(\cos(\pi h) - 1) \\
&= \frac{2}{h^2}\left(-\frac{1}{2}\pi^2 h^2 + \frac{1}{24}\pi^4 h^4 + O(h^6)\right) \\
&= -\pi^2 + O(h^2).
\end{aligned}$$

This is clearly bounded away from zero as $h \to 0$, and so we see that the method is stable in the 2-norm. Moreover we get an error bound from this:

$$\|E^h\|_2 \le \|(A^h)^{-1}\|_2 \|\tau^h\|_2 \approx \frac{1}{\pi^2}\|\tau^h\|_2.$$

Since $\tau_j^h \approx \frac{1}{12}h^2 u''''(x_j)$, we expect $\|\tau^h\|_2 \approx \frac{1}{12}h^2\|u''''\|_2 = \frac{1}{12}h^2\|f''\|_2$. The 2-norm of the function f'' here means the grid-function norm of this function evaluated at the discrete points x_j, although this is approximately equal to the function space norm of f'' defined using (A.14).

Note that the eigenvector (2.24) is closely related to the eigenfunction of the corresponding differential operator $\frac{\partial^2}{\partial x^2}$. The functions

$$u^p(x) = \sin(p\pi x), \qquad p = 1,\ 2,\ 3,\ \ldots,$$

satisfy the relation

$$\frac{\partial^2}{\partial x^2}u^p(x) = \mu_p u^p(x)$$

with eigenvalue $\mu_p = -p^2\pi^2$. These functions also satisfy $u^p(0) = u^p(1) = 0$, and hence they are eigenfunctions of $\frac{\partial^2}{\partial x^2}$ on $[0, 1]$ with homogeneous boundary conditions. The discrete approximation to this operator given by the matrix A has only m eigenvalues instead of an infinite number, and the corresponding eigenvectors (2.24) are simply the first m eigenfunctions of $\frac{\partial^2}{\partial x^2}$ evaluated at the grid points. The eigenvalue λ_p is not exactly the same as μ_p, but at least for small values of p it is very nearly the same, since Taylor series expansion of the cosine in (2.23) gives

$$\begin{aligned}\lambda_p &= \frac{2}{h^2}\left(-\frac{1}{2}p^2\pi^2h^2 + \frac{1}{24}p^4\pi^4h^4 + \cdots\right)\\ &= -p^2\pi^2 + O(h^2) \quad \text{as } h \to 0 \text{ for } p \text{ fixed.}\end{aligned}$$

This relationship will be illustrated further when we study numerical methods for the heat equation (2.1).

2.11 Green's functions and max-norm stability

In Section 2.10 we demonstrated that A from (2.10) is stable in the 2-norm, and hence that $\|E\|_2 = O(h^2)$. Suppose, however, that we want a bound on the maximum error over the interval, i.e., a bound on $\|E\|_\infty = \max|E_j|$. We can obtain one such bound directly from the bound we have for the 2-norm. From (A.19) we know that

$$\|E\|_\infty \le \frac{1}{\sqrt{h}}\|E\|_2 = O(h^{3/2}) \quad \text{as } h \to 0.$$

However, this does not show the second order accuracy that we hope to have. To show that $\|E\|_\infty = O(h^2)$ we will explicitly calculate the inverse of A and then show that $\|A^{-1}\|_\infty = O(1)$, and hence

$$\|E\|_\infty \le \|A^{-1}\|_\infty\|\tau\|_\infty = O(h^2)$$

since $\|\tau\|_\infty = O(h^2)$. As in the computation of the eigenvalues in the last section, we can do this only because our model problem (2.6) is so simple. In general it would be impossible to obtain closed form expressions for the inverse of the matrices A^h as h varies.

But again it is worth working out the details for this simple case because it gives a great deal of insight into the nature of the inverse matrix and what it represents more generally.

Each column of the inverse matrix can be interpreted as the solution of a particular BVP. The columns are discrete approximations to the *Green's functions* that are commonly introduced in the study of the differential equation. An understanding of this is valuable in developing an intuition for what happens if we introduce relatively large errors at a few points within the interval. Such difficulties arise frequently in practice, typically at the boundary or at an internal interface where there are discontinuities in the data or solution.

We begin by reviewing the Green's function solution to the BVP

$$u''(x) = f(x) \qquad \text{for } 0 < x < 1 \tag{2.25}$$

with Dirichlet boundary conditions

$$u(0) = \alpha, \qquad u(1) = \beta. \tag{2.26}$$

To keep the expressions simple below we assume we are on the unit interval, but everything can be shifted to an arbitrary interval $[a, b]$.

For any fixed point $\bar{x} \in [0, 1]$, the Green's function $G(x; \bar{x})$ is the function of x that solves the particular BVP of the above form with $f(x) = \delta(x - \bar{x})$ and $\alpha = \beta = 0$. Here $\delta(x - \bar{x})$ is the "delta function" centered at $\bar{x}$. The delta function, $\delta(x)$, is not an ordinary function but rather the mathematical idealization of a sharply peaked function that is nonzero only on an interval $(-\epsilon, \epsilon)$ near the origin and has the property that

$$\int_{-\infty}^{\infty} \phi_\epsilon(x)\, dx = \int_{-\epsilon}^{\epsilon} \phi_\epsilon(x)\, dx = 1. \tag{2.27}$$

For example, we might take

$$\phi_\epsilon(x) = \begin{cases} (\epsilon + x)/\epsilon & \text{if } -\epsilon \le x \le 0, \\ (\epsilon - x)/\epsilon & \text{if } 0 \le x \le \epsilon, \\ 0 & \text{otherwise.} \end{cases} \tag{2.28}$$

This piecewise linear function is the "hat function" with width ϵ and height $1/\epsilon$. The exact shape of ϕ_ϵ is not important, but note that it must attain a height that is $O(1/\epsilon)$ in order for the integral to have the value 1. We can think of the delta function as being a sort of limiting case of such functions as $\epsilon \to 0$. Delta functions naturally arise when we differentiate functions that are discontinuous. For example, consider the *Heaviside function* (or step function) $H(x)$ that is defined by

$$H(x) = \begin{cases} 0 & x < 0, \\ 1 & x \ge 0. \end{cases} \tag{2.29}$$

What is the derivative of this function? For $x \neq 0$ the function is constant and so $H'(x) = 0$. At $x = 0$ the derivative is not defined in the classical sense. But if we smooth out the function a little bit, making it continuous and differentiable by changing $H(x)$ only on the interval $(-\epsilon, \epsilon)$, then the new function $H_\epsilon(x)$ is differentiable everywhere and has a

derivative $H'_\epsilon(x)$ that looks something like $\phi_\epsilon(x)$. The exact shape of $H'_\epsilon(x)$ depends on how we choose $H_\epsilon(x)$, but note that regardless of its shape, its integral must be 1, since

$$\begin{aligned}\int_{-\infty}^{\infty} H'_\epsilon(x)\,dx &= \int_{-\epsilon}^{\epsilon} H'_\epsilon(x)\,dx \\ &= H_\epsilon(\epsilon) - H_\epsilon(-\epsilon) \\ &= 1 - 0 = 1.\end{aligned}$$

This explains the normalization (2.27). By letting $\epsilon \to 0$, we are led to define

$$H'(x) = \delta(x).$$

This expression makes no sense in terms of the classical definition of derivatives, but it can be made rigorous mathematically through the use of "distribution theory"; see, for example, [31]. For our purposes it suffices to think of the delta function as being a very sharply peaked function that is nonzero only on a very narrow interval but with total integral 1.

If we interpret the problem (2.25) as a steady-state heat conduction problem with source $\psi(x) = -f(x)$, then setting $f(x) = \delta(x - \bar{x})$ in the BVP is the mathematical idealization of a heat sink that has unit magnitude but that is concentrated near a single point. It might be easier to first consider the case $f(x) = -\delta(x - \bar{x})$, which corresponds to a heat source localized at $\bar{x}$, the idealization of a blow torch pumping heat into the rod at a single point. With the boundary conditions $u(0) = u(1) = 0$, holding the temperature fixed at each end, we would expect the temperature to be highest at the point $\bar{x}$ and to fall linearly to zero to each side (linearly because $u''(x) = 0$ away from $\bar{x}$). With $f(x) = \delta(x - \bar{x})$, a heat sink at $\bar{x}$, we instead have the minimum temperature at $\bar{x}$, rising linearly to each side, as shown in Figure 2.1. This figure shows a typical Green's function $G(x; \bar{x})$ for one particular choice of $\bar{x}$. To complete the definition of this function we need to know the value $G(\bar{x}; \bar{x})$ that it takes at the minimum. This value is determined by the fact that the jump in slope at this point must be 1, since

$$\begin{aligned}u'(\bar{x} + \epsilon) - u'(\bar{x} - \epsilon) &= \int_{\bar{x}-\epsilon}^{\bar{x}+\epsilon} u''(x)\,dx \\ &= \int_{\bar{x}-\epsilon}^{\bar{x}+\epsilon} \delta(x - \bar{x})\,dx \\ &= 1.\end{aligned} \tag{2.30}$$

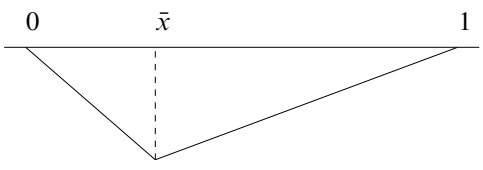

Figure 2.1. *The Green's function $G(x; \bar{x})$ from* (2.31).

A little algebra shows that the piecewise linear function $G(x;\bar{x})$ is given by

$$G(x;\bar{x}) = \begin{cases} (\bar{x}-1)x & \text{for } 0 \le x \le \bar{x}, \\ \bar{x}(x-1) & \text{for } \bar{x} \le x \le 1. \end{cases} \tag{2.31}$$

Note that by linearity, if we replaced $f(x)$ with $c\delta(x-\bar{x})$ for any constant c, the solution to the BVP would be $cG(x;\bar{x})$. Moreover, any linear combination of Green's functions at different points $\bar{x}$ is a solution to the BVP with the corresponding linear combination of delta functions on the right-hand side. So if we want to solve

$$u''(x) = 3\delta(x-0.3) - 5\delta(x-0.7), \tag{2.32}$$

for example (with $u(0) = u(1) = 0$), the solution is simply

$$u(x) = 3G(x;0.3) - 5G(x;0.7). \tag{2.33}$$

This is a piecewise linear function with jumps in slope of magnitude 3 at $x = 0.3$ and -5 at $x = 0.7$. More generally, if the right-hand side is a sum of weighted delta functions at any number of points,

$$f(x) = \sum_{k=1}^{n} c_k \delta(x - x_k), \tag{2.34}$$

then the solution to the BVP is

$$u(x) = \sum_{k=1}^{n} c_k G(x;x_k). \tag{2.35}$$

Now consider a general source $f(x)$ that is not a discrete sum of delta functions. We can view this as a continuous distribution of point sources, with $f(\bar{x})$ being a density function for the weight assigned to the delta function at $\bar{x}$, i.e.,

$$f(x) = \int_0^1 f(\bar{x})\delta(x-\bar{x})\,d\bar{x}. \tag{2.36}$$

(Note that if we smear out δ to ϕ_ϵ, then the right-hand side becomes a weighted average of values of f very close to x.) This suggests that the solution to $u''(x) = f(x)$ (still with $u(0) = u(1) = 0$) is

$$u(x) = \int_0^1 f(\bar{x})G(x;\bar{x})\,d\bar{x}, \tag{2.37}$$

and indeed it is.

Now let's consider more general boundary conditions. Since each Green's function $G(x;\bar{x})$ satisfies the homogeneous boundary conditions $u(0) = u(1) = 0$, any linear combination does as well. To incorporate the effect of nonzero boundary conditions, we introduce two new functions $G_0(x)$ and $G_1(x)$ defined by the BVPs

$$G_0''(x) = 0, \quad G_0(0) = 1, \quad G_0(1) = 0 \tag{2.38}$$

and

$$G_1''(x) = 0, \quad G_1(0) = 0, \quad G_1(1) = 1. \tag{2.39}$$

The solutions are

$$\begin{aligned} G_0(x) &= 1 - x, \\ G_1(x) &= x. \end{aligned} \tag{2.40}$$

These functions give the temperature distribution for the heat conduction problem with the temperature held at 1 at one boundary and 0 at the other with no internal heat source. Adding a scalar multiple of $G_0(x)$ to the solution $u(x)$ of (2.37) will change the value of $u(0)$ without affecting $u''(x)$ or $u(1)$, so adding $\alpha G_0(x)$ will allow us to satisfy the boundary condition at $x = 0$, and similarly adding $\beta G_1(x)$ will give the desired boundary value at $x = 1$. The full solution to (2.25) with boundary conditions (2.26) is thus

$$u(x) = \alpha G_0(x) + \beta G_1(x) + \int_0^1 f(\bar{x}) G(x;\bar{x})\, d\bar{x}. \tag{2.41}$$

Note that using the formula (2.31), we can rewrite this as

$$u(x) = \left(\alpha - \int_0^x \bar{x} f(\bar{x})\, d\bar{x}\right)(1 - x) + \left(\beta + \int_x^1 (\bar{x} - 1) f(\bar{x})\, d\bar{x}\right) x. \tag{2.42}$$

Of course this simple BVP can also be solved simply by integrating the function f twice, and the solution (2.42) can be put in this same form using integration by parts. But for our current purposes it is the form (2.41) that is of interest, since it shows clearly how the effect of each boundary condition and the local source at each point feeds into the global solution. The values α, β, and $f(x)$ are the data for this linear differential equation and (2.41) writes the solution as a linear operator applied to this data, analogous to writing the solution to the linear system $AU = F$ as $U = A^{-1}F$.

We are finally ready to return to the study of the max-norm stability of the finite difference method, which will be based on explicitly determining the inverse matrix for the matrix arising in this discretization. We will work with a slightly different formulation of the linear algebra problem in which we view U_0 and U_{m+1} as additional "unknowns" in the problem and introduce two new equations in the system that simply state that $U_0 = \alpha$ and $U_{m+1} = \beta$. The modified system has the form $AU = F$, where now

$$A = \frac{1}{h^2}\begin{bmatrix} h^2 & 0 & & & & & \\ 1 & -2 & 1 & & & & \\ & 1 & -2 & 1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & 1 & -2 & 1 & \\ & & & & 1 & -2 & 1 \\ & & & & & 0 & h^2 \end{bmatrix}, \quad U = \begin{bmatrix} U_0 \\ U_1 \\ U_2 \\ \vdots \\ U_{m-1} \\ U_m \\ U_{m+1} \end{bmatrix}, \quad F = \begin{bmatrix} \alpha \\ f(x_1) \\ f(x_2) \\ \vdots \\ f(x_{m-1}) \\ f(x_m) \\ \beta \end{bmatrix}. \tag{2.43}$$

While we could work directly with the matrix A from (2.10), this reformulation has two advantages:

1. It separates the algebraic equations corresponding to the boundary conditions from the algebraic equations corresponding to the ODE $u''(x) = f(x)$. In the system (2.10), the first and last equations contain a mixture of ODE and boundary conditions. Separating these terms will make it clearer how the inverse of A relates to the Green's function representation of the true solution found above.

2. In the next section we will consider Neumann boundary conditions $u'(0) = \sigma$ in place of $u(0) = \alpha$. In this case the value U_0 really is unknown and our new formulation is easily extended to this case by replacing the first row of A with a discretization of this boundary condition.

Let B denote the $(m + 2) \times (m + 2)$ inverse of A from (2.43), $B = A^{-1}$. We will index the elements of B by B_{00} through $B_{m+1,m+1}$ in the obvious manner. Let B_j denote the jth column of B for $j = 0, 1, \ldots, m + 1$. Then

$$AB_j = e_j,$$

where e_j is the jth column of the identity matrix. We can view this as a linear system to be solved for B_j. Note that this linear system is simply the discretization of the BVP for a special choice of right-hand side F in which only one element of this vector is nonzero. This is exactly analogous to the manner in which the Green's function for the ODE is defined. The column B_0 corresponds to the problem with $\alpha = 1$, $f(x) = 0$, and $\beta = 0$, and so we expect B_0 to be a discrete approximation of the function $G_0(x)$. In fact, the first (i.e., $j = 0$) column of B has elements obtained by simply evaluating G_0 at the grid points,

$$B_{i0} = G_0(x_i) = 1 - x_i. \tag{2.44}$$

Since this is a linear function, the second difference operator applied at any point yields zero. Similarly, the last ($j = m + 1$) column of B has elements

$$B_{i,m+1} = G_1(x_i) = x_i. \tag{2.45}$$

The interior columns ($1 \le j \le m$) correspond to the Green's function for zero boundary conditions and the source concentrated at a single point, since $F_j = 1$ and $F_i = 0$ for $i \ne j$. Note that this is a discrete version of $h\delta(x - x_j)$ since as a grid function F is nonzero over an interval of length h but has value 1 there, and hence total mass h. Thus we expect that the column B_j will be a discrete approximation to the function $hG(x; x_j)$. In fact, it is easy to check that

$$B_{ij} = hG(x_i; x_j) = \begin{cases} h(x_j - 1)x_i, & i = 1, 2, \ldots, j, \\ h(x_i - 1)x_j, & i = j, j + 1, \ldots, m. \end{cases} \tag{2.46}$$

An arbitrary right-hand side F for the linear system can be written as

$$F = \alpha e_0 + \beta e_{m+1} + \sum_{j=1}^{m} f_j e_j, \tag{2.47}$$

and the solution $U = BF$ is

$$U = \alpha B_0 + \beta B_{m+1} + \sum_{j=1}^{m} f_j B_j \tag{2.48}$$

with elements

$$U_i = \alpha(1-x_i) + \beta x_i + h\sum_{j=1}^{m} f_j G(x_i; x_j). \tag{2.49}$$

This is the discrete analogue of (2.41).

In fact, something more is true: suppose we define a function $v(x)$ by

$$v(x) = \alpha(1-x) + \beta x + h\sum_{j=1}^{m} f_j G(x; x_j). \tag{2.50}$$

Then $U_i = v(x_i)$ and $v(x)$ is the piecewise linear function that interpolates the numerical solution. This function $v(x)$ is the exact solution to the BVP

$$v''(x) = h\sum_{j=1}^{m} f(x_j)\delta(x-x_j), \quad v(0) = \alpha, \quad v(1) = \beta. \tag{2.51}$$

Thus we can interpret the discrete solution as the exact solution to a modified problem in which the right-hand side $f(x)$ has been replaced by a finite sum of delta functions at the grid points x_j, with weights $hf(x_j) \approx \int_{x_{j-1/2}}^{x_{j+1/2}} f(x)\,dx$.

To verify max-norm stability of the numerical method, we must show that $\|B\|_\infty$ is uniformly bounded as $h \to 0$. The infinity norm of the matrix is given by

$$\|B\|_\infty = \max_{0\le i\le m+1} \sum_{j=0}^{m+1} |B_{ij}|,$$

the maximum row sum of elements in the matrix. Note that the first row of B has $B_{00} = 1$ and $B_{0j} = 0$ for $j > 0$, and hence row sum 1. Similarly the last row contains all zeros except for $B_{m+1,m+1} = 1$. The intermediate rows are dense and the first and last elements (from columns B_0 and B_{m+1}) are bounded by 1. The other m elements of each of these rows are all bounded by h from (2.46), and hence

$$\sum_{j=0}^{m+1} |B_{ij}| \le 1 + 1 + mh < 3$$

since $h = 1/(m+1)$. Every row sum is bounded by 3 at most, and so $\|A^{-1}\|_\infty < 3$ for all h, and stability is proved.

While it may seem like we've gone to a lot of trouble to prove stability, the explicit representation of the inverse matrix in terms of the Green's functions is a useful thing to have, and if it gives additional insight into the solution process. Note, however, that it would *not* be a good idea to use the explicit expressions for the elements of $B = A^{-1}$ to solve the linear system by computing $U = BF$. Since B is a dense matrix, doing this matrix-vector multiplication requires $O(m^2)$ operations. We are much better off solving the original system $AU = F$ by Gaussian elimination. Since A is tridiagonal, this requires only $O(m)$ operations.

The Green's function representation also clearly shows the effect that each local truncation error has on the global error. Recall that the global error E is related to the local truncation error by $AE = -\tau$. This continues to hold for our reformulation of the problem, where we now define τ_0 and τ_{m+1} as the errors in the imposed boundary conditions, which are typically zero for the Dirichlet problem. Solving this system gives $E = -B\tau$. If we did make an error in one of the boundary conditions, setting F_0 to $\alpha + \tau_0$, the effect on the global error would be $\tau_0 B_0$. The effect of this error is thus nonzero across the entire interval, decreasing linearly from the boundary where the error is made at the other end. Each truncation error τ_i for $1 \le i \le m$ in the difference approximation to $u''(x_i) = f(x_i)$ likewise has an effect on the global error everywhere, although the effect is largest at the grid point x_i, where it is $hG(x_i; x_i)\tau_i$, and decays linearly toward each end. Note that since $\tau_i = O(h^2)$, the contribution of this error to the global error at each point is only $O(h^3)$. However, since all m local errors contribute to the global error at each point, the total effect is $O(mh^3) = O(h^2)$.

As a final note on this topic, observe that we have also worked out the inverse of the original matrix A defined in (2.10). Because the first row of B consists of zeros beyond the first element, and the last row consists of zeros, except for the last element, it is easy to check that the inverse of the $m \times m$ matrix from (2.10) is the $m \times m$ central block of B consisting of B_{11} through B_{mm}. The max-norm of this matrix is bounded by 1 for all h, so our original formulation is stable as well.

2.12 Neumann boundary conditions

Now suppose that we have one or more Neumann boundary conditions instead of Dirichlet boundary conditions, meaning that a boundary condition on the derivative u' is given rather than a condition on the value of u itself. For example, in our heat conduction example we might have one end of the rod insulated so that there is no heat flux through this end, and hence $u' = 0$ there. More generally we might have heat flux at a specified rate giving $u' = \sigma$ at this boundary.

We will see in the next section that imposing Neumann boundary conditions at both ends gives an ill-posed problem that has either no solution or infinitely many solutions. In this section we consider (2.25) with one Neumann condition, say,

$$u'(0) = \sigma, \qquad u(1) = \beta. \tag{2.52}$$

Figure 2.2 shows the solution to this problem with $f(x) = e^x$, $\sigma = 0$, and $\beta = 0$ as one example.

To solve this problem numerically, we need to determine U_0 as one of the unknowns. If we use the formulation of (2.43), then the first row of the matrix A must be modified to model the boundary condition (2.52).

First approach. As a first try, we might use a one-sided expression for $u'(0)$, such as

$$\frac{U_1 - U_0}{h} = \sigma. \tag{2.53}$$

If we use this equation in place of the first line of the system (2.43), we obtain the following system of equations for the unknowns $U_0,\ U_1,\ \ldots, U_m,\ U_{m+1}$:

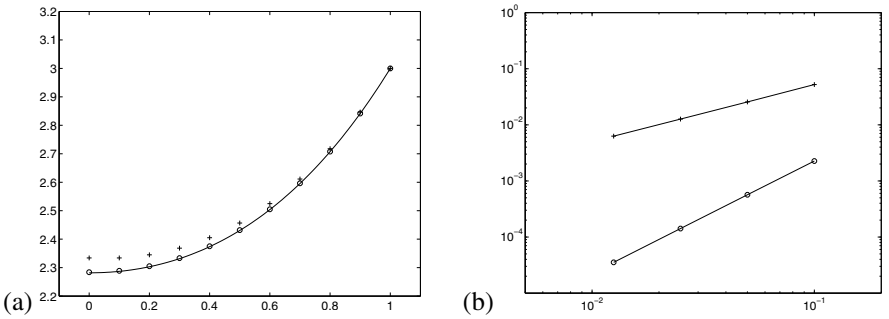

Figure 2.2. (a) *Sample solution to the steady-state heat equation with a Neumann boundary condition at the left boundary and Dirichlet at the right. The solid line is the true solution. The plus sign shows a solution on a grid with* 20 *points using* (2.53)*. The circle shows the solution on the same grid using* (2.55)*.* (b) *A log-log plot of the max-norm error as the grid is refined is also shown for each case.*

$$\frac{1}{h^2}\begin{bmatrix} -h & h & & & & & & & \\ 1 & -2 & 1 & & & & & & \\ & 1 & -2 & 1 & & & & & \\ & & 1 & -2 & 1 & & & & \\ & & & \ddots & \ddots & \ddots & & & \\ & & & & 1 & -2 & 1 & & \\ & & & & & 1 & -2 & 1 & \\ & & & & & & 0 & h^2 & \end{bmatrix}\begin{bmatrix} U_0 \\ U_1 \\ U_2 \\ U_3 \\ \vdots \\ U_{m-1} \\ U_m \\ U_{m+1} \end{bmatrix} = \begin{bmatrix} \sigma \\ f(x_1) \\ f(x_2) \\ f(x_3) \\ \vdots \\ f(x_{m-1}) \\ f(x_m) \\ \beta \end{bmatrix}. \tag{2.54}$$

Solving this system of equations does give an approximation to the true solution (see Figure 2.2), but checking the errors shows that this is only first order accurate. Figure 2.2 also shows a log-log plot of the max-norm errors as we refine the grid. The problem is that the local truncation error of the approximation (2.53) is $O(h)$, since

$$\begin{aligned} \tau_0 &= \frac{1}{h^2}(hu(x_1) - hu(x_0)) - \sigma \\ &= u'(x_0) + \frac{1}{2}hu''(x_0) + O(h^2) - \sigma \\ &= \frac{1}{2}hu''(x_0) + O(h^2). \end{aligned}$$

This translates into a global error that is only $O(h)$ as well.

Remark: It is sometimes possible to achieve second order accuracy even if the local truncation error is $O(h)$ at a single point, as long as it is $O(h^2)$ everywhere else. This is true here if we made an $O(h)$ truncation error at a single interior point, since the effect on the global error would be this $\tau_j B_j$, where B_j is the jth column of the appropriate inverse matrix. As in the Dirichlet case, this column is given by the corresponding Green's function scaled by h, and so the $O(h)$ local error would make an $O(h^2)$ contribution to the global error at each point. However, introducing an $O(h)$ error in τ_0 gives a contribution of $\tau_0 B_0$

to the global error, and as in the Dirichlet case this first column of B contains elements that are $O(1)$, resulting in an $O(h)$ contribution to the global error at every point.

Second approach. To obtain a second order accurate method, we can use a centered approximation to $u'(0) = \sigma$ instead of the one-sided approximation (2.53). We might introduce another unknown U_{-1} and, instead of the single equation (2.53), use the following two equations:

$$\begin{aligned} \frac{1}{h^2}(U_{-1} - 2U_0 + U_1) &= f(x_0), \\ \frac{1}{2h}(U_1 - U_{-1}) &= \sigma. \end{aligned} \tag{2.55}$$

This results in a system of $m + 3$ equations.

Introducing the unknown U_{-1} outside the interval $[0, 1]$ where the original problem is posed may seem unsatisfactory. We can avoid this by eliminating the unknown U_{-1} from the two equations (2.55), resulting in a single equation that can be written as

$$\frac{1}{h}(-U_0 + U_1) = \sigma + \frac{h}{2} f(x_0). \tag{2.56}$$

We have now reduced the system to one with only $m + 2$ equations for the unknowns $U_0, U_1, \ldots, U_{m+1}$. The matrix is exactly the same as the matrix in (2.54), which came from the one-sided approximation. The only difference in the linear system is that the first element in the right-hand side of (2.54) is now changed from σ to $\sigma + \frac{h}{2} f(x_0)$. We can interpret this as using the one-sided approximation to $u'(0)$, but with a modified value for this Neumann boundary condition that adjusts for the fact that the approximation has an $O(h)$ error by introducing the same error in the data σ.

Alternatively, we can view the left-hand side of (2.56) as a centered approximation to $u'(x_0 + h/2)$ and the right-hand side as the first two terms in the Taylor series expansion of this value,

$$u'\left(x_0 + \frac{h}{2}\right) = u'(x_0) + \frac{h}{2}u''(x_0) + \cdots = \sigma + \frac{h}{2} f(x_0) + \cdots .$$

Third approach. Rather than using a second order accurate centered approximation to the Neumann boundary condition, we could instead use a second order accurate one-sided approximation based on the three unknowns U_0, U_1, and U_2. An approximation of this form was derived in Example 1.2, and using this as the boundary condition gives the equation

$$\frac{1}{h}\left(\frac{3}{2}U_0 - 2U_1 + \frac{1}{2}U_2\right) = \sigma.$$

This results in the linear system

$$\frac{1}{h^2}\begin{bmatrix} \frac{3h}{2} & -2h & \frac{h}{2} & & & & & & \\ 1 & -2 & 1 & & & & & & \\ & 1 & -2 & 1 & & & & & \\ & & 1 & -2 & 1 & & & & \\ & & & \ddots & \ddots & \ddots & & & \\ & & & & 1 & -2 & 1 & & \\ & & & & & 1 & -2 & 1 & \\ & & & & & & 0 & h^2 \end{bmatrix} \begin{bmatrix} U_0 \\ U_1 \\ U_2 \\ U_3 \\ \vdots \\ U_{m-1} \\ U_m \\ U_{m+1} \end{bmatrix} = \begin{bmatrix} \sigma \\ f(x_1) \\ f(x_2) \\ f(x_3) \\ \vdots \\ f(x_{m-1}) \\ f(x_m) \\ \beta \end{bmatrix}. \quad (2.57)$$

This boundary condition is second order accurate from the error expression (1.12). The use of this equation slightly disturbs the tridiagonal structure but adds little to the cost of solving the system of equations and produces a second order accurate result. This approach is often the easiest to generalize to other situations, such as higher order accurate methods, nonuniform grids, or more complicated boundary conditions.

2.13 Existence and uniqueness

In trying to solve a mathematical problem by a numerical method, it is always a good idea to check that the original problem has a solution and in fact that it is *well posed* in the sense developed originally by Hadamard. This means that the problem should have a unique solution that depends continuously on the data used to define the problem. In this section we will show that even seemingly simple BVPs may fail to be well posed.

Consider the problem of Section 2.12 but now suppose we have Neumann boundary conditions at both ends, i.e., we have (2.6) with

$$u'(0) = \sigma_0, \qquad u'(1) = \sigma_1.$$

In this case the techniques of Section 2.12 would naturally lead us to the discrete system

$$\frac{1}{h^2}\begin{bmatrix} -h & h & & & & & & \\ 1 & -2 & 1 & & & & & \\ & 1 & -2 & 1 & & & & \\ & & 1 & -2 & 1 & & & \\ & & & \ddots & \ddots & \ddots & & \\ & & & & 1 & -2 & 1 & \\ & & & & & h & -h \end{bmatrix} \begin{bmatrix} U_0 \\ U_1 \\ U_2 \\ U_3 \\ \vdots \\ U_m \\ U_{m+1} \end{bmatrix} = \begin{bmatrix} \sigma_0 + \frac{h}{2} f(x_0) \\ f(x_1) \\ f(x_2) \\ f(x_3) \\ \vdots \\ f(x_m) \\ -\sigma_1 + \frac{h}{2} f(x_{m+1}) \end{bmatrix}. \quad (2.58)$$

If we try to solve this system, however, we will soon discover that the matrix is singular, and in general the system has no solution. (Or, if the right-hand side happens to lie in the range of the matrix, it has infinitely many solutions.) It is easy to verify that the matrix is singular by noting that the constant vector $e = [1, 1, \ldots, 1]^T$ is a null vector.

This is not a failure in our numerical model. In fact it reflects that the problem we are attempting to solve is not well posed, and the differential equation will also have either no solution or infinitely many solutions. This can be easily understood physically by again considering the underlying heat equation discussed in Section 2.1. First consider the case

where $\sigma_0 = \sigma_1 = 0$ and $f(x) \equiv 0$ so that both ends of the rod are insulated, there is no heat flux through the ends, and there is no heat source within the rod. Recall that the BVP is a simplified equation for finding the steady-state solution of the heat equation (2.2) with some initial data $u^0(x)$. How does $u(x,t)$ behave with time? In the case now being considered the total heat energy in the rod must be conserved with time, so $\int_0^1 u(x,t)\,dx \equiv \int_0^1 u^0(x)\,dx$ for all time. Diffusion of the heat tends to redistribute it until it is uniformly distributed throughout the rod, so we expect the steady state solution $u(x)$ to be constant in x,

$$u(x) = c, \tag{2.59}$$

where the constant c depends on the initial data $u^0(x)$. In fact, by conservation of energy, $c = \int_0^1 u^0(x)\,dx$ for our rod of unit length. But notice now that *any* constant function of the form (2.59) is a solution of the steady-state BVP, since it satisfies all the conditions $u''(x) \equiv 0$, $u'(0) = u'(1) = 0$. The ODE has infinitely many solutions in this case. The physical problem has only one solution, but in attempting to simplify it by solving for the steady state alone, we have thrown away a crucial piece of data, which is the heat content of the initial data for the heat equation. If at least one boundary condition is a Dirichlet condition, then it can be shown that the steady-state solution is *independent* of the initial data and we can solve the BVP uniquely, but not in the present case.

Now suppose that we have a source term $f(x)$ that is not identically zero, say, $f(x) < 0$ everywhere. Then we are constantly adding heat to the rod (recall that $f = -\psi$ in (2.4)). Since no heat can escape through the insulated ends, we expect the temperature to keep rising without bound. In this case we never reach a steady state, and the BVP has no solution. On the other hand, if f is positive over part of the interval and negative elsewhere, and the net effect of the heat sources and sinks exactly cancels out, then we expect that a steady state might exist. In fact, solving the BVP exactly by integrating twice and trying to determine the constants of integration from the boundary conditions shows that a solution exists (in the case of insulated boundaries) only if $\int_0^1 f(x)\,dx = 0$, in which case there are infinitely many solutions. If σ_0 and/or σ_1 are nonzero, then there is heat flow at the boundaries and the net heat source must cancel the boundary fluxes. Since

$$u'(1) = u'(0) + \int_0^1 u''(x)\,dx = \int_0^1 f(x)\,dx, \tag{2.60}$$

this requires

$$\int_0^1 f(x)\,dx = \sigma_1 - \sigma_0. \tag{2.61}$$

Similarly, the singular linear system (2.58) has a solution (in fact infinitely many solutions) only if the right-hand side F is orthogonal to the null space of A^T. This gives the condition

$$\frac{h}{2} f(x_0) + h \sum_{i=1}^{m} f(x_i) + \frac{h}{2} f(x_{m+1}) = \sigma_1 - \sigma_0, \tag{2.62}$$

which is the trapezoidal rule approximation to the condition (2.61).

2.14 Ordering the unknowns and equations

Note that in general we are always free to change the order of the equations in a linear system without changing the solution. Modifying the order corresponds to permuting the rows of the matrix and right-hand side. We are also free to change the ordering of the unknowns in the vector of unknowns, which corresponds to permuting the columns of the matrix. As an example, consider the difference equations given by (2.9). Suppose we reordered the unknowns by listing first the unknowns at odd numbered grid points and then the unknowns at even numbered grid points, so that $\tilde{U} = [U_1,\ U_3,\ U_5,\ \ldots, U_2,\ U_4,\ \ldots]^T$. If we also reorder the equations in the same way, i.e., we write down first the difference equation centered at U_1, then at U_3, U_5, etc., then we would obtain the following system:

$$\frac{1}{h^2}\left[\begin{array}{ccccccc|ccccccc}
-2 & & & & & & & 1 & & & & & & \\
 & -2 & & & & & & 1 & 1 & & & & & \\
 & & -2 & & & & & & 1 & 1 & & & & \\
 & & & \ddots & & & & & & \ddots & \ddots & & & \\
 & & & & & -2 & & & & & & 1 & 1 & \\
\hline
1 & 1 & & & & & & -2 & & & & & & \\
 & 1 & 1 & & & & & & -2 & & & & & \\
 & & 1 & 1 & & & & & & -2 & & & & \\
 & & & \ddots & \ddots & & & & & & \ddots & & & \\
 & & & & 1 & & & & & & & -2 &
\end{array}\right]\left[\begin{array}{c} U_1 \\ U_3 \\ U_5 \\ \vdots \\ U_{m-1} \\ \hline U_2 \\ U_4 \\ U_6 \\ \vdots \\ U_m \end{array}\right]$$

$$= \left[\begin{array}{c} f(x_1) - \alpha/h^2 \\ f(x_3) \\ f(x_5) \\ \vdots \\ f(x_{m-1}) \\ \hline f(x_2) \\ f(x_4) \\ f(x_6) \\ \vdots \\ f(x_m) - \beta/h^2 \end{array}\right]. \tag{2.63}$$

This linear system has the same solution as (2.9) modulo the reordering of unknowns, but it looks very different. For this one-dimensional problem there is no point in reordering things this way, and the natural ordering $[U_1,\ U_2,\ U_3,\ \ldots]^T$ clearly gives the optimal matrix structure for the purpose of applying Gaussian elimination. By ordering the unknowns so that those which occur in the same equation are close to one another in the vector, we keep the nonzeros in the matrix clustered near the diagonal. In two or three space dimensions there are more interesting consequences of choosing different orderings, a topic we return to in Section 3.3.

2.15 A general linear second order equation

We now consider the more general linear equation

$$a(x)u''(x) + b(x)u'(x) + c(x)u(x) = f(x), \tag{2.64}$$

together with two boundary conditions, say, the Dirichlet conditions

$$u(a) = \alpha, \qquad u(b) = \beta. \tag{2.65}$$

This equation can be discretized to second order by

$$a_i\left(\frac{U_{i-1} - 2U_i + U_{i+1}}{h^2}\right) + b_i\left(\frac{U_{i+1} - U_{i-1}}{2h}\right) + c_iU_i = f_i, \tag{2.66}$$

where, for example, $a_i = a(x_i)$. This gives the linear system $AU = F$, where A is the tridiagonal matrix

$$A = \frac{1}{h^2}\begin{bmatrix} (h^2c_1 - 2a_1) & (a_1 + hb_1/2) & & & \\ (a_2 - hb_2/2) & (h^2c_2 - 2a_2) & (a_2 + hb_2/2) & & \\ \ddots & \ddots & \ddots & & \\ & (a_{m-1} - hb_{m-1}/2) & (h^2c_{m-1} - 2a_{m-1}) & (a_{m-1} + hb_{m-1}/2) & \\ & & (a_m - hb_m/2) & (h^2c_m - 2a_m) \end{bmatrix} \tag{2.67}$$

and

$$U = \begin{bmatrix} U_1 \\ U_2 \\ \vdots \\ U_{m-1} \\ U_m \end{bmatrix}, \qquad F = \begin{bmatrix} f_1 - (a_1/h^2 - b_1/2h)\alpha \\ f_2 \\ \vdots \\ f_{m-1} \\ f_m - (a_m/h^2 + b_m/2h)\beta \end{bmatrix}. \tag{2.68}$$

This linear system can be solved with standard techniques, assuming the matrix is nonsingular. A singular matrix would be a sign that the discrete system does not have a unique solution, which may occur if the original problem, or a nearby problem, is not well posed (see Section 2.13).

The discretization used above, while second order accurate, may not be the best discretization to use for certain problems of this type. Often the physical problem has certain properties that we would like to preserve with our discretization, and it is important to understand the underlying problem and be aware of its mathematical properties before blindly applying a numerical method. The next example illustrates this.

Example 2.1. Consider heat conduction in a rod with varying heat conduction properties, where the parameter $\kappa(x)$ varies with x and is always positive. The steady-state heat-conduction problem is then

$$(\kappa(x)u'(x))' = f(x) \tag{2.69}$$

together with some boundary conditions, say, the Dirichlet conditions (2.65). To discretize this equation we might be tempted to apply the chain rule to rewrite (2.69) as

$$\kappa(x)u''(x) + \kappa'(x)u'(x) = f(x) \tag{2.70}$$

and then apply the discretization (2.67), yielding the matrix

$$A = \frac{1}{h^2}\begin{bmatrix} -2\kappa_1 & (\kappa_1 + h\kappa_1'/2) & & & \\ (\kappa_2 - h\kappa_2'/2) & -2\kappa_2 & (\kappa_2 + h\kappa_2'/2) & & \\ \ddots & \ddots & \ddots & & \\ & (\kappa_{m-1} - h\kappa_{m-1}'/2) & -2\kappa_{m-1} & (\kappa_{m-1} + h\kappa_{m-1}'/2) & \\ & & (\kappa_m - h\kappa_m'/2) & -2\kappa_m \end{bmatrix}. \tag{2.71}$$

However, this is not the best approach. It is better to discretize the physical problem (2.69) directly. This can be done by first approximating $\kappa(x)u'(x)$ at points halfway between the grid points, using a centered approximation

$$\kappa(x_{i+1/2})u'(x_{i+1/2}) = \kappa_{i+1/2}\left(\frac{U_{i+1} - U_i}{h}\right)$$

and the analogous approximation at $x_{i-1/2}$. Differencing these then gives a centered approximation to $(\kappa u')'$ at the grid point x_i:

$$\begin{aligned}(\kappa u')'(x_i) &\approx \frac{1}{h}\left[\kappa_{i+1/2}\left(\frac{U_{i+1} - U_i}{h}\right) - \kappa_{i-1/2}\left(\frac{U_i - U_{i-1}}{h}\right)\right] \\ &= \frac{1}{h^2}[\kappa_{i-1/2}U_{i-1} - (\kappa_{i-1/2} + \kappa_{i+1/2})U_i + \kappa_{i+1/2}U_{i+1}].\end{aligned} \tag{2.72}$$

This leads to the matrix

$$A = \frac{1}{h^2}\begin{bmatrix} -(\kappa_{1/2} + \kappa_{3/2}) & \kappa_{3/2} & & & \\ \kappa_{3/2} & -(\kappa_{3/2} + \kappa_{5/2}) & \kappa_{5/2} & & \\ \ddots & \ddots & \ddots & & \\ & \kappa_{m-3/2} & -(\kappa_{m-3/2} + \kappa_{m-1/2}) & \kappa_{m-1/2} & \\ & & \kappa_{m-1/2} & -(\kappa_{m-1/2} + \kappa_{m+1/2}) \end{bmatrix}. \tag{2.73}$$

Comparing (2.71) to (2.73), we see that they agree to $O(h^2)$, noting, for example, that

$$\kappa(x_{i+1/2}) = \kappa(x_i) + \frac{1}{2}h\kappa'(x_i) + O(h^2) = \kappa(x_{i+1}) - \frac{1}{2}h\kappa'(x_{i+1}) + O(h^2).$$

However, the matrix (2.73) has the advantage of being *symmetric*, as we would hope, since the original differential equation is *self-adjoint*. Moreover, since $\kappa > 0$, the matrix can be shown to be nonsingular and *negative definite*. This means that all the eigenvalues are negative, a property also shared by the differential operator $\frac{\partial}{\partial x}\kappa(x)\frac{\partial}{\partial x}$ (see Section C.8). It is generally desirable to have important properties such as these modeled by the discrete approximation to the differential equation. One can then show, for example, that the solution to the difference equations satisfies a *maximum principle* of the same type as the solution to the differential equation: for the homogeneous equation with $f(x) \equiv 0$, the values of $u(x)$ lie between the values of the boundary values α and β everywhere, so the maximum and minimum values of u arise on the boundaries. For the heat conduction problem this is physically obvious: the steady-state temperature in the rod won't exceed what's imposed at the boundaries if there is no heat source.

When solving the resulting linear system by iterative methods (see Chapters 3 and 4) it is also often desirable that the matrix have properties such as negative definiteness, since some iterative methods (e.g., the conjugate-gradient (CG) method in Section 4.3) depend on such properties.

2.16 Nonlinear equations

We next consider a nonlinear BVP to illustrate the new complications that arise in this case. We will consider a specific example that has a simple physical interpretation which makes it easy to understand and interpret solutions. This example also illustrates that not all 2-point BVPs are steady-state problems.

Consider the motion of a pendulum with mass m at the end of a rigid (but massless) bar of length L, and let $\theta(t)$ be the angle of the pendulum from vertical at time t, as illustrated in Figure 2.3. Ignoring the mass of the bar and forces of friction and air resistance, we see that the differential equation for the pendulum motion can be well approximated by

$$\theta''(t) = -(g/L)\sin(\theta(t)), \tag{2.74}$$

where g is the gravitational constant. Taking $g/L = 1$ for simplicity we have

$$\theta''(t) = -\sin(\theta(t)) \tag{2.75}$$

as our model problem.

For small amplitudes of the angle θ it is possible to approximate $\sin(\theta) \approx \theta$ and obtain the approximate *linear* differential equation

$$\theta''(t) = -\theta(t) \tag{2.76}$$

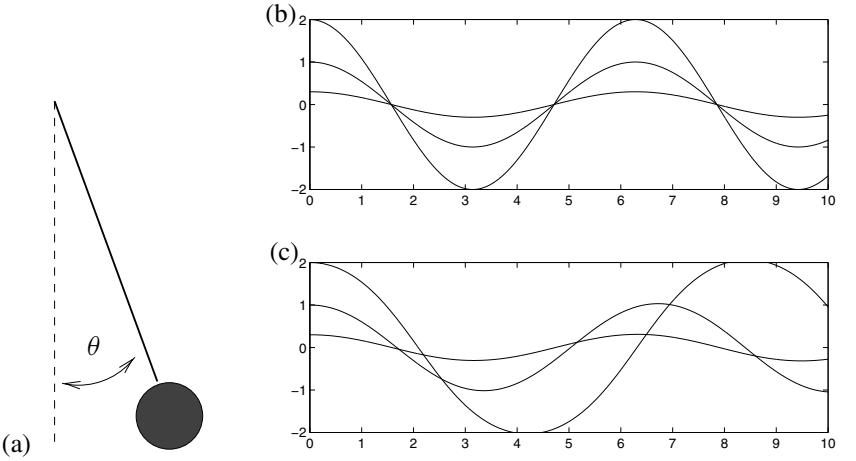

Figure 2.3. (a) *Pendulum.* (b) *Solutions to the linear equation* (2.76) *for various initial* θ *and zero initial velocity.* (c) *Solutions to the nonlinear equation* (2.75) *for various initial* θ *and zero initial velocity.*

with general solutions of the form $A\cos(t) + B\sin(t)$. The motion of a pendulum that is oscillating only a small amount about the equilibrium at $\theta = 0$ can be well approximated by this sinusoidal motion, which has period 2π independent of the amplitude. For larger-amplitude motions, however, solving (2.76) does not give good approximations to the true behavior. Figures 2.3(b) and (c) show some sample solutions to the two equations.

To fully describe the problem we also need to specify two auxiliary conditions in addition to the second order differential equation (2.75). For the pendulum problem the IVP is most natural—we set the pendulum swinging from some initial position $\theta(0)$ with some initial angular velocity $\theta'(0)$, which gives two initial conditions that are enough to determine a unique solution at all later times.

To obtain instead a BVP, consider the situation in which we wish to set the pendulum swinging from some initial given location $\theta(0) = \alpha$ with some unknown angular velocity $\theta'(0)$ in such a way that the pendulum will be at the desired location $\theta(T) = \beta$ at some specified later time T. Then we have a 2-point BVP

$$\begin{aligned} &\theta''(t) = -\sin(\theta(t)) \quad \text{for } 0 < t < T, \\ &\theta(0) = \alpha, \qquad \theta(T) = \beta. \end{aligned} \tag{2.77}$$

Similar BVPs do arise in more practical situations, for example, trying to shoot a missile in such a way that it hits a desired target. In fact, this latter example gives rise to the name *shooting method* for another approach to solving 2-point BVPs that is discussed in [4] and [54], for example.

2.16.1 Discretization of the nonlinear boundary value problem

We can discretize the nonlinear problem (2.75) in the obvious manner, following our approach for linear problems, to obtain the system of equations

$$\frac{1}{h^2}(\theta_{i-1} - 2\theta_i + \theta_{i+1}) + \sin(\theta_i) = 0 \tag{2.78}$$

for $i = 1, 2, \ldots, m$, where $h = T/(m+1)$ and we set $\theta_0 = \alpha$ and $\theta_{m+1} = \beta$. As in the linear case, we have a system of m equations for m unknowns. However, this is now a *nonlinear system* of equations of the form

$$G(\theta) = 0, \tag{2.79}$$

where $G : \mathbb{R}^m \to \mathbb{R}^m$. This cannot be solved as easily as the tridiagonal linear systems encountered so far. Instead of a direct method we must generally use some *iterative method*, such as Newton's method. If $\theta^{[k]}$ is our approximation to θ in step k, then *Newton's method* is derived via the Taylor series expansion

$$G(\theta^{[k+1]}) = G(\theta^{[k]}) + G'(\theta^{[k]})(\theta^{[k+1]} - \theta^{[k]}) + \cdots.$$

Setting $G(\theta^{[k+1]}) = 0$ as desired, and dropping the higher order terms, results in

$$0 = G(\theta^{[k]}) + G'(\theta^{[k]})(\theta^{[k+1]} - \theta^{[k]}).$$

This gives the Newton update

$$\theta^{[k+1]} = \theta^{[k]} + \delta^{[k]}, \tag{2.80}$$

where $\delta^{[k]}$ solves the linear system

$$J(\theta^{[k]})\delta^{[k]} = -G(\theta^{[k]}). \tag{2.81}$$

Here $J(\theta) \equiv G'(\theta) \in \mathbb{R}^{m\times m}$ is the *Jacobian matrix* with elements

$$J_{ij}(\theta) = \frac{\partial}{\partial\theta_j} G_i(\theta),$$

where $G_i(\theta)$ is the ith component of the vector-valued function G. In our case $G_i(\theta)$ is exactly the left-hand side of (2.78), and hence

$$J_{ij}(\theta) = \begin{cases} 1/h^2 & \text{if } j = i-1 \text{ or } j = i+1, \\ -2/h^2 + \cos(\theta_i) & \text{if } j = i, \\ 0 & \text{otherwise,} \end{cases}$$

so that

$$J(\theta) = \frac{1}{h^2}\begin{bmatrix} (-2+h^2\cos(\theta_1)) & 1 & & & \\ 1 & (-2+h^2\cos(\theta_2)) & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & (-2+h^2\cos(\theta_m)) \end{bmatrix}. \tag{2.82}$$

In each iteration of Newton's method we must solve a tridiagonal linear system similar to the single tridiagonal system that must be solved in the linear case.

Consider the nonlinear problem with $T = 2\pi$, $\alpha = \beta = 0.7$. Note that the linear problem (2.76) has infinitely many solutions in this particular case since the linearized pendulum has period 2π independent of the amplitude of motion; see Figure 2.3. This is not true of the nonlinear equation, however, and so we might expect a unique solution to the full nonlinear problem. With Newton's method we need an initial guess for the solution, and in Figure 2.4(a) we take a particular solution to the linearized problem, the one with initial angular velocity 0.5, as a first approximation, i.e., $\theta_i^{[0]} = 0.7\cos(t_i) + 0.5\sin(t_i)$. Figure 2.4(a) shows the different $\theta^{[k]}$ for $k = 0,\ 1,\ 2,\ \ldots$ that are obtained as we iterate with Newton's method. They rapidly converge to a solution to the nonlinear system (2.78). (Note that the solution looks similar to the solution to the linearized equation with $\theta'(0) = 0$, as we should have expected, and taking this as the initial guess, $\theta^{[0]} = 0.7\cos(t)$, would have given even more rapid convergence.)

Table 2.1 shows $\|\delta^{[k]}\|_\infty$ in each iteration, which measures the change in the solution. As expected, Newton's method appears to be converging quadratically.

If we start with a different initial guess $\theta^{[0]}$ (but still close enough to this solution), we would find that the method still converges to this same solution. For example, Figure 2.4(b) shows the iterates $\theta^{[k]}$ for $k = 0,\ 1,\ 2,\ \ldots$ with a different choice of $\theta^{[0]} \equiv 0.7$.

Newton's method can be shown to converge if we start with an initial guess that is sufficiently close to a solution. How close depends on the nature of the problem. For the

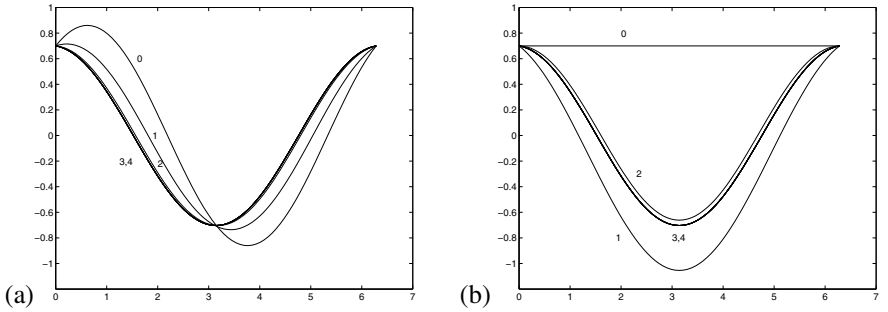

Figure 2.4. *Convergence of Newton iterates toward a solution of the pendulum problem. The iterates $\theta^{[k]}$ for $k = 1, 2, \ldots$ are denoted by the number k in the plots.* (a) *Starting from* $\theta_i^{[0]} = 0.7\cos(t_i) + 0.5\sin(t_i)$. (b) *Starting from* $\theta_i^{[0]} = 0.7$.

Table 2.1. *Change $\|\delta^{[k]}\|_\infty$ in solution in each iteration of Newton's method.*

k	Figure 2.4(a)	Figure 2.5
0	3.2841e−01	4.2047e+00
1	1.7518e−01	5.3899e+00
2	3.1045e−02	8.1993e+00
3	2.3739e−04	7.7111e−01
4	1.5287e−08	3.8154e−02
5	5.8197e−15	2.2490e−04
6	1.5856e−15	9.1667e−09
7		1.3395e−15

problem considered above one need not start very close to the solution to converge, as seen in the examples, but for more sensitive problems one might have to start extremely close. In such cases it may be necessary to use a technique such as *continuation* to find suitable initial data; see Section 2.19.

2.16.2 Nonuniqueness

The nonlinear problem does not have an infinite family of solutions the way the linear equation does on the interval $[0, 2\pi]$, and the solution found above is an *isolated solution* in the sense that there are no other solutions very nearby (it is also said to be *locally unique*). However, it does not follow that this is the unique solution to the BVP (2.77). In fact physically we should expect other solutions. The solution we found corresponds to releasing the pendulum with nearly zero initial velocity. It swings through nearly one complete cycle and returns to the initial position at time T.

Another possibility would be to propel the pendulum upward so that it rises toward the top (an unstable equilibrium) at $\theta = \pi$, before falling back down. By specifying the correct velocity we should be able to arrange it so that the pendulum falls back to $\theta = 0.7$ again at $T = 2\pi$. In fact it is possible to find such a solution for any $T > 0$.

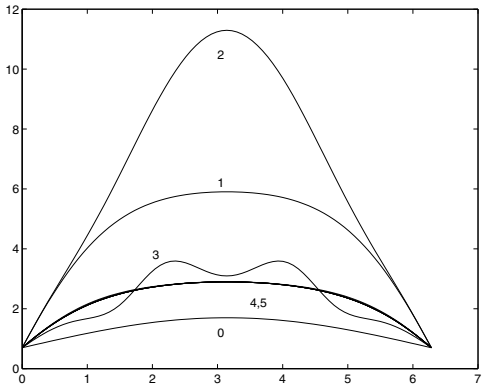

Figure 2.5. *Convergence of Newton iterates toward a different solution of the pendulum problem starting with initial guess* $\theta_i^{[0]} = 0.7 + \sin(t_i/2)$. *The iterates* k *for* $k = 1, 2, \ldots$ *are denoted by the number* k *in the plots.*

Physically it seems clear that there is a second solution to the BVP. To find it numerically we can use the same iteration as before, but with a different initial guess $\theta^{[0]}$ that is sufficiently close to this solution. Since we are now looking for a solution where θ initially increases and then falls again, let's try a function with this general shape. In Figure 2.5 we see the iterates $\theta^{[k]}$ generated with data $\theta_i^{[0]} = 0.7 + \sin(t_i/2)$. We have gotten lucky here on our first attempt, and we get convergence to a solution of the desired form. (See Table 2.1.) Different guesses with the same general shape might not work. Note that some of the iterates $\theta^{[k]}$ obtained along the way in Figure 2.5 do not make physical sense (since θ goes above π and then back down—what does this mean?), but the method still converges.

2.16.3 Accuracy on nonlinear equations

The solutions plotted above are not exact solutions to the BVP (2.77). They are only solutions to the discrete system of (2.78) with $h = 1/80$. How well do they approximate true solutions of the differential equation? Since we have used a second order accurate centered approximation to the second derivative in (2.8), we again hope to obtain second order accuracy as the grid is refined. In this section we will investigate this.

Note that it is very important to keep clear the distinction between the convergence of Newton's method to a solution of the finite difference equations and the convergence of this finite difference approximation to the solution of the differential equation. Table 2.1 indicates that we have obtained a solution to machine accuracy (roughly 10^{-15}) of the non linear system of equations by using Newton's method. This does *not* mean that our solution agrees with the true solution of the differential equation to the same degree. This depends on the size of h, the size of the truncation error in our finite difference approximation, and the relation between the local truncation error and the resulting global error.

Let's start by computing the local truncation error of the finite difference formula. Just as in the linear case, we define this by inserting the true solution of the differential

equation into the finite difference equations. This will not satisfy the equations exactly, and the residual is what we call the *local truncation error* (LTE):

$$\begin{aligned}\tau_i &= \frac{1}{h^2}(\theta(t_{i-1}) - 2\theta(t_i) + \theta(t_{i+1})) + \sin(\theta(t_i)) \\ &= (\theta''(t_i) + \sin(\theta(t_i))) + \frac{1}{12}h^2\theta''''(t_i) + O(h^4) \\ &= \frac{1}{12}h^2\theta''''(t_i) + O(h^4).\end{aligned} \tag{2.83}$$

Note that we have used the differential equation to set $\theta''(t_i) + \sin(\theta(t_i)) = 0$, which holds exactly since $\theta(t)$ is the exact solution. The LTE is $O(h^2)$ and has exactly the same form as in the linear case. (For a more complicated nonlinear problem it might not work out so simply, but similar expressions result.) The vector τ with components τ_i is simply $G(\hat{\theta})$, where $\hat{\theta}$ is the vector made up of the true solution at each grid point. We now want to obtain an estimate on the global error E based on this local error. We can attempt to follow the path used in Section 2.6 for linear problems. We have

$$\begin{aligned}G(\theta) &= 0, \\ G(\hat{\theta}) &= \tau,\end{aligned}$$

and subtracting gives

$$G(\theta) - G(\hat{\theta}) = -\tau. \tag{2.84}$$

We would like to derive from this a relation for the global error $E = \theta - \hat{\theta}$. If G were linear (say, $G(\theta) = A\theta - F$), we would have $G(\theta) - G(\hat{\theta}) = A\theta - A\hat{\theta} = A(\theta - \hat{\theta}) = AE$, giving an expression in terms of the global error $E = \theta - \hat{\theta}$. This is what we used in Section 2.7.

In the nonlinear case we cannot express $G(\theta) - G(\hat{\theta})$ directly in terms of $\theta - \hat{\theta}$. However, we can use Taylor series expansions to write

$$G(\theta) = G(\hat{\theta}) + J(\hat{\theta})E + O(\|E\|^2),$$

where $J(\hat{\theta})$ is again the Jacobian matrix of the difference formulas, evaluated now at the exact solution. Combining this with (2.84) gives

$$J(\hat{\theta})E = -\tau + O(\|E\|^2).$$

If we ignore the higher order terms, then we again have a linear relation between the local and global errors.

This motivates the following definition of stability. Here we let $\hat{J}^h$ denote the Jacobian matrix of the difference formulas evaluated at the true solution on a grid with grid spacing h.

Definition 2.2. *The nonlinear difference method $G(\theta) = 0$ is stable in some norm $\|\cdot\|$ if the matrices $(\hat{J}^h)^{-1}$ are uniformly bounded in this norm as $h \to 0$, i.e., there exist constants C and h_0 such that*

$$\|(\hat{J}^h)^{-1}\| \le C \quad \text{for all } h < h_0. \tag{2.85}$$

It can be shown that if the method is stable in this sense, and consistent in this norm ($\|\tau^h\| \to 0$), then the method converges and $\|E^h\| \to 0$ as $h \to 0$. This is not obvious in the nonlinear case: we obtain a linear system for E only by dropping the $O(\|E\|^2)$ nonlinear terms. Since we are trying to show that E is small, we can't necessarily assume that these terms are negligible in the course of the proof, at least not without some care. See [54] for a proof.

It makes sense that it is uniform boundedness of the inverse Jacobian at the exact solution that is required for stability. After all, it is essentially this Jacobian matrix that is used in solving linear systems in the course of Newton's method, once we get very close to the solution.

Warning: We state a final reminder that there is a difference between convergence of the difference method as $h \to 0$ and convergence of Newton's method, or some other iterative method, to the solution of the difference equations for some particular h. Stability of the difference method does not imply that Newton's method will converge from a poor initial guess. It can be shown, however, that with a stable method, Newton's method will converge from a sufficiently good initial guess; see [54]. Also, the fact that Newton's method has converged to a solution of the nonlinear system of difference equations, with an error of 10^{-15}, say, does not mean that we have a good solution to the original differential equation. The global error of the difference equations determines this.

2.17 Singular perturbations and boundary layers

In this section we consider some singular perturbation problems to illustrate the difficulties that can arise when numerically solving problems with boundary layers or other regions where the solution varies rapidly. See [55], [56] for more detailed discussions of singular perturbation problems. In particular, the example used here is very similar to one that can be found in [55], where solution by matched asymptotic expansions is discussed.

As a simple example we consider a steady-state advection-diffusion equation. The time-dependent equation has the form

$$u_t + au_x = \kappa u_{xx} + \psi \tag{2.86}$$

in the simplest case. This models the temperature $u(x,t)$ of a fluid flowing through a pipe with constant velocity a, where the fluid has constant heat diffusion coefficient κ and ψ is a source term from heating through the walls of the tube.

If $a > 0$, then we naturally have a boundary condition at the left boundary (say, $x = 0$),

$$u(0,t) = \alpha(t),$$

specifying the temperature of the incoming fluid. At the right boundary (say, $x = 1$) the fluid is flowing out and so it may seem that the temperature is determined only by what is happening in the pipe, and no boundary condition is needed here. This is correct if $\kappa = 0$ since the first order advection equation needs only one boundary condition and we are allowed to specify u only at the left boundary. However, if $\kappa > 0$, then heat can diffuse upstream, and we need to also specify $u(1,t) = \beta(t)$ to determine a unique solution.

If α, β, and ψ are all independent of t, then we expect a steady-state solution, which we hope to find by solving the linear 2-point boundary value problem

$$\begin{aligned} au'(x) &= \kappa u''(x) + \psi(x), \\ u(0) &= \alpha, \qquad u(1) = \beta. \end{aligned} \tag{2.87}$$

This can be discretized using the approach of Section 2.4. If a is small relative to κ, then this problem is easy to solve. In fact for $a = 0$ this is just the steady-state heat equation discussed in Section 2.15, and for small a the solution appears nearly identical.

But now suppose a is large relative to κ (i.e., we crank up the velocity, or we decrease the ability of heat to diffuse with the velocity $a > 0$ fixed). More properly we should work in terms of the nondimensional *Péclet number*, which measures the ratio of advection velocity to transport speed due to diffusion. Here we introduce a parameter ϵ which is like the inverse of the Péclet number, $\epsilon = \kappa/a$, and rewrite (2.87) in the form

$$\epsilon u''(x) - u'(x) = f(x). \tag{2.88}$$

Then taking a large relative to κ (large Péclet number) corresponds to the case $\epsilon \ll 1$.

We should expect difficulties physically in this case where advection overwhelms diffusion. It would be very difficult to maintain a fixed temperature at the outflow end of the tube in this situation. If we had a thermal device that was capable of doing so by instantaneously heating the fluid to the desired temperature as it passes the right boundary, independent of the temperature of the fluid flowing toward this point, then we would expect the temperature distribution to be essentially discontinuous at this boundary.

Mathematically we expect trouble as $\epsilon \to 0$ because in the limit $\epsilon = 0$ the equation (2.88) reduces to a *first order* equation (the steady advection equation)

$$-u'(x) = f(x), \tag{2.89}$$

which allows only one boundary condition, rather than two. For $\epsilon > 0$, no matter how small, we have a second order equation that needs two conditions, but we expect to perhaps see strange behavior at the outflow boundary as $\epsilon \to 0$, since in the limit we are over specifying the problem.

Figure 2.6(a) shows how solutions to (2.88) appear for various values of ϵ in the case $\alpha = 1$, $\beta = 3$, and $f(x) = -1$. In this case the exact solution is

$$u(x) = \alpha + x + (\beta - \alpha - 1)\left(\frac{e^{x/\epsilon} - 1}{e^{1/\epsilon} - 1}\right). \tag{2.90}$$

Note that as $\epsilon \to 0$ the solution tends toward a discontinuous function that jumps to the value β at the last possible moment. This region of rapid transition is called the *boundary layer* and it can be shown that for this problem the width of this layer is $O(\epsilon)$ as $\epsilon \to 0$.

The equation (2.87) with $0 < \epsilon \ll 1$ is called a *singularly perturbed equation*. It is a small perturbation of (2.89), but this small perturbation completely changes the character of the equation (from a first order to a second order equation). Typically any differential equation having a small parameter multiplying the highest order derivative will give a singular perturbation problem.

By contrast, going from the pure diffusion equation $\kappa u_{xx} = f$ to an advection diffusion equation $\kappa u_{xx} - a u_x = f$ for very small a is a *regular perturbation*. Both of these equations are second order differential equations requiring the same number of

boundary conditions. The solution of the perturbed equation looks nearly identical to the solution of the unperturbed equation for small a, and the difference in solutions is $O(a)$ as $a \to 0$.

Singular perturbation problems cause numerical difficulties because the solution changes rapidly over a very small interval in space. In this region derivatives of $u(x)$ are large, giving rise to large errors in our finite difference approximations. Recall that the error in our approximation to $u''(x)$ is proportional to $h^2 u''''(x)$, for example. If h is not small enough, then the local truncation error will be very large in the boundary layer. Moreover, even if the truncation error is large only in the boundary layer, the resulting global error may be large everywhere. (Recall that the global error E is obtained from the truncation error τ by solving a linear system $AE = -\tau$, which means that each element of E depends on *all* elements of τ since A^{-1} is a dense matrix.) This is clearly seen in Figure 2.6(b), where the numerical solution with $h = 1/10$ is plotted. Errors are large even in regions where the exact solution is nearly linear and $u'''' \approx 0$.

On finer grids the solution looks better (see Figure 2.6(c) and (d)), and as $h \to 0$ the method does exhibit second order accurate convergence. But it is necessary to have a sufficiently fine grid before reasonable results are obtained; we need enough grid points to enable the boundary layer to be well resolved.

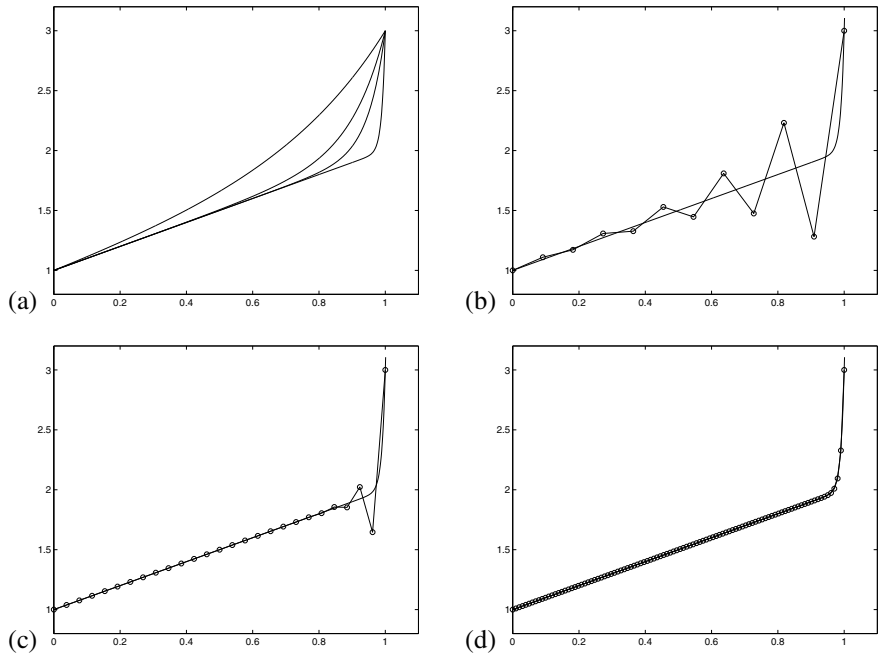

Figure 2.6. (a) *Solutions to the steady state advection-diffusion equation* (2.88) *for different values of* ϵ. *The four lines correspond to* $\epsilon = 0.3,\ 0.1,\ 0.05$, *and* 0.01 *from top to bottom.* (b) *Numerical solution with* $\epsilon = 0.01$ *and* $h = 1/10$. (c) $h = 1/25$. (d) $h = 1/100$.

2.17.1 Interior layers

The above example has a boundary layer, a region of rapid transition at one boundary. Other problems may have *interior layers* instead. In this case the solution is smooth except for some thin region interior to the interval where a rapid transition occurs. Such problems can be even more difficult to solve since we often don't know a priori where the interior layer will be. Perturbation theory can often be used to analyze singular perturbation problems and predict where the layers will occur, how wide they will be (as a function of the small parameter ϵ), and how the solution behaves. The use of perturbation theory to obtain good approximations to problems of this type is a central theme of classical applied mathematics.

These analytic techniques can often be used to good advantage along with numerical methods, for example, to obtain a good initial guess for Newton's method, or to choose an appropriate nonuniform grid as discussed in the next section. In some cases it is possible to develop special numerical methods that have the correct singular behavior built into the approximation in such a way that far better accuracy is achieved than with a naive numerical method.

Example 2.2. Consider the nonlinear boundary value problem

$$\begin{aligned} &\epsilon u'' + u(u' - 1) = 0 \qquad \text{for } a \le x \le b, \\ &u(a) = \alpha, \quad u(b) = \beta. \end{aligned} \tag{2.91}$$

For small ϵ this is a singular perturbation problem since ϵ multiplies the highest order derivative. Setting $\epsilon = 0$ gives a reduced equation

$$u(u' - 1) = 0 \tag{2.92}$$

for which we generally can enforce only one boundary condition. Solutions to (2.92) are $u(x) \equiv 0$ or $u(x) = x + C$ for some constant C. If the boundary condition imposed at $x = a$ or $x = b$ is nonzero, then the solution has the latter form and is either

$$u(x) = x + \alpha - a \quad \text{if } u(a) = \alpha \text{ is imposed} \tag{2.93}$$

or

$$u(x) = x + \beta - b \quad \text{if } u(b) = \beta \text{ is imposed.} \tag{2.94}$$

These two solutions are shown in Figure 2.7.

For $0 < \epsilon \ll 1$, the full equation (2.91) has a solution that satisfies both boundary conditions, and Figure 2.7 also shows such a solution. Over most of the domain the solution is smooth and u'' is small, in which case $\epsilon u''$ is negligible and the solution must nearly satisfy (2.92). Thus over most of the domain the solution follows one of the linear solutions to the reduced equation. Both boundary conditions can be satisfied by following one solution (2.93) near $x = a$ and the other solution (2.94) near $x = b$. Connecting these two smooth portions of the solution is a narrow zone (the interior solution) where $u(x)$ is rapidly varying. In this layer u'' is very large and the $\epsilon u''$ term of (2.91) is not negligible, and hence $u(u' - 1)$ may be far from zero in this region.

To determine the location and width of the interior layer, and the approximate form of the solution in this layer, we can use perturbation theory. Focusing attention on this

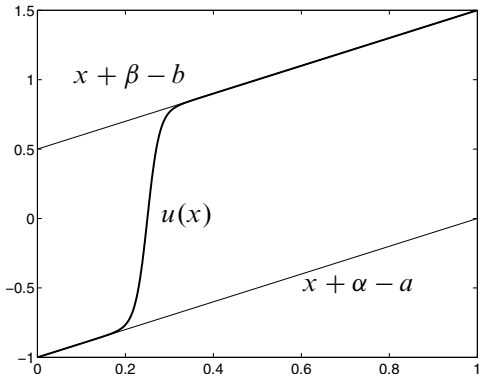

Figure 2.7. *Outer solutions and full solution to the singular perturbation problem with* $a = 0$, $b = 1$, $\alpha = -1$, *and* $\beta = 1.5$. *The solution has an interior layer centered about* $\bar{x} = 0.25$.

layer, which we now assume is centered at some location $x = \bar{x}$, we can zoom in on the solution by assuming that $u(x)$ has the approximate form

$$u(x) = W((x - \bar{x})/\epsilon^k) \tag{2.95}$$

for some power k to be determined. We are zooming in on a layer of width $O(\epsilon^k)$ asymptotically, so determining k will tell us how wide the layer is. From (2.95) we compute

$$\begin{aligned} u'(x) &= \epsilon^{-k} W'((x - \bar{x})/\epsilon^k), \\ u''(x) &= \epsilon^{-2k} W'((x - \bar{x})/\epsilon^k). \end{aligned} \tag{2.96}$$

Inserting these expressions in (2.91) gives

$$\epsilon \cdot \epsilon^{-2k} W''(\xi) + W(\xi)(\epsilon^{-k} W'(\xi) - 1) = 0,$$

where $\xi = (x - \bar{x})/\epsilon^k$. Multiply by ϵ^{2k-1} to obtain

$$W''(\xi) + W(\xi)(\epsilon^{k-1} W'(\xi) - \epsilon^{2k-1}) = 0. \tag{2.97}$$

By rescaling the independent variable by a factor ϵ^k, we have converted the singular perturbation problem (2.91) into a problem where the highest order derivative W'' has coefficient 1 and the small parameter appears only in the lower order term. However, the lower order term behaves well in the limit $\epsilon \to 0$ only if we take $k \geq 1$. For smaller values of k (zooming in on too large a region asymptotically), the lower order term blows up as $\epsilon \to 0$, or dividing by ϵ^{k-1} shows that we still have a singular perturbation problem. This gives us some information on k.

If we fix x at any value away from $\bar{x}$, then $\xi \to \pm\infty$ as $\epsilon \to 0$. So the boundary value problem (2.97) for $W(\xi)$ has boundary conditions at $\pm\infty$,

$$\begin{aligned} W(\xi) &\to \bar{x} + \alpha - a \quad \text{as } \xi \to -\infty, \\ W(\xi) &\to \bar{x} + \beta - b \quad \text{as } \xi \to +\infty. \end{aligned} \tag{2.98}$$

The "inner solution" $W(\xi)$ will then match up with the "outer solutions" given by (2.93) and (2.94) at the edges of the layer. We also require

$$W'(\xi) \to 0 \quad \text{as } \xi \to \pm\infty \tag{2.99}$$

since outside the layer the linear functions (2.93) and (2.94) have the desired slope.

For (2.97) to give a reasonable 2-point boundary value problem with these three boundary conditions (2.98) and (2.99), we must take $k = 1$. We already saw that we need $k \geq 1$, but we also cannot take $k > 1$ since in this case the lower order term in (2.97) vanishes as $\epsilon \to 0$ and the equation reduces to $W''(\xi) = 0$. In this case we are zooming in too far on the solution near $x = \bar{x}$ and the solution simply appears linear, as does any sufficiently smooth function if we zoom in on its behavior at a fixed point. While this does reveal the behavior extremely close to $\bar{x}$, it does not allow us to capture the full behavior in the interior layer. We cannot satisfy all four boundary conditions on W with a solution to $W''(x) = 0$.

Taking $k = 1$ gives the proper interior problem, and (2.97) becomes

$$W''(\xi) + W(\xi)(W'(\xi) - \epsilon) = 0. \tag{2.100}$$

Now letting $\epsilon \to 0$ we obtain

$$W''(\xi) + W(\xi)W'(\xi) = 0. \tag{2.101}$$

This equation has solutions of the form

$$W(\xi) = w_0 \tanh(w_0\xi/2) \tag{2.102}$$

for arbitrary constants w_0. The boundary conditions (2.98) lead to

$$w_0 = \frac{1}{2}(a - b + \beta - \alpha) \tag{2.103}$$

and

$$\bar{x} = \frac{1}{2}(a + b - \alpha - \beta). \tag{2.104}$$

To match this solution to the outer solutions, we require $a < \bar{x} < b$. If the value of $\bar{x}$ determined by (2.104) doesn't satisfy this condition, then the original problem has a boundary layer at $x = a$ (if $\bar{x} \leq a$) or at $x = b$ (if $\bar{x} \geq b$) instead of an interior layer. For the remainder of this discussion we assume $a < \bar{x} < b$.

We can combine the inner and outer solutions to obtain an approximate solution of the form

$$u(x) \approx \tilde{u}(x) \equiv x - \bar{x} + w_0 \tanh(w_0(x - \bar{x})/2\epsilon). \tag{2.105}$$

Singular perturbation analysis has given us a great deal of information about the solution to the problem (2.91). We know that the solution has an interior layer of width $O(\epsilon)$ at $x = \bar{x}$ with roughly linear solution (2.93), (2.94) outside the layer. This type of information may be all we need to know about the solution for some applications. If we want to determine a more detailed numerical approximation to the full solution, this analytical

information can be helpful in devising an accurate and efficient numerical method, as we now consider.

The problem (2.91) can be solved numerically on a uniform grid using the finite difference equations

$$G_i(U) \equiv \epsilon \left(\frac{U_{i-1} - 2U_i + U_{i+1}}{h^2} \right) + U_i \left(\frac{U_{i+1} - U_{i-1}}{2h} - 1 \right) = 0 \tag{2.106}$$

for $i = 1, 2, \ldots, m$ with $U_0 = \alpha$ and $U_{m+1} = \beta$ (where, as usual, $h = (b-a)/(m+1)$). This gives a nonlinear system of equations $G(U) = 0$ that can be solved using Newton's method as described in Section 2.16.1. One way to use the singular perturbation approximation is to generate a good initial guess for Newton's method, e.g.,

$$U_i = \tilde{u}(x_i), \tag{2.107}$$

where $\tilde{u}(x)$ is the approximate solution from (2.105). We then have an initial guess that is already very accurate at nearly all grid points. Newton's method converges rapidly from such a guess. If the grid is fine enough that the interior layer is well resolved, then a good approximation to the full solution is easily obtained. By contrast, starting with a more naive initial guess such as $U_i = \alpha + (x - a)(\beta - \alpha)/(b - a)$ leads to nonconvergence when ϵ is small.

When ϵ is very small, highly accurate numerical results can be obtained with less computation by using a nonuniform grid, with grid points clustered in the layer. To construct such a grid we can use the singular perturbation analysis to tell us where the points should be clustered (near $\bar{x}$) and how wide to make the clustering zone. The width of the layer is $O(\epsilon)$ and, moreover, from (2.102) we expect that most of the transition occurs for, say, $|\frac{1}{2} w_0 \xi| < 2$. This translates into

$$|x - \bar{x}| < 4\epsilon / w_0, \tag{2.108}$$

where w_0 is given by (2.103). The construction and use of nonuniform grids is pursued further in the next section.

2.18 Nonuniform grids

From Figure 2.6 it is clear that we need to choose our grid to be fine enough so that several points are within the boundary layer and we can obtain a reasonable solution. If we wanted high accuracy within the boundary layer we would have to choose a much finer grid than shown in this figure. With a uniform grid this means using a very large number of grid points, the vast majority of which are in the region where the solution is very smooth and could be represented well with far fewer points. This waste of effort may be tolerable for simple one-dimensional problems but can easily be intolerable for more complicated problems, particularly in more than one dimension.

Instead it is preferable to use a nonuniform grid for such calculations, with grid points clustered in regions where they are most needed. This requires using formulas that are sufficiently accurate on nonuniform grids. For example, a four-point stencil can be used to obtain second order accuracy for the second derivative operator. Using this for a linear

problem would give a banded matrix with four nonzero diagonals. A little extra care is needed at the boundaries.

One way to specify nonuniform grid points is to start with a uniform grid in some "computational coordinate" z, which we will denote by $z_i = ih$ for $i = 0, 1, \ldots, m+1$, where $h = 1/(m+1)$, and then use some appropriate *grid mapping* function $X(z)$ to define the "physical grid points" $x_i = X(z_i)$. This is illustrated in Figure 2.8, where z is plotted on the vertical axis and x is on the horizontal axis. The curve plotted represents a function $X(z)$, although with this choice of axes it is more properly the graph of the inverse function $z = X^{-1}(x)$. The horizontal and vertical lines indicate how the uniform grid points on the z axis are mapped to nonuniform points in x. If the problem is posed on the interval $[a, b]$, then the function $X(z)$ should be monotonically increasing and satisfy $X(0) = a$ and $X(1) = b$.

Note that grid points are clustered in regions where the curve is steepest, which means that $X(z)$ varies slowly with z, and spread apart in regions where $X(z)$ varies rapidly with z. Singular perturbation analysis of the sort done in the previous section may provide guidelines for where the grid points should be clustered.

Once a set of grid points x_i is chosen, it is necessary to set up and solve an appropriate system of difference equations on this grid. In general a different set of finite difference coefficients will be required at each grid point, depending on the spacing of the grid points nearby.

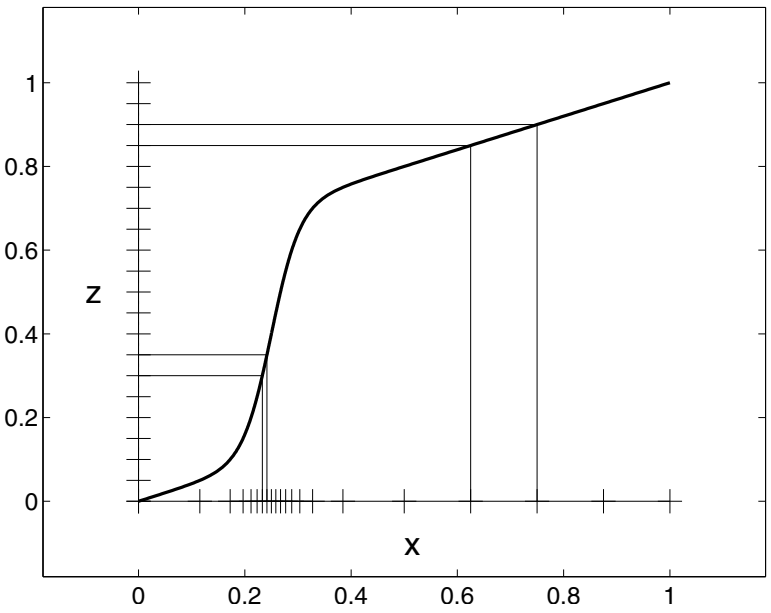

Figure 2.8. *Grid mapping from a uniform grid in* $0 \le z \le 1$ *(vertical axis) to the nonuniform grid in physical* x*-space shown on the horizontal axis. This particular mapping may be useful for solving the singular perturbation problem illustrated in Fig. 2.7.*

Example 2.3. As an example, again consider the simple problem $u''(x) = f(x)$ with the boundary conditions (2.52), $u'(0) = \sigma$, and $u(1) = \beta$. We would like to generalize the matrix system (2.57) to the situation where the x_i are nonuniformly distributed in the interval $[0, 1]$. In MATLAB this is easily accomplished using the `fdcoeffV` function discussed in Section 1.5, and the code fragment below shows how this matrix can be computed. Note that in MATLAB the vector `x` must be indexed from 1 to `m+2` rather than from 0 to `m+1`.

```
A = speye(m+2);   % initialize using sparse storage
% first row for Neumann BC, approximates u'(x(1))
A(1,1:3) = fdcoeffV(1, x(1), x(1:3));
% interior rows approximate u''(x(i))
for i=2:m+1
   A(i,i-1:i+1) = fdcoeffV(2, x(i), x((i-1):(i+1)));
   end
% last row for Dirichlet BC, approximates u(x(m+2))
A(m+2,m:m+2) = fdcoeffV(0,x(m+2),x(m:m+2));
```

A complete program that uses this and tests the order of accuracy of this method is available on the Web page.

Note that in this case the finite difference coefficients for the 3-point approximations to $u''(x_i)$ also can be explicitly calculated from the formula (1.14), but it is simpler to use `fdcoeffV`, and the above code also can be easily generalized to higher order methods by using more points in the stencils.

What accuracy do we expect from this method? In general if x_{i-1}, x_i and x_{i+1} are not equally spaced, then we expect an approximation to the second derivative $u''(x_i)$ based on these three points to be only first order accurate ($n = 3$ and $k = 2$ in the terminology of Section 1.5, so we expect $p = n - k = 1$). This is confirmed by the error expression (1.16), and this is generally what we will observe if we take randomly spaced grid points x_i.

However, in practice we normally use grids that are smoothly varying, for example, $x_i = X(z_i)$, where $X(z)$ is some smooth function, as discussed in Section 2.18. In this case it turns out that we should expect to achieve second order accuracy with the method just presented, and that is what is observed in practice. This can be seen from the error expressions (1.16): the "first order" portion of the error is proportional to

$$h_2 - h_1 = (x_{i+1} - x_i) - (x_i - x_{i-1}) = X(z_{i+1}) - 2X(z_i) + X(z_{i-1}) \approx h^2 X''(z_i),$$

where $h = \Delta z$ is the grid spacing in z. So we see that for a smoothly varying grid the difference $h_2 - h_1$ is actually $O(h^2)$. Hence the local truncation error is $O(h^2)$ at each grid point and we expect second order accuracy globally.

2.18.1 Adaptive mesh selection

Ideally a numerical method would work robustly on problems with interior or boundary layers without requiring that the user know beforehand how the solution is behaving. This can often be achieved by using methods that incorporate some form of *adaptive mesh selection*, which means that the method selects the mesh based on the behavior of the solution

and automatically clusters grid points in regions where they are needed. A discussion of this topic is beyond the scope of this book. See, for example, [4].

Readers who wish to use methods on nonuniform grids are encouraged to investigate software that will automatically choose an appropriate grid for a given problem (perhaps with some initial guidance) and take care of all the details of discretizing on this grid. In MATLAB, the routine `bvp4c` can be used and links to other software may be found on the book's Web page.

2.19 Continuation methods

For a difficult problem (e.g., a boundary layer or interior layer problem with $\epsilon \ll 1$), an adaptive mesh refinement program may not work well unless a reasonable initial grid is provided that already has some clustering in the appropriate layer location. Moreover, Newton's method may not converge unless we have a good initial guess for the solution. We have seen how information about the layer location and width and the approximate form of the solution can sometimes be obtained by using singular perturbation analysis.

There is another approach that is often easier in practice, known as *continuation* or the *homotopy method*. As an example, consider again the interior layer problem considered in Example 2.2 and suppose we want to solve this problem for a very small value of ϵ, say, $\epsilon = 10^{-6}$. Rather than immediately tackling this problem, we could first solve the problem with a much larger value of ϵ, say, $\epsilon = 0.1$, for which the solution is quite smooth and convergence is easily obtained on a uniform grid with few points. This solution can then be used as an initial guess for the problem with a smaller value of ϵ, say, $\epsilon = 10^{-2}$. We can repeat this process as many times as necessary to get to the desired value of ϵ, perhaps also adapting the grid as we go along to cluster points near the location of the interior layer (which is independent of ϵ and becomes clearly defined as we reduce ϵ).

More generally, the idea of following the solution to a differential equation as some parameter in the equation varies arises in other contexts as well. Difficulties sometimes arise at particular parameter values, such as *bifurcation points*, where two paths of solutions intersect.

2.20 Higher order methods

So far we have considered only second order methods for solving BVPs. Several approaches can be used to obtain higher order accurate methods. In this section we will look at various approaches to achieving higher polynomial order, such as fourth order or sixth order approximations. In Section 2.21 we briefly introduce spectral methods that can achieve convergence at exponential rates under some conditions.

2.20.1 Fourth order differencing

The obvious approach is to use a better approximation to the second derivative operator in place of the second order difference used in (2.8). For example, the finite difference approximation

$$\frac{1}{12h^2}[-U_{j-2} + 16U_{j-1} - 30U_j + 16U_{j+1} - U_{j+2}] \tag{2.109}$$

gives a fourth order accurate approximation to $u''(x_j)$. Note that this formula can be easily found in MATLAB by `fdcoeffV(2,0,-2:2)`.

For the BVP $u''(x) = f(x)$ on a grid with m interior points, this approximation can be used at grid points $j = 2,\ 3,\ \ldots,\ m-1$ but not for $j = 1$ or $j = m$. At these points we must use methods with only one point in the stencil to the left or right, respectively. Suitable formulas can again be found using `fdcoeffV`; for example,

$$\frac{1}{12h^2}[11U_0 - 20U_1 + 6U_2 + 4U_3 - U_4] \tag{2.110}$$

is a third order accurate formula for $u''(x_1)$ and

$$\frac{1}{12h^2}[10U_0 - 15U_1 - 4U_2 + 14U_3 - 6U_4 + U_5] \tag{2.111}$$

is fourth order accurate. As in the case of the second order methods discussed above, we can typically get away with one less order at one point near the boundary, but somewhat better accuracy is expected if (2.111) is used.

These methods are easily extended to nonuniform grids using the same approach as in Section 2.18. The matrix is essentially pentadiagonal except in the first and last two rows, and using sparse matrix storage ensures that the system is solved in $O(m)$ operations. Fourth order accuracy is observed as long as the grid is smoothly varying.

2.20.2 Extrapolation methods

Another approach to obtaining fourth order accuracy is to use the second order accurate method on two different grids, with spacing h (the coarse grid) and $h/2$ (the fine grid), and then to extrapolate in h to obtain a better approximation on the coarse grid that turns out to have $O(h^4)$ errors for this problem.

Denote the coarse grid solution by

$$U_j \approx u(jh), \qquad i = 1,\ 2,\ \ldots,\ m,$$

and the fine grid solution by

$$V_i \approx u(ih/2), \qquad i = 1,\ 2,\ \ldots,\ 2m+1,$$

and note that both U_j and V_{2j} approximate $u(jh)$. Because the method is a centered second order accurate method, it can be shown that the error has the form of an even-order expansion in powers of h,

$$U_j - u(jh) = C_2h^2 + C_4h^4 + C_6h^6 + \cdots, \tag{2.112}$$

provided $u(x)$ is sufficiently smooth. The coefficients $C_2,\ C_4,\ \ldots$ depend on high order derivatives of u but are independent of h at each fixed point jh. (This follows from the fact that the local truncation error has an expansion of this form and the fact that the inverse matrix has columns that are an exact discretization of the Green's function, as shown in Section 2.11, but we omit the details of justifying this.)

On the fine grid we therefore have an error of the form

$$\begin{aligned} V_{2j} - u(jh) &= C_2\left(\frac{h}{2}\right)^2 + C_4\left(\frac{h}{2}\right)^4 + C_6\left(\frac{h}{2}\right)^6 + \cdots \\ &= \frac{1}{4}C_2h^2 + \frac{1}{16}C_4h^4 + \frac{1}{64}C_6h^6 + \cdots . \end{aligned} \tag{2.113}$$

The extrapolated value is given by

$$\bar{U}_j = \frac{1}{3}(4V_{2j} - U_j), \tag{2.114}$$

which is chosen so that the h^2 term of the errors cancels out and we obtain

$$\bar{U}_j - u(jh) = \frac{1}{3}\left(\frac{1}{4} - 1\right)C_4h^4 + O(h^6). \tag{2.115}$$

The result has fourth order accuracy as h is reduced and a much smaller error than either U_j or V_{2j} (provided C_4h^2 is not larger than C_2, and usually it is much smaller).

Implementing extrapolation requires solving the problem twice, once on the coarse grid and once on the fine grid, but to obtain similar accuracy with the second order method alone would require a far finer grid than either of these and therefore much more work.

The extrapolation method is more complicated to implement than the fourth order method described in Section 2.20.1, and for this simple one-dimensional boundary value problem it is probably easier to use the fourth order method directly. For more complicated problems, particularly in more than one dimension, developing a higher order method may be more difficult and extrapolation is often a powerful tool.

It is also possible to extrapolate further to obtain higher order accurate approximations. If we also solve the problem on a grid with spacing $h/4$, then this solution can be combined with V to obtain a fourth order accurate approximation on the $(h/2)$-grid. This can be combined with $\bar{U}$ determined above to eliminate the $O(h^4)$ error and obtain a sixth order accurate approximation on the original grid.

2.20.3 Deferred corrections

Another way to combine two different numerical solutions to obtain a higher order accurate approximation, called deferred corrections, has the advantage that it solves both of the problems on the same grid rather than refining the grid as in the extrapolation method. We first solve the system $AU = F$ of Section 2.4 to obtain the second order accurate approximation U. Recall that the global error $E = U - \hat{U}$ satisfies the difference equation (2.15),

$$AE = -\tau, \tag{2.116}$$

where τ is the local truncation error. Suppose we knew the vector τ. Then we could solve the system (2.116) to obtain the global error E and hence obtain the exact solution $\hat{U}$ as $\hat{U} = U - E$. We cannot do this exactly because the local truncation error has the form

$$\tau_j = \frac{1}{12}h^2u''''(x_j) + O(h^4)$$

and depends on the exact solution, which we do not know. However, from the approximate solution U we can estimate τ by approximating the fourth derivative of U.

For the simple problem $u''(x) = f(x)$ that we are now considering we have $u''''(x) = f''(x)$, and so the local truncation error can be estimated directly from the given function $f(x)$. In fact for this simple problem we can avoid solving the problem twice by simply modifying the right-hand side of the original problem $AU = F$ by setting

$$F_j = f(x_j) + \frac{1}{12}h^2 f''(x_j) \tag{2.117}$$

with boundary terms added at $j = 1$ and $j = m$. Solving $AU = F$ then gives a fourth order accurate solution directly. An analogue of this for the two-dimensional Poisson problem is discussed in Section 3.5.

For other problems, we would typically have to use the computed solution U to estimate τ_j and then solve a second problem to estimate E. This general approach is called the method of deferred corrections. In summary, the procedure is to use the approximate solution to estimate the local truncation error and then solve an auxiliary problem of the form (2.116) to estimate the global error. The global error estimate can then be used to improve the approximate solution. For more details see, e.g., [54], [4].

2.21 Spectral methods

The term *spectral method* generally refers to a numerical method that is capable (under suitable smoothness conditions) of converging at a rate that is faster than polynomial in the mesh width h. Originally the term was more precisely defined. In the classical spectral method the solution to the differential equation is approximated by a function $U(x)$ that is a linear combination of a finite set of orthogonal basis functions, say,

$$U(x) = \sum_{j=1}^{N} c_j \phi_j(x), \tag{2.118}$$

and the coefficients chosen to minimize an appropriate norm of the residual function ($= U''(x) - f(x)$ for the simple BVP (2.4)). This is sometimes called a *Galerkin approach*. The method we discuss in this section takes a different approach and can be viewed as expressing $U(x)$ as in (2.118) but then requiring $U''(x_i) = f(x_i)$ at $N - 2$ grid points, along with the two boundary conditions. The differential equation will be exactly satisfied at the grid points by the function $U(x)$, although in between the grid points the ODE generally will not be satisfied. This is called *collocation* and the method presented below is sometimes called a *spectral collocation* or *pseudospectral* method.

In Section 2.20.1 we observed that the second order accurate method could be extended to obtain fourth order accuracy by using more points in the stencil at every grid point, yielding better approximations to the second derivative. We can increase the order further by using even wider stencils.

Suppose we take this idea to its logical conclusion and use the data at *all* the grid points in the domain in order to approximate the derivative at each point. This is easy to try in MATLAB using a simple extension of the approach discussed in Example 2.3. For

the test problem considered with a Neumann boundary condition at the left boundary and a Dirichlet condition at the right, the code from Example 2.3 can be rewritten to use all the grid values in every stencil as

```
A = zeros(m+2);   % A is dense
% first row for Neumann BC, approximates u'(x(1))
A(1,:) = fdcoeffF(1, x(1), x);
% interior rows approximate u''(x(i))
for i=2:m+1
   A(i,:) = fdcoeffF(2, x(i), x);
   end
% last row for Dirichlet BC, approximates u(x(m+2))
A(m+2,:) = fdcoeffF(0,x(m+2),x);
```

We have also switched from using `fdcoeffV` to the more stable `fdcoeffF`, as discussed in Section 1.5.

Note that the matrix will now be dense, since each finite difference stencil involves all the grid points. Recall that `x` is a vector of length $m + 2$ containing all the grid points, so each vector returned by a call to `fdcoeffF` is a full row of the matrix A.

If we apply the resulting A to a vector $U = [U_1\ U_2\ \cdots U_{m+2}]^T$, the values in $W = AU$ will simply be

$$\begin{aligned} W_1 &= p'(x_1), \\ W_i &= p''(x_i) \quad \text{for } i = 2,\ \ldots,\ m+1, \\ W_{m+2} &= p(x_{m+2}), \end{aligned} \tag{2.119}$$

where we're now using the MATLAB indexing convention as in the code and $p(x)$ is the unique polynomial of degree $m + 1$ that interpolates the $m + 2$ data points in U. The same high degree polynomial is used to approximate the derivatives at every grid point.

What sort of accuracy might we hope for? Interpolation through n points generally gives $O(h^{(n-2)})$ accuracy for the second derivative, or one higher order if the stencil is symmetric. We are now interpolating through $m + 2$ points, where $m = O(1/h)$, as we refine the grid, so we might hope that the approximation is $O(h^{1/h})$ accurate. Note that $h^{1/h}$ approaches zero faster than any fixed power of h as $h \to 0$. So we might expect very rapid convergence and small errors.

However, it is not at all clear that we will really achieve the accuracy suggested by the argument above, since increasing the number of interpolation points spread over a fixed interval as $h \to 0$ is qualitatively different than interpolating at a fixed number of points that are all approaching a single point as $h \to 0$. In particular, if we take the points x_i to be equally spaced, then we generally expect to obtain disastrous results. High order polynomial interpolation at equally spaced points on a fixed interval typically leads to a highly oscillatory polynomial that does not approximate the underlying smooth function well at all away from the interpolation points (the Runge phenomenon), and it becomes exponentially *worse* as the grid is refined and the degree increases. Approximating second derivatives by twice differentiating such a function would not be wise and would lead to an unstable method.

This idea can be saved, however, by choosing the grid points to be clustered near the ends of the interval in a particular manner. A very popular choice, which can be shown to be optimal in a certain sense, is to use the extreme points of the Chebyshev polynomial of degree $m + 1$, shifted to the interval $[a, b]$. The expression (B.25) in Appendix B gives the extreme points of $T_m(x)$ on the interval $[-1, 1]$. Shifting to the desired interval, changing m to $m + 1$, and reordering them properly gives the *Chebyshev grid points*

$$x_i = a + \frac{1}{2}(b - a)(1 + \cos(\pi(1 - z_i))) \quad \text{for } i = 0,\ 1, \ldots,\ m + 1, \tag{2.120}$$

where the z_i are again $m + 2$ equally spaced points in the unit interval, $z_i = i/(m + 1)$ for $i = 0,\ 1,\ \ldots,\ m + 1$.

The resulting method is called a *Chebyshev spectral method* (or pseudospectral/ spectral collocation method). For many problems these methods give remarkably good accuracy with relatively few grid points. This is certainly true for the simple boundary value problem $u''(x) = f(x)$, as the following example illustrates.

Example 2.4. Figure 2.9 shows the error as a function of h for three methods we have discussed on the simplest BVP of the form

$$\begin{aligned} u''(x) &= e^x \quad \text{for } 0 \le x \le 3, \\ u(0) &= -5, \quad u(3) = 3. \end{aligned} \tag{2.121}$$

The error behaves in a textbook fashion: the errors for the second order method of Section 2.4 lie on a line with slope 2 (in this log-log plot), and those obtained with the fourth order method of Section 2.20.1 lie on a line with slope 4. The Chebyshev pseudospectral method behaves extremely well for this problem; an error less than 10^{-6} is already

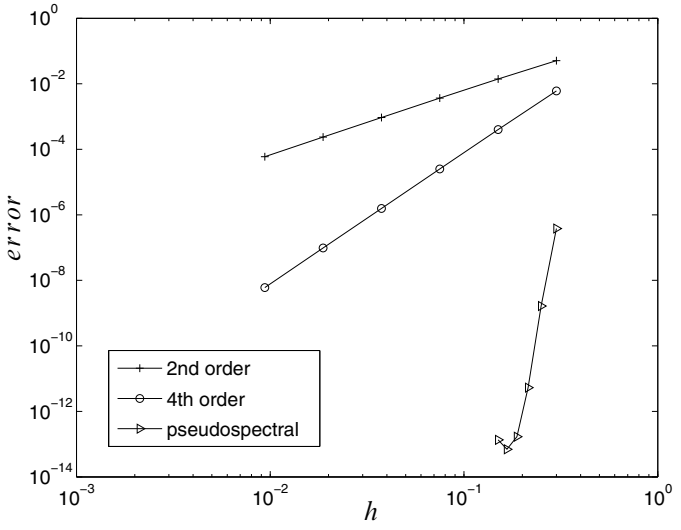

Figure 2.9. *Error as a function of h for two finite difference methods and the Chebyshev pseudospectral method on* (2.121).

observed on the coarsest grid (with $m = 10$) and rounding errors become a problem by $m = 20$. The finest grids used for the finite difference methods in this figure had $m = 320$ points.

For many problems, spectral or pseudospectral methods are well suited and should be seriously considered, although they can have difficulties of their own on realistic nonlinear problems in complicated geometries, or for problems where the solution is not sufficiently smooth. In fact the solution is required to be analytic in some region of the complex plane surrounding the interval over which the solution is being computed in order to get full "spectral accuracy."

Note that the polynomial $p(x)$ in (2.119) is exactly the function $U(x)$ from (2.118), although in the way we have presented the method we do not explicitly compute the coefficients of this polynomial in terms of polynomial basis functions. One could compute this interpolating polynomial if desired once the grid values U_j are known. This may be useful if one needs to approximate the solution $u(x)$ at many more points in the interval than were used in solving the BVP.

For some problems it is natural to use Fourier series representations for the function $U(x)$ in (2.118) rather than polynomials, in particular for problems with periodic boundary conditions. In this case the dense matrix systems that arise can generally be solved using fast Fourier transform (FFT) algorithms. The FFT also can be used in solving problems with Chebyshev polynomials because of the close relation between these polynomials and trigonometric functions, as discussed briefly in Section B.3.2. In many applications, however, a spectral method uses sufficiently few grid points that using direct solvers (Gaussian elimination) is a reasonable approach.

The analysis and proper application of pseudospectral methods goes well beyond the scope of this book. See, for example, [10], [14], [29], [38], or [90] for more thorough introductions to spectral methods.

Note also that spectral approximation of derivatives often is applied only in spatial dimensions. For time-dependent problems, a time-stepping procedure is often based on finite difference methods, of the sort developed in Part II of this book. For PDEs this time stepping may be coupled with a spectral approximation of the spatial derivatives, a topic we briefly touch on in Sections 9.9 and 10.13. One time-stepping procedure that has the flavor of a spectral procedure in time is the recent spectral deferred correction method presented in [28].