5

Finite Difference Methods

The methods considered in this chapter are conceptually different from those of the previous chapter. Here, no initial value problems are explicitly integrated. Rather, an approximate solution representation is sought over the entire interval of interest. Thus, these methods are sometimes referred to as *global methods*. We shall see, nonetheless, that there is a close theoretical relationship between these methods and initial value techniques. This is particularly highlighted in Chapters 6 and 7.

Compared to initial value methods, the class of finite difference methods presented in this chapter is more uniform and admits a more unified theory. Also, the development here must begin with more basic concepts, not assuming that any major part of the process is already taken care of.

5.1 INTRODUCTION

The basic steps of a finite difference method are outlined as follows.

Outline: Finite difference method

Input: A BVP

Output: An approximate solution $\mathbf{y}_\pi(x)$

1. Choose a mesh

$$\pi\colon a = x_1 < x_2 < \cdots < x_N < x_{N+1} = b \tag{5.1}$$

Approximate solution values are then sought at these mesh points x_j.

2. Form a set of algebraic equations for the approximate solution values by replacing derivatives with difference quotients in the differential equations and boundary conditions that the exact solution satisfies.

3. Solve the resulting system of equations for the approximate solution. This gives a set of discrete solution values $\mathbf{y}_i \equiv \mathbf{y}_\pi(x_i)$ and, optionally, interpolation can be used to construct $\mathbf{y}_\pi(x)$ for any $x \in [a, b]$. □

A simple illustration of the above process is given in Section 5.1.1. We give there a simple method for a second-order linear problem using a uniform mesh. This gives us the opportunity to introduce some concepts on a simple structure, as well as to discuss a model problem frequently occurring in the solution of partial differential equations (PDEs).

However, the method of Section 5.1.1 is not easy to generalize. To be able to come up with general methods, we consider in most of this chapter first-order systems of ODEs. Two simple difference schemes for linear ODEs are introduced in Section 5.1.2. These are the *midpoint* scheme and the *trapezoidal* scheme of (2.139), (2.140). Their application for nonlinear problems is described in Section 5.1.3.

Let us recall from Section 2.7 that the midpoint and trapezoidal schemes are *one-step* schemes. A unified theory of stability and convergence for one-step schemes is given in Section 5.2 for linear and nonlinear problems.

A disadvantage of the basic schemes introduced in this section is that they all have only second-order accuracy. This is not very efficient for many applications, where higher-order methods can be exploited to obtain accurate answers on relatively coarse meshes. Two types of extensions are possible. The first is to define families of higher-order methods, of which the basic schemes of Section 5.1.2 are the simplest members. This is done in Section 5.3 for first-order ODEs, where implicit Runge-Kutta schemes are introduced. Their relationship with certain collocation methods is explored in Section 5.4, and a convergence analysis is given.

A second approach towards obtaining higher-order methods is to take a basic scheme and to accelerate its convergence by applying extrapolation or deferred correction to it, repeatedly if necessary. These techniques are considered in Section 5.5.

The treatment in Sections 5.2–5.5 is for first-order systems of ODEs. In Section 5.6 we consider higher-order ODEs. Although many different methods have been proposed in the literature, we consider only an extension of the simple second-order scheme of Section 5.1.1, because of its usefulness as a special-purpose scheme and extendability to PDEs, and collocation, because of its generality and systematic presentation.

In the final section of this chapter, we briefly consider finite element methods. A number of books and *many* papers are available on this topic. However, we feel that the main power of finite element methods is in PDEs, rather than ODEs. Hence we give in this book a brief description of some main ideas only.

In this chapter we use, unless explicitly noted otherwise, max or sup norms. We also use c as a generic constant.

5.1.1 A simple scheme for a second-order problem.

Consider a linear scalar boundary value problem, as in (4.1)

$$u'' - p(x)u' - r(x)u = q(x) \qquad 0 < x < 1 \tag{5.2a}$$

$$u(0) = \beta_1, \qquad u(1) = \beta_2 \tag{5.2b}$$

where $p(x)$, $r(x)$, and $q(x)$ are continuous functions on [0, 1]. We illustrate a simple difference scheme on a uniform mesh for this problem: Thus, in (5.1) we take $x_i = (i-1)h$, $i = 1, 2, \ldots, N+1$, where $h = 1/N$ (so π is a function of h), and we seek a mesh function $u_\pi \equiv \{u_j\}_{j=1}^{N+1}$ such that $u_i \sim u(x_i)$, $i = 1, 2, \ldots, N+1$, where $u(x)$ is the exact solution of (5.2), assumed to exist.

By Taylor series,

$$u(x_i \pm h) = u(x_i) \pm hu'(x_i) + \frac{h^2}{2}u''(x_i) \pm \frac{h^3}{6}u'''(x_i) + O(h^4)$$

so we can express first and second derivatives as

$$u'(x_i) = \frac{u(x_{i+1}) - u(x_{i-1})}{2h} + O(h^2) \tag{5.3a}$$

$$u''(x_i) = \frac{u(x_{i+1}) - 2u(x_i) + u(x_{i-1})}{h^2} + O(h^2) \tag{5.3b}$$

Using (5.3), we can approximate the ODE (5.2a) at $x = x_i$ by

$$\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} - p(x_i)\frac{u_{i+1} - u_{i-1}}{2h} - r(x_i)u_i = q(x_i) \qquad 2 \le i \le N \tag{5.4a}$$

and the BC (5.2b) give

$$u_1 = \beta_1, \qquad u_{N+1} = \beta_2 \tag{5.4b}$$

This completes the second step of the outline above.

For the third step of the general method outline note that (5.4a, b) are $N+1$ linear equations for $N+1$ unknowns,

$$\mathbf{A}\mathbf{u}_\pi = \mathbf{b} \tag{5.5a}$$

where $\mathbf{A}$ is a *tridiagonal* matrix given by

$$\mathbf{A} = \begin{bmatrix} a_1 & c_1 & & & & \\ b_2 & a_2 & c_2 & & & \\ & b_3 & a_3 & c_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & b_N & a_N & c_N \\ & & & & b_{N+1} & a_{N+1} \end{bmatrix} \tag{5.5b}$$

$$a_i = -(\frac{2}{h^2} + r(x_i)), \qquad b_i = \frac{1}{h^2} + \frac{p(x_i)}{2h}, \qquad c_i = \frac{1}{h^2} - \frac{p(x_i)}{2h}, \qquad 2 \le i \le N$$

$$a_1 = a_{N+1} = 1, \qquad c_1 = b_{N+1} = 0$$

$$\mathbf{u}_\pi = (u_1, u_2, \ldots, u_{N+1})^T, \qquad \mathbf{b} = (\beta_1, q(x_2), \ldots, q(x_N), \beta_2)^T \tag{5.5c}$$

Before solving (5.5), we could perform row equilibration by multiplying the 2nd through the Nth equations by h^2. The $(i+1)^{st}$ row of the equilibrated **A** would then have

$$1 + \frac{h}{2}p(x_i) \qquad -2 - h^2 r(x_i) \qquad 1 - \frac{h}{2}p(x_i)$$

as its nonzero elements.

Finite difference schemes generally lead to systems of algebraic equations involving banded matrices, and in this particular case the matrix **A** is tridiagonal. Gaussian elimination strategies which utilize this special structure are very simple and efficient, cf. Section 2.2. No pivoting is needed if **A** is diagonally dominant, which occurs here if $h \le \frac{2}{\|p\|}$ and $r \ge 0$ (why?).

Thus the approximate mesh function $\mathbf{u}_\pi$ exists, at least under the restrictions above. But does it really approximate the values $u(x_i)$, and how well? To answer this we turn to the equations (5.4), which are supposed to approximate the BVP (5.2). Since we cannot ask how well $\mathbf{u}_\pi$ satisfies the ODE (5.2a) (the mesh function is only defined at a discrete set of points), we ask how well the exact solution $u(x)$ satisfies the difference equations (5.4).

It is convenient here to use simple operator notation. Thus, the *differential operator* L for (5.2a) is

$$Lu(x) := u''(x) - p(x)u'(x) - r(x)u(x) \tag{5.6a}$$

while the corresponding *difference operator* L_π for (5.4a) is

$$L_\pi u(x) := \frac{u(x+h) - 2u(x) + u(x-h)}{h^2} - p(x)\frac{u(x+h) - u(x-h)}{2h} - r(x)u(x) \tag{5.6b}$$

So, for instance, we may write (5.4a) as

$$L_\pi u_i = q(x_i) \qquad 2 \le i \le N \tag{5.6c}$$

For any smooth function $v(x)$, the *local truncation error* $\tau_i[v]$ is defined by

$$\tau_i[v] := L_\pi v(x_i) - Lv(x_i) \qquad 2 \le i \le N \tag{5.6d}$$

By our construction [see (5.3)] it is clear that if $v \in C^{(4)}[0,1]$, then

$$\tau[v] := \max_{2 \le i \le N} |\tau_i[v]| \le ch^2, \qquad c = \text{constant} \tag{5.7}$$

Thus $\tau[v] \to 0$ as $h \to 0$. Furthermore, there is no error in the approximation of the BC in (5.4b). We may then say that the difference operator L_π approaches the differential operator L in the limit as $h \to 0$. A difference method satisfying this is called *con-*

sistent. From (5.7) we see that the particular method described here is not only consistent but is also of *order* 2; i. e., $\tau[v]$ approaches 0 as fast as h^2.

Consistency is necessary, but not sufficient, to get discrete *convergence*, i. e., that the approximate solution values at a point x_i approach $u(x_i)$ as $h \to 0$ with ih kept fixed. Since consistency ensures that the difference operator is "close" to the differential operator, all we need in addition is to be able to bound the solution values in terms of the values of the operator defining it. For the solution $u(x)$ of (5.2), this is assumed as part of the well-posedness of the problem to be solved. Indeed, from Section 3.2 we have

$$\| u \| \leq \kappa \max \{ |\beta_1|, |\beta_2|, \| Lu \| \} \tag{5.8}$$

where the BVP is well-conditioned if κ is of "moderate" size. We expect the finite difference scheme to preserve this conditioning (at least for h small) and say that it is *stable* if for all meshes π with h sufficiently small and all corresponding mesh functions v_π,

$$|v_i| \leq K \max \{ |v_1|, |v_{N+1}|, \max_{2 \leq j \leq N} |L_\pi v_j| \} \qquad 1 \leq i \leq N+1 \tag{5.9}$$

with K a constant (independent of π) which is of moderate size when κ is. By (5.5), (5.9) is obtained iff

$$\| \mathbf{A}^{-1} \| \leq K \tag{5.10}$$

Note that stability is a property inherited from the differential operator (plus the BC) and does not depend on the inhomogeneities $q(x)$ and β. Also, stability in this chapter is an *absolute* concept, analogous to well-conditioning of the differential problem.

For our scheme (5.4), it is not difficult to show stability when $h \leq \dfrac{2}{\| p \|}$ and $r(x)$ is positive and bounded away from 0 (cf. Exercise 1). This is actually a weak result, because K is not related to κ in a natural way. But we shall reserve attempts at a more careful stability analysis to the more general schemes in Section 5.2. Here, let us see what to do with stability and consistency, once we have them.

For the global error

$$e_i := u(x_i) - u_i \qquad 1 \leq i \leq N+1 \tag{5.11a}$$

we have by (5.6), (5.4), (5.2)

$$L_\pi e_i = L_\pi u(x_i) - q(x_i) = \tau_i[u] \tag{5.11b}$$

$$e_1 = e_{N+1} = 0 \tag{5.11c}$$

Thus, the error satisfies the difference equations with the local truncation error as an inhomogeneity! We now use the stability bound (5.9) for e_i to obtain, with (5.7),

$$|e_i| \leq K \tau[u] \leq Kch^2 \qquad 2 \leq i \leq N \tag{5.11d}$$

if u has a bounded 4th derivative.

The finite difference scheme is said to be *convergent* if[1]

$$\max_{1 \le i \le N+1} |u(x_i) - u_i| \to 0 \qquad \text{as } h \to 0 \tag{5.12}$$

Here we have obtained the basic result that

$$\text{consistency} + \text{stability} \Rightarrow \text{convergence}$$

Moreover, in (5.11d) we see how fast e_i approaches 0 as $h \to 0$. The *order* of the method is p if $e_i \to 0$ like $0(h^p)$.

5.1.2 Simple one-step schemes for linear systems

The attraction of the finite difference scheme of the previous section is in its simplicity. When it can be applied, the scheme is very efficient, and so it is useful as a basic discretization rule in PDEs as well. However, it is too restricted for most applications of ODEs, because it has been applied only to one second-order differential equation. Note also that the meshes are uniform. Most nontrivial applications arise as mixed-order systems of differential equations, and many require very dense (fine) meshes in some sections of the interval of definition. If such a mesh is required to be uniform then it becomes dense everywhere, making the computation very expensive. A generalization of the simple scheme (5.6) for nonuniform meshes is considered in Section 5.6.1. A few more involved issues arise there.

We consider now the linear first-order system (3.2), (3.19) which we rewrite here,

$$\mathbf{y}' = A(x)\mathbf{y} + \mathbf{q}(x) \qquad a < x < b \tag{5.13a}$$

$$B_a \mathbf{y}(a) + B_b \mathbf{y}(b) = \boldsymbol{\beta} \tag{5.13b}$$

where $A(x)$, B_a and $B_b \in \mathbf{R}^{n \times n}$, and we seek numerical methods which work equally well for nonuniform meshes. This naturally leads to one-step schemes (cf. Section 2.7), i.e., schemes which define the difference operator based only on values related to one subinterval $[x_i, x_{i+1}]$ of the mesh π of (5.1) at a time. The two simplest such finite difference schemes are the midpoint and the trapezoidal schemes, discussed next.

For a (generally nonuniform) mesh π of (5.1), a discrete numerical solution $\mathbf{y}_\pi = (\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_{N+1})^T \in \mathbf{R}^{n(N+1)}$ is sought, where $\mathbf{y}_i$ is to approximate component-wise the exact solution $\mathbf{y}(x)$ at $x = x_i$. [A unique solution to (5.13) is assumed to exist.] The numerical solution (in all methods based on one-step schemes) is required to satisfy the boundary conditions:

$$B_a \mathbf{y}_1 + B_b \mathbf{y}_{N+1} = \boldsymbol{\beta} \tag{5.14}$$

For the interior mesh points, two difference schemes are presented [cf. (2.139), (2.140)]. For each subinterval $[x_i, x_{i+1}]$ of π, the derivative in (5.13a) is replaced by $\dfrac{\mathbf{y}_{i+1} - \mathbf{y}_i}{h_i}$. This approximation is centered at $x_{i+1/2} := x_i + \dfrac{1}{2}h_i$, with $h_i := x_{i+1} - x_i$, i.e., at the middle of the subinterval. Then $A(x)\mathbf{y}(x) + \mathbf{q}(x)$ is approximated by a centered approximation, yielding second-order accuracy. The *trapezoidal* scheme is defined by

[1] In (5.12) we have preferred simplicity over rigor. We really consider the method on a family of meshes with $h \to 0$, and vary i with h so that $\hat{x} = (i-1)h$ remains fixed. The sequence of approximate solution values u_i thus obtained should approach $u(\hat{x})$.

$$\frac{\mathbf{y}_{i+1} - \mathbf{y}_i}{h_i} = \frac{1}{2}[A(x_{i+1})\mathbf{y}_{i+1} + A(x_i)\mathbf{y}_i] + \frac{1}{2}[\mathbf{q}(x_{i+1}) + \mathbf{q}(x_i)] \qquad 1 \le i \le N \tag{5.15}$$

and the *midpoint* scheme is defined by

$$\frac{\mathbf{y}_{i+1} - \mathbf{y}_i}{h_i} = \frac{1}{2}A(x_{i+1/2})(\mathbf{y}_{i+1} + \mathbf{y}_i) + \mathbf{q}(x_{i+1/2}) \qquad 1 \le i \le N \tag{5.16}$$

The latter scheme (5.16) is also known as the *box scheme*. In matrix form, both of these methods can be written as

$$\mathbf{A}\mathbf{y}_\pi = \hat{\boldsymbol{\beta}} \tag{5.17a}$$

and, in detail,

$$\begin{bmatrix} S_1 & R_1 & & & \\ & S_2 & R_2 & & \\ & & \ddots & & \\ & & & S_N & R_N \\ B_a & & & & B_b \end{bmatrix} \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_{N+1} \end{bmatrix} = \begin{bmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \\ \vdots \\ \mathbf{q}_N \\ \boldsymbol{\beta} \end{bmatrix} \tag{5.17b}$$

where S_i, R_i are $n \times n$ matrices. For the trapezoidal scheme

$$S_i = -h_i^{-1}I - \frac{1}{2}A(x_i), \qquad R_i = h_i^{-1}I - \frac{1}{2}A(x_{i+1}) \qquad 1 \le i \le N$$
$$\mathbf{q}_i = \frac{1}{2}[\mathbf{q}(x_{i+1}) + \mathbf{q}(x_i)] \tag{5.18}$$

while for the midpoint scheme

$$S_i = -h_i^{-1}I - \frac{1}{2}A(x_{i+1/2}), \quad R_i = h_i^{-1}I - \frac{1}{2}A(x_{i+1/2}) \qquad 1 \le i \le N$$
$$\mathbf{q}_i = \mathbf{q}(x_{i+1/2}) \tag{5.19}$$

Note that $\mathbf{A}$ has the same block structure as the matrices $\mathbf{A}$ for multiple shooting and stabilized march discussed in Chapter 4 [see (4.37)]. A similar block structure also arises from other one-step schemes, as will be discussed in Sections 5.3 and 5.6.

In Fig. 5.1a we depict the structure of $\mathbf{A}$ for the case $n = 2$, $N = 3$. Here, the symbol $\times$ stands for a possibly nonzero element in the matrix. If the BC are separated, as is often the case in applications, then it is natural to move the BC at the left endpoint a to the top of the matrix (and split $\boldsymbol{\beta}$ accordingly). Thus, if $n = 2$ and we have one boundary condition at each end, then we obtain the structure depicted in Fig. 5.1b.

The existence of an approximate solution depends on the invertibility of $\mathbf{A}$. This in turn is a weaker requirement than stability, discussed in the next section. Once we show stability of the two schemes, the existence of the approximate solutions will follow. Here, we just give some numerical examples. Before doing that, we remark that the solution of the large sparse system of equations (5.17) is discussed in Chapter 7 rather than here (see also Exercise 3), since similar considerations hold for a variety of different discretization methods.

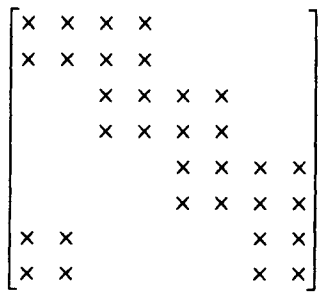

Figure 5.1a. Structure of **A**.

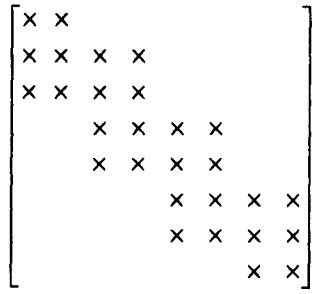

Figure 5.1b. **A** with separated BC

Example 5.1

When a partial differential equation is reduced to an ordinary one by (cylindrical) symmetry considerations, often a harmless singularity appears in the coefficient matrix $A(x)$. As a simple example, consider the problem

$$u'' = -\frac{1}{x}u' + \left(\frac{8}{8-x^2}\right)^2$$

$$u'(0) = u(1) = 0$$

The exact solution

$$u(x) = 2 \ln \left(\frac{7}{8-x^2}\right)$$

is an analytic function on [0, 1], but even without knowing it we can use l'Hôpital's rule to find that

$$\frac{u'(x)}{x} \to u''(0) \qquad \text{as} \quad x \to 0$$

so at $x = 0$ the differential equation behaves regularly and reads

$$u''(0) = 1/2$$

This modification of the differential equation at $x = 0$ needs to be explicitly specified when the trapezoidal scheme (but not the midpoint scheme) is used, in order to avoid a division by 0 in the computation.

Converting the problem to a first-order system for $y_1 = u$ and $y_2 = u'$, we obtain

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix}' = \begin{pmatrix} 0 & 1 \\ 0 & -1/x \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} + \begin{pmatrix} 0 \\ \left(\frac{8}{8-x^2}\right)^2 \end{pmatrix}$$

In Table 5.1 we list results when using uniform meshes of $N + 1$ points, $N = 10, 20, 40, 80$. The maximum absolute errors in y_1 and y_2 are listed under the heading $e(y_1)$ and $e(y_2)$, respectively, for the midpoint and trapezoidal schemes.

TABLE 5.1: Midpoint and trapezoidal schemes for Example 5.1.

	Midpoint		Trapezoidal	
N	$e(y_1)$	$e(y_2)$	$e(y_1)$	$e(y_2)$
10	.10−4	.44−3	.31−3	.29−3
20	.26−5	.11−3	.76−4	.73−4
40	.65−6	.27−4	.19−4	.18−4
80	.16−6	.69−5	.47−5	.45−5

The second-order convergence rate in h (i. e., an error reduction factor of 4 each time h is halved) is clearly demonstrated here. Observe also that for high accuracy, higher-order methods may be desirable. The application of shooting techniques to this problem is not so straightforward. □

Example 5.2

Consider, once again, Example 4.3. For $\lambda = 50$ and uniform meshes as in the previous example, the results listed in Table 5.2 clearly leave something to be desired: The errors are comparatively large. Inspecting the exact solution, we see that the higher derivatives of $y(x)$, to which the local truncation error relates, are much larger near the ends than in the middle, so a nonuniform mesh with h_i smaller near the interval ends is suggested. We pick such a mesh of size $N = 10$ as

$$\pi(\lambda) = \{0, \frac{1}{\lambda}, \frac{3}{\lambda}, \frac{8}{\lambda}, .25, .5, .75, 1-\frac{8}{\lambda}, 1-\frac{3}{\lambda}, 1-\frac{1}{\lambda}, 1\}$$

and then generate more meshes by repeatedly halving each subinterval of the current mesh. The results for $\lambda = 50$ and $\lambda = 5000$ are displayed in Table 5.2. Note the improvement over the uniform mesh results. An even better nonuniform mesh can be chosen to reduce the errors further, with the same number of mesh points. This topic of mesh selection is considered in Chapter 9.

Recall that, using initial value techniques for this example, only the full multiple shooting method with $N = 10$ performs satisfactorily for $\lambda = 50$. (Note that the adequate "mesh" of shooting points in Example 4.9 is uniform. It is constructed for different purposes than the finite difference mesh, despite the similar structure of A). With the mesh $\pi(\lambda)$ properly adjusted, the finite difference methods perform almost as well for $\lambda = 5000$ as for $\lambda = 50$ (see Section 10.3.2). In contrast, the performance of the standard multiple shooting method would rapidly deteriorate as λ increases.

TABLE 5.2: Midpoint and trapezoidal schemes for Example 5.2.

			Midpoint		Trapezoidal	
λ	mesh	N	$e(y_1)$	$e(y_2)$	$e(y_1)$	$e(y_2)$
50	uniform	10	.47	.45	.44	.44
		20	.20	.19	.19	.19
		40	.57-1	.55-1	.56-1	.56-1
		80	.-1	.12-1	.12-1	.12-1
50	nonuniform	10	.16	.51-1	.55	.56
		20	.42-1	.14-1	.15-1	.15-1
		40	.96-2	.34-2	.36-2	.36-2
		80	.24-2	.85-3	.91-3	.91-3
5000	nonuniform	10	.95	.95	.45	.45
		20	.82-1	.15-1	.15-1	.15-1
		40	.19-1	.36-2	.36-2	.36-2
		80	.44-2	.91-3	.91-3	.91-3

□

5.1.3 Simple schemes for nonlinear problems

Here we consider the (generally) nonlinear first-order system of n ODEs as in (3.1), (3.12),

$$\mathbf{y}' = \mathbf{f}(x, \mathbf{y}) \qquad a < x < b \tag{5.20a}$$

$$\mathbf{g}(\mathbf{y}(a), \mathbf{y}(b)) = \mathbf{0} \tag{5.20b}$$

For numerical approximation we again consider a mesh π of (5.1) and denote the vector of approximate solution values at mesh points by $\mathbf{y}_\pi$. The two schemes presented in Section 5.1.2 are extended in a straightforward way: The trapezoidal scheme is given by

$$\frac{\mathbf{y}_{i+1} - \mathbf{y}_i}{h_i} = \frac{1}{2}(\mathbf{f}(x_{i+1}, \mathbf{y}_{i+1}) + \mathbf{f}(x_i, \mathbf{y}_i)) \qquad 1 \le i \le N \tag{5.21}$$

$$\mathbf{g}(\mathbf{y}_1, \mathbf{y}_{N+1}) = \mathbf{0} \tag{5.22}$$

while the midpoint scheme is given by

$$\frac{\mathbf{y}_{i+1} - \mathbf{y}_i}{h_i} = \mathbf{f}(x_{i+1/2}, \frac{1}{2}(\mathbf{y}_i + \mathbf{y}_{i+1})) \qquad 1 \le i \le N \tag{5.23}$$

and (5.22).

Thus we obtain a system of $n(N+1)$ algebraic equations for the $n(N+1)$ unknowns $\mathbf{y}_\pi$. Unlike before, though, these equations are nonlinear. It is not uncommon to have, for instance, $n = 5$ and $N = 200$, yielding more than 1000 nonlinear equations! Fortunately, the Jacobian matrix of this system is rather sparse, as we shall see below.

The basic method for solving the nonlinear problems arising in this book is Newton's method (Section 2.3). It is also intimately related to basic questions of stability of the difference scheme and existence of its solution, as discussed in Section 5.2.2. For these reasons we next consider the application of Newton's method in the context of a one-step difference scheme for (5.20). More sophisticated practical considerations and modifications are deferred to Chapter 8.

Recall from Sections 2.3 and 4.6.1 that for the system of equations

$$\mathbf{F}(\mathbf{s}) = \mathbf{0} \tag{5.24a}$$

Newton's method is a fixed-point iteration

$$\mathbf{s}^{m+1} = \mathbf{G}(\mathbf{s}^m) \qquad m = 0, 1, 2, \ldots,$$

with the iteration function $\mathbf{G}(\mathbf{s})$ defined by

$$\mathbf{G}(\mathbf{s}) := \mathbf{s} - [\mathbf{F}'(\mathbf{s})]^{-1}\mathbf{F}(\mathbf{s})$$

where $\mathbf{F}'(\mathbf{s}) = \dfrac{\partial \mathbf{F}(\mathbf{s})}{\partial \mathbf{s}}$ is the Jacobian matrix. This can be written as [cf. (4.85)].

$$\mathbf{F}'(\mathbf{s}^m)\,\boldsymbol{\xi} = -\mathbf{F}(\mathbf{s}^m) \tag{5.24b}$$

and

$$\mathbf{s}^{m+1} := \mathbf{s}^m + \boldsymbol{\xi} \tag{5.24c}$$

Let the nonlinear algebraic equations (5.24a) be given, for instance, by the trapezoidal scheme (5.21), (5.22), with $\mathbf{s} \equiv \mathbf{y}_\pi = (\mathbf{y}_1, \ldots, \mathbf{y}_{N+1})^T$, an $n(N+1)$ vector. Using a difference operator notation for (5.21), we obtain

$$\mathbf{N}_\pi \mathbf{y}_i := \frac{\mathbf{y}_{i+1} - \mathbf{y}_i}{h_i} - \frac{1}{2}(\mathbf{f}(x_{i+1}, \mathbf{y}_{i+1}) + \mathbf{f}(x_i, \mathbf{y}_i)) \tag{5.25a}$$

and $\mathbf{F}(\mathbf{s}) := (\mathbf{N}_\pi \mathbf{y}_1, \ldots, \mathbf{N}_\pi \mathbf{y}_N, \mathbf{g}(\mathbf{y}_1, \mathbf{y}_{N+1}))^T$. Newton's iteration (5.24b) gives

$$\frac{\mathbf{w}_{i+1} - \mathbf{w}_i}{h_i} - \frac{1}{2}[A(x_{i+1})\mathbf{w}_{i+1} + A(x_i)\mathbf{w}_i] = -\mathbf{N}_\pi \mathbf{y}_i^m \qquad 1 \le i \le N \tag{5.25b}$$

$$B_a \mathbf{w}_1 + B_b \mathbf{w}_{N+1} = -\mathbf{g}(\mathbf{y}_1^m, \mathbf{y}_{N+1}^m) \tag{5.25c}$$

Here $\boldsymbol{\xi} \equiv \mathbf{w}_\pi = (\mathbf{w}_1, \ldots, \mathbf{w}_{N+1})^T$, $\mathbf{y}_\pi^m$ are known values from a previous iteration ($\mathbf{y}_\pi^0$ is an initial guess) and

$$A(x_j) := \frac{\partial \mathbf{f}}{\partial \mathbf{y}}(x_j, \mathbf{y}_j^m) \tag{5.26a}$$

$$B_a = \frac{\partial \mathbf{g}(\mathbf{u}, \mathbf{v})}{\partial \mathbf{u}}, \qquad B_b = \frac{\partial \mathbf{g}(\mathbf{u}, \mathbf{v})}{\partial \mathbf{v}} \qquad \text{at } \mathbf{u} = \mathbf{y}_1^m, \quad \mathbf{v} = \mathbf{y}_{N+1}^m \tag{5.26b}$$

[The reader not yet used to the derivation of (5.26) should refer back to Section 3.1 at this point.] The next iterate is given, according to (5.24c), by $\mathbf{y}_i^{m+1} := \mathbf{y}_i^m + \mathbf{w}_i$, $i = 1, \ldots, N+1$.

The system (5.25) for the correction vector $\mathbf{w}_\pi$ is a linear system of equations which looks like a trapezoidal discretization of some linear problem. In fact, note that in each iteration we perform two operations in succession—discretization and linearization (i. e., Newton's method).

It now becomes apparent that the possibly large Jacobian matrix $\mathbf{F}'(\mathbf{s})$ is indeed sparse: It has the form of $\mathbf{A}$ in (5.17). Let us assume that we have an efficient linear system solver for (5.17). (This is discussed in Chapter 7.) Then we can present, as an example, the following crude algorithm.

Algorithm: Trapezoidal scheme with Newton's method

Input: A BVP (5.20), a mesh π, an initial guess of solution values $\mathbf{y}_\pi$ at mesh points, and a tolerance *TOL*.

Output: Solution values at mesh points, $\mathbf{y}_\pi$.

REPEAT

1. Generate B_a, B_b by (5.26b) and set $\boldsymbol{\beta} := -\mathbf{g}(\mathbf{y}_1, \mathbf{y}_{N+1})$.
2. FOR $i = 1, \ldots, N$ DO
   Generate S_i, R_i and $\mathbf{q}_i$ of (5.18) using (5.26a) and
   $$\mathbf{q}_i := -\mathbf{N}_\pi \mathbf{y}_i$$
   {At this point, $\mathbf{A}$ and $\hat{\boldsymbol{\beta}}$ of (5.17) have been generated for the current iteration}.
3. Solve $\mathbf{A}\mathbf{w}_\pi = \hat{\boldsymbol{\beta}}$ for $\mathbf{w}_\pi$ {using the linear system solver the existence of which has been postulated above}.
4. FOR $i = 1, \ldots, N + 1$ DO
   $$\mathbf{y}_i := \mathbf{y}_i + \mathbf{w}_i$$

UNTIL $|\mathbf{w}_\pi| \le TOL$ (or iteration limit exceeded).

□

A similar algorithm can be constructed, of course, with the midpoint scheme replacing the trapezoidal scheme. Thus, in step 1.2.1 we would use (5.19) instead of (5.18).

Example 5.3

Let us consider again the simple problem (4.5) of Example 4.1, converted to a first-order system as in Example 4.15. Thus we have (5.20) with

$$n = 2, \qquad \mathbf{y} = \begin{bmatrix} u \\ v \end{bmatrix}, \qquad \mathbf{f} = \begin{bmatrix} v \\ -\exp(u) \end{bmatrix}, \qquad \mathbf{g} = \begin{bmatrix} u(0) \\ u(1) \end{bmatrix}$$

and (5.25b, c) reads, with $\mathbf{y}_i^m = \begin{bmatrix} u_i^m \\ v_i^m \end{bmatrix}$ given $(1 \le i \le N + 1)$

$$\frac{\mathbf{w}_{i+1}-\mathbf{w}_i}{h_i} - \frac{1}{2}\left[\begin{pmatrix} 0 & 1 \\ -\exp(u_{i+1}^m) & 0 \end{pmatrix}\mathbf{w}_{i+1} + \begin{pmatrix} 0 & 1 \\ -\exp(u_i^m) & 0 \end{pmatrix}\mathbf{w}_i\right] = -\left[\frac{\mathbf{y}_{i+1}^m - \mathbf{y}_i^m}{h_i}\right.$$

$$\left. - \frac{1}{2}\begin{pmatrix} v_{i+1}^m \\ -\exp(u_{i+1}^m) \end{pmatrix} - \frac{1}{2}\begin{pmatrix} v_i^m \\ -\exp(u_i^m) \end{pmatrix}\right] \qquad 1 \le i \le N$$

$$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}\mathbf{w}_1 + \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}\mathbf{w}_{N+1} = -\begin{pmatrix} u_1^m \\ u_{N+1}^m \end{pmatrix}$$

These equations are *linear* for the unknowns $\mathbf{w}_\pi$.

Now, recall that this BVP has two solutions. The initial guess $\mathbf{y}_\pi^0$ determines to which of them (if at all) our algorithm converges. It turns out that taking a uniform mesh with stepsize h and $\mathbf{y}_\pi^0 \equiv \mathbf{0}$, the algorithm converges to the physically stable solution (cf. Example 4.15). Corresponding to Table 4.4 we obtain the numerical data in Table 5.3 below.

The observed reductions in the error as h is decreased are expected, since the trapezoidal and the midpoint schemes are second-order accurate. This is also the reason for the observed errors, which are "worse," for a given h, than the ones in Table 4.4: The shooting method was used with a fourth-order integrator (cf. Example 5.8 below). However, the amount of work per step h is smaller here.

Note that the number of Newton iterations needed for this simple example is minimal. The initial guess here is different from the one used for the shooting scheme in Table 4.4. □

Recall now the quasilinearization method, discussed in Section 2.3.4: In the arguments leading to (5.25) we discretized first and then linearized, but now consider reversing the order of these operations. Letting $\mathbf{y}^m(x)$ be an appropriately smooth function satisfying

$$\mathbf{y}^m(x_i) = \mathbf{y}_i^m \qquad i = 1, \ldots, N+1$$

and linearizing the problem as in Section 2.3.4, (2.73), we obtain

$$\mathbf{w}' - A(x)\mathbf{w} = -[(\mathbf{y}^m)'(x) - \mathbf{f}(x; \mathbf{y}^m(x))] \tag{5.27a}$$

TABLE 5.3: Nonlinear Example 3.2 with $\lambda = 1$

h	itn	e (trapezoidal)*
.2	2	.11−2
.1	2	.28−3
.05	2	.71−4
.025	2	.18−4
.0125	2	.44−5

* Similar results are obtained with the midpoint scheme.

$$B_a \mathbf{w}(a) + B_b \mathbf{w}(b) = -\mathbf{g}(\mathbf{y}^m(a), \mathbf{y}^m(b)) \tag{5.27b}$$

where $A(x)$, B_a and B_b are given in (5.26) with an obvious extension. Now use the trapezoidal scheme (5.15), (5.14) for the *linear* problem (5.27). The result is precisely (5.25)! Thus the two operations of linearization and discretization commute for the trapezoidal scheme. The next iterate $\mathbf{y}^{m+1}(x)$ for the quasilinearization process is again needed only at the mesh points π, and is obtained by $\mathbf{y}^{m+1}(x_i) := \mathbf{y}^m(x_i) + \mathbf{w}_i$

It is easy to verify that quasilinearization is equivalent to Newton's method applied to the discretized difference equations for most methods considered in this chapter. In practice, the quasilinearization approach is the recommended way of implementation because it leads to a modular program design. Once we have a program module to solve linear problems with a given method, it can be invoked repeatedly for each iteration of a nonlinear problem by first linearizing as in (5.27), (5.26).

5.2 CONSISTENCY, STABILITY, AND CONVERGENCE

We have already seen in Section 5.1.1 that when investigating the convergence properties of a finite difference scheme, two concepts naturally emerge. One is *consistency*, which says that the difference operator approaches the differential operator as the mesh width, or stepsize, $h \to 0$. More specifically, the *local truncation error*, which is the residual when substituting the exact solution of the BVP into the difference equations, is $O(h^p)$ for some $p \geq 1$. The other concept is that of *stability*, which requires that the inverse of the difference operator, including the BC, is suitably bounded.

Below we define these terms more precisely and discuss the conditions under which they can be guaranteed to hold. We do this for general one-step schemes for solving linear and nonlinear first-order systems of ODEs. It turns out that stability for such schemes essentially follows from consistency, under fairly weak assumptions. We shall show this for both linear and nonlinear BVPs and, in both cases, return to the fundamental theorem:

Theorem 5.28

$$\text{consistency} + \text{stability} \Rightarrow \text{convergence}$$

□

5.2.1 Linear problems

Consider the BVP (5.13). Here again it is convenient to speak in terms of the differential operator

$$\mathbf{L}\mathbf{y}(x) \equiv \mathbf{y}'(x) - A(x)\mathbf{y}(x) \qquad a < x < b \tag{5.29a}$$

and a corresponding one-step difference operator

$$\mathbf{L}_\pi \mathbf{y}_i \equiv \frac{\mathbf{y}_{i+1} - \mathbf{y}_i}{h_i} - \Psi(\mathbf{y}_i, \mathbf{y}_{i+1}; x_i, h_i) \qquad 1 \leq i \leq N \tag{5.29b}$$

For example, the midpoint scheme gives

$$\Psi(\mathbf{y}_i, \mathbf{y}_{i+1}; x_i, h_i) = \frac{1}{2}A(x_{i+1/2})(\mathbf{y}_i + \mathbf{y}_{i+1})$$

Using the notation of (5.17), we can write the general one-step scheme as

$$\mathbf{L}_\pi \mathbf{y}_i \equiv S_i \mathbf{y}_i + R_i \mathbf{y}_{i+1} = \mathbf{q}_i \qquad 1 \le i \le N \tag{5.30a}$$

$$B_a \mathbf{y}_1 + B_b \mathbf{y}_{N+1} = \boldsymbol{\beta} \tag{5.30b}$$

The two schemes which we have already seen are given by (5.30) with (5.18) or (5.19). Note that by the notation $\mathbf{L}_\pi \mathbf{y}_i$ we mean the ith segment of the nN vector resulting from applying $\mathbf{L}_\pi$ to the $n(N+1)$ vector $\mathbf{y}_\pi$. Similarly,

$$\mathbf{L}_\pi \mathbf{y}(x_i) \equiv S_i \mathbf{y}(x_i) + R_i \mathbf{y}(x_{i+1})$$

Definitions 5.31

The *local truncation error* $\boldsymbol{\tau}_i[\mathbf{y}]$ is defined as

$$\boldsymbol{\tau}_i[\mathbf{y}] := \mathbf{L}_\pi \mathbf{y}(x_i) - \mathbf{q}_i \qquad 1 \le i \le N \tag{5.31a}$$

The finite difference scheme (5.30) is said to be *consistent* of *order* p (p a positive integer) if for every (smooth) solution of (5.13a) there exist constants c and $h_0 > 0$ such that for all meshes π with $h := \max_{1 \le i \le N} h_i \le h_0$,

$$\tau[\mathbf{y}] := \max_{1 \le i \le N} |\boldsymbol{\tau}_i[\mathbf{y}]| \le ch^p \tag{5.31b}$$

[cf. (5.7)].

The scheme is said to be *stable* if there exist constants K and $h_0 > 0$ such that for all meshes π with $h \le h_0$ and all mesh functions $\mathbf{v}_\pi$,

$$|\mathbf{v}_i| \le K \max\{|B_a \mathbf{v}_1 + B_b \mathbf{v}_{N+1}|, \max_{1 \le j \le N} |\mathbf{L}_\pi \mathbf{v}_j|\} \qquad 1 \le i \le N+1 \tag{5.31c}$$

and K is of moderate size if the BVP conditioning constant κ is.

The solution of (5.30) is said to *converge* to the solution of the BVP (5.13) if

$$\max_{1 \le i \le N+1} |\mathbf{y}_i - \mathbf{y}(x_i)| \to 0 \qquad \text{as } h \to 0 \tag{5.31d}$$

□

(Note again that the notation in (5.31d) is a bit sloppy, but, we hope, clear.)

Remarks

(a) From our definition of a consistent method of order p it follows that the local truncation error $\tau[\mathbf{y}]$ approaches 0 at least as fast as h does [see (5.31b)]. In fact, for Theorem 5.28 to hold it is sufficient to require only that

$$\tau[\mathbf{y}] \to 0 \qquad \text{as } h \to 0$$

for consistency. However, we have no interest in schemes that do not satisfy (5.31b), so we require that consistency imply (at least) $\boldsymbol{\tau}_i[\mathbf{y}] = O(h_i)$. Similarly, we have defined stability in a rather strict way, requiring K to be of moderate size when approximating a well-conditioned BVP, because we are interested only in schemes that satisfy this requirement. [In order for Theorem 5.28 to hold, any constant K would do.]

(b) Establishing consistency and order of a scheme is usually straightforward. In fact, schemes are often derived in such a way that consistency naturally follows. For the midpoint and trapezoidal schemes, second-order accuracy is easily shown by expanding $\mathbf{y}(x_i)$ and $\mathbf{y}(x_{i+1})$ in a Taylor series about the midpoint $x_{i+1/2}$ (see Exercise 5).

(c) In (5.29b) we have emphasized the homogeneous part of the difference equations, because this is what is essential for stability. In other circumstances, particularly in the nonlinear case, it is more natural to write the general one-step scheme as

$$\frac{\mathbf{y}_{i+1} - \mathbf{y}_i}{h_i} = \Phi(\mathbf{y}_i, \mathbf{y}_{i+1}; x_i, h_i) \qquad 1 \le i \le N \tag{5.32a}$$

$$\Phi(\mathbf{y}_i, \mathbf{y}_{i+1}; x_i, h_i) = \Psi(\mathbf{y}_i, \mathbf{y}_{i+1}; x_i, h_i) + \mathbf{q}_i \tag{5.32b}$$

The consistency requirement can then be written as

$$\Phi(\mathbf{y}(x), \mathbf{y}(x); x, 0) = A(x)\mathbf{y}(x) + \mathbf{q}(x) \qquad a \le x \le b \tag{5.33}$$

(d) The definitions above can be easily extended to more general schemes for more general ODE forms. A thing to note, though, is that in general there may be a local truncation error $\boldsymbol{\tau}_0[\mathbf{y}]$ from the BC as well [here it is zero - see (5.30b), (5.13b)]. Then in (5.31b) the max is taken for $0 \le i \le N$; see Exercise 6.

(e) Proving Theorem 5.28 here is straightforward, and can be done as in Section 5.1.1: Writing the difference equations for the global error

$$\mathbf{e}_i := \mathbf{y}(x_i) - \mathbf{y}_i \qquad 1 \le i \le N+1 \tag{5.34a}$$

we have

$$\mathbf{L}_\pi \mathbf{e}_i = \boldsymbol{\tau}_i[\mathbf{y}] \qquad 1 \le i \le N \tag{5.34b}$$

$$B_a \mathbf{e}_1 + B_b \mathbf{e}_{N+1} = \mathbf{0} \tag{5.34c}$$

Now inserting in (5.31c), with $\mathbf{e}_i$ playing the role of $\mathbf{v}_i$, (5.31b) yields the result

$$|\mathbf{e}_i| \le Kch^p \qquad 1 \le i \le N+1. \tag{5.34d}$$

We therefore have obtained not only Theorem 5.28 with the order of convergence, but also that *for h small enough, the maximum error is essentially bounded by the maximum local truncation error times the conditioning constant of the BVP.* □

Let us consider now a general, consistent one-step scheme (5.32), where we assume that

$$|\Psi(\mathbf{u}, \mathbf{v}; x, h)| \le c(|\mathbf{u}| + |\mathbf{v}|) \qquad h \le h_0, a \le x \le b \tag{5.35}$$

c a constant independent of h and x. Then

$$h_i R_i = I + O(h_i), \qquad h_i S_i = -I + O(h_i)$$

so (5.30a) can be written as

$$\mathbf{y}_{i+1} = \Gamma_i \mathbf{y}_i + \mathbf{r}_i \tag{5.36a}$$

$$\Gamma_i = -R_i^{-1} S_i, \qquad \mathbf{r}_i = R_i^{-1} \mathbf{q}_i = O(h_i) \tag{5.36b}$$

The form (5.36a) reminds us of standard multiple shooting (4.39), so it is natural to ask whether Γ_i here also approximates the fundamental solution $Y(x_{i+1}; x_i)$. To recall, the jth column of $Y(x; x_i)$, denoted $\mathbf{v}(x)$, satisfies

$$\mathbf{L}\mathbf{v} = \mathbf{0}, \qquad \mathbf{v}(x_i) = \mathbf{e}^j$$

where $\mathbf{e}^j = (0, \ldots, 0, 1, 0, \ldots, 0)^T$ is the jth unit vector, $1 \le j \le n$. For a one-step approximation of $\mathbf{v}$, consistency implies that

$$S_i \mathbf{e}^j + R_i \mathbf{v}(x_{i+1}) = O(h_i)$$

because any consistent scheme of order $p \ge 1$ is at least first-order accurate. So, by (5.36),

$$\mathbf{v}(x_{i+1}) = \Gamma_i \mathbf{e}^j + O(h_i^2) \qquad 1 \le j \le n$$

i. e.,

$$\Gamma_i = Y(x_{i+1}; x_i) + O(h_i^2) \tag{5.37}$$

The estimate (5.37) shows, in fact, that a one-step difference scheme can be viewed as an instance of the standard multiple shooting method! The mesh points correspond to the shooting points in Section 4.3 and the IVP integrations over each subinterval $[x_i, x_{i+1}]$ are done here in one step. Of course, this particular form of IVP integration is rather special, and the entire approach towards implementation of the finite difference method, in particular mesh selection, is completely different here than in the previous chapter. Note, also, that we have assumed that h^{-1} is much larger than any other quantity in (5.13) or (5.30). Still, we can capitalize on this theoretical equivalence to multiple shooting and obtain stability:

Theorem 5.38 Suppose that the finite difference scheme (5.30) satisfies (5.35) and that the BVP (5.13) is well-posed with conditioning constant κ [cf. (3.34)]. Then the scheme is stable with a stability constant

$$K \le \kappa(b - a + 1) + O(h) \tag{5.38a}$$

Moreover, $\mathbf{A}$ of (5.17) can be equilibrated so that for the resulting matrix $\hat{\mathbf{A}}$,

$$\text{cond } (\hat{\mathbf{A}}) \le \text{const } \kappa N \tag{5.38b}$$

Proof: We equilibrate $\mathbf{A}$ by multiplying its ith block of n rows by R_i^{-1}. Calling the equilibrated matrix $\hat{\mathbf{A}}$ we have, by (5.36), (5.37),

$$\hat{\mathbf{A}} = \mathbf{D}\mathbf{A} = \mathbf{M} + \mathbf{E} \tag{5.39a}$$

where

$$\mathbf{D} = \begin{pmatrix} R_1^{-1} & & & \\ & \ddots & & \\ & & R_N^{-1} & \\ & & & I \end{pmatrix}, \qquad \mathbf{E} = \begin{pmatrix} E_1 & & & \\ & \ddots & & \\ & & E_N & \\ & & & 0 \end{pmatrix} \tag{5.39b}$$

$$\mathbf{M} = \begin{pmatrix} -Y(x_2; x_1) & I & & & \\ & -Y(x_3; x_2) & I & & \\ & & \ddots & & \\ & & & -Y(x_{N+1}; x_N) & I \\ B_a & & & & B_b \end{pmatrix}$$

with $\|R_i^{-1}\| = h_i(1 + O(h_i))$, $\|E_i\| = O(h_i^2)$, $1 \le i \le N$. The inverse of $\mathbf{M}$ is explicitly given [cf. Theorem 4.42] by

$$\mathbf{M}^{-1} = \begin{pmatrix} G(x_1, x_2) & \cdots & G(x_1, x_{N+1}) & Y(x_1)Q^{-1} \\ \vdots & & \vdots & \vdots \\ G(x_{N+1}, x_2) & \cdots & G(x_{N+1}, x_{N+1}) & Y(x_{N+1})Q^{-1} \end{pmatrix} \tag{5.39c}$$

Manipulating (5.39a) a bit, we write

$$\mathbf{A}^{-1} = \hat{\mathbf{A}}^{-1}\mathbf{D} = (\mathbf{M} + \mathbf{E})^{-1}\mathbf{D} = (\mathbf{I} + \mathbf{M}^{-1}\mathbf{E})^{-1}\mathbf{M}^{-1}\mathbf{D} \tag{5.40a}$$

Now, recalling (3.34), we have

$$\begin{aligned} \|\mathbf{M}^{-1}\mathrm{D}\| &= \max_{1 \le i \le N+1} \{ \sum_{j=1}^{N} h_j \| G(x_i, x_{j+1}) \| + O(h) + \|Y(x_i)Q^{-1}\| \} \\ &\le \kappa(b - a + 1) + O(h) \end{aligned} \tag{5.40b}$$

Similarly,

$$\|\mathbf{M}^{-1}\mathbf{E}\| \le c\,\kappa h \tag{5.40c}$$

and therefore, if $c\,\kappa h \ll 1$,

$$\|(\mathbf{I} + \mathbf{M}^{-1}\mathbf{E})^{-1}\| = 1 + O(h) \tag{5.40d}$$

Substituting (5.40b) and (5.40d) in (5.40a) we have for some constant c

$$\|\mathbf{A}^{-1}\| \le \kappa(b - a + 1) + ch =: K \tag{5.40e}$$

This proves stability.

Now, the condition number of the nonequilibrated $\mathbf{A}$ is $\sim K\underline{h}^{-1}$ with $\underline{h} := \min_{1 \le i \le N} h_i$. But this can be improved by considering $\hat{\mathbf{A}}$. Here

$$\|\hat{\mathbf{A}}\| \le \max\{1 + O(h), \|B_a\|, \|B_b\|\}$$

and for $\hat{\mathbf{A}}^{-1}$ we utilize (5.40d) together with (4.42b) to obtain (5.38b). □

Thus we have stability for virtually all sensible one-step schemes. Note the contrast with Chapter 4, where some innocent-looking methods (namely, single shooting and multiple shooting with compactification) are not always stable. Note also that the absolute stability bound (5.31c), (5.38a) yields a rather satisfactory bound in (5.38b) on the *relative* growth of roundoff errors, namely a linear growth as a function of the mesh size N (with no other dependence on the mesh) with a modest constant of proportionality.

Remark Consider again the relationship between a one-step finite difference method and multiple shooting. Recall that the matrix $\mathbf{M}$ of (5.39b) could still have elements with large magnitude if some kinematic eigenvalue λ of $A(x)$ satisfies $\mathrm{Re}(\lambda) \gg 0$, since $\|Y(x_{i+1}; x_i)\|$ reflects the rapid growth of certain fundamental solution modes over the subinterval $[x_i, x_{i+1}]$. Indeed, one would choose shooting points so that $e^{\lambda h_i}$ is of an acceptable size. Yet in Example 5.2 we are able to perform the computations with the particular one-step difference schemes for $\lambda = 5000$ and $h > .01$ (cf. Table 5.2). Many more mesh points would be needed for the standard multiple shooting technique to work properly for that case. We wish to emphasize, though, that this performance by the midpoint and trapezoidal schemes is not covered by Theorem 5.38, because h is not "small enough" and Γ_i of (5.36b) does not closely approximate $-Y(x_{i+1}; x_i)$. This case is treated in Chapter 10. □

5.2.2 Nonlinear problems

Here we consider the nonlinear BVP (5.20). Let us define the differential operator

$$\mathbf{N}\mathbf{y}(x) \equiv \mathbf{y}'(x) - \mathbf{f}(x, \mathbf{y}(x)) \tag{5.41a}$$

and the corresponding one-step difference operator [cf. (5.32a)]

$$\mathbf{N}_\pi \mathbf{y}_i \equiv \frac{\mathbf{y}_{i+1} - \mathbf{y}_i}{h_i} - \Phi(\mathbf{y}_i, \mathbf{y}_{i+1}; x_i, h_i) \qquad 1 \le i \le N \tag{5.41b}$$

We can look at Φ as follows: Integrating (5.20a), we have

$$\mathbf{y}(x_{i+1}) - \mathbf{y}(x_i) = \int_{x_i}^{x_{i+1}} \mathbf{f}(x, \mathbf{y}(x))\, dx$$

so $h_i \Phi(\mathbf{y}(x_i), \mathbf{y}(x_{i+1}); x_i, h_i)$ is an approximation to $\int_{x_i}^{x_{i+1}} \mathbf{f}(x, \mathbf{y}(x))\, dx$. Taking $h_i \to 0$, it is clear that a consistency requirement on Φ is

$$\Phi(\mathbf{y}(x), \mathbf{y}(x); x, 0) = \mathbf{f}(x, \mathbf{y}(x)) \qquad a \le x \le b \tag{5.42}$$

Two examples for Φ are given in (5.21) and in (5.23).

Assumption. We will assume, for the rest of this chapter, the following *smoothness properties:* The functions $\mathbf{f}(x, \mathbf{u})$ and $\mathbf{g}(\mathbf{u}, \mathbf{v})$ appearing in (5.20) have continuous second partial derivatives, while $\Phi(\mathbf{u}, \mathbf{v}; x, h)$ of (5.41b) has continuous first partial derivatives with respect to $\mathbf{u}$ and $\mathbf{v}$. □

As was noted in Chapter 3, with nonlinear problems certain properties hold only in a local sense: The differential problem (5.20) may have a number of solutions, and all we can hope for in general is that a solution $\mathbf{y}(x)$ is isolated (or, locally unique) and that if we begin our iteration sufficiently close to $\mathbf{y}(x)$, it will converge to an approximation of that solution. Recall from Section 3.1 that $\mathbf{y}(x)$ is an isolated solution of the BVP (5.20) if the linearized homogeneous problem at $\mathbf{y}$,

$$\mathbf{z}' = A(x)\mathbf{z} \tag{5.43a}$$

$$B_a\mathbf{z}(a) + B_b\mathbf{z}(b) = \mathbf{0} \tag{5.43b}$$

has only the trivial solution. In (5.43) we have used

$$A(x) := \frac{\partial \mathbf{f}}{\partial \mathbf{y}}(x, \mathbf{y}(x)) \tag{5.44a}$$

$$B_a := \frac{\partial \mathbf{g}(\mathbf{u}, \mathbf{v})}{\partial \mathbf{u}}, \qquad B_b := \frac{\partial \mathbf{g}(\mathbf{u}, \mathbf{v})}{\partial \mathbf{v}}, \qquad \text{at } \mathbf{u} = \mathbf{y}(a), \quad \mathbf{v} = \mathbf{y}(b) \tag{5.44b}$$

Note that in (5.44) the linearization is about the (unknown) exact solution, while in (5.26), (5.27), a similar linearization about $\mathbf{y}^m$ is used for the $(m+1)$st Newton iteration.

Let us denote the linear operator in (5.43a) by

$$\mathbf{L}[\mathbf{y}]\mathbf{z}(x) \equiv \mathbf{z}'(x) - A(x)\mathbf{z}(x) \tag{5.45}$$

while that of (5.27a) would be $\mathbf{L}[\mathbf{y}^m]\mathbf{w}(x)$. Similarly, the difference operator of the left-hand side of (5.25b) is denoted by $\mathbf{L}_\pi[\mathbf{y}^m]\mathbf{w}_i$. We can write, as in (5.30a),

$$\mathbf{L}_\pi[\mathbf{y}^m]\mathbf{w}_i \equiv S_i\mathbf{w}_i + R_i\mathbf{w}_{i+1} \qquad 1 \le i \le N \tag{5.46a}$$

where for a general nonlinear one-step scheme (5.41b)

$$S_i = -h_i^{-1}I - \frac{\partial \Phi}{\partial \mathbf{y}_i}(\mathbf{y}_i^m, \mathbf{y}_{i+1}^m; x_i, h_i), \quad R_i = h_i^{-1}I - \frac{\partial \Phi}{\partial \mathbf{y}_{i+1}}(\mathbf{y}_i^m, \mathbf{y}_{i+1}^m; x_i, h_i) \tag{5.46b}$$

When using a difference operator $L_\pi[\mathbf{u}]$ linearized about some $\mathbf{u}$ with similarly linearized BC [see (5.27b), (5.43b)], we obtain a linear system as in (5.17). We denote the corresponding matrix A by $A[\mathbf{u}]$. This matrix contains the linearizations of both $\mathbf{f}(x, \mathbf{y})$ and $\mathbf{g}(\mathbf{y}(a), \mathbf{y}(b))$.

Definition 5.47 The *local truncation error* $\tau_i[\mathbf{y}]$ is defined as

$$\tau_i[\mathbf{y}] := \mathbf{N}_\pi\mathbf{y}(x_i) \qquad 1 \le i \le N \tag{5.47a}$$

The finite difference scheme

$$\mathbf{N}_\pi\mathbf{y}_i = \mathbf{0} \qquad 1 \le i \le N$$

$$\mathbf{g}(\mathbf{y}_1, \mathbf{y}_{N+1}) = \mathbf{0}$$

is said to be *consistent of order p* if for every smooth solution of (5.20a) there exist positive constants c and h_0 such that for all meshes π with $h \le h_0$

$$\tau[\mathbf{y}] := \max_{1 \le i \le N} |\boldsymbol{\tau}_i[\mathbf{y}]| \le ch^p \tag{5.47b}$$

□

Thus, consistency and order of accuracy are defined precisely as in the linear case. Stability, though, is defined only in the vicinity of a solution. Consider a "discrete tube" around $\mathbf{y}(x)$,

$$S_\rho^\pi(\mathbf{y}) := \{\mathbf{u}_\pi;\ |\mathbf{u}_i - \mathbf{y}(x_i)| \le \rho,\ 1 \le i \le N+1\} \tag{5.48}$$

for some radius $\rho > 0$.

Definition 5.49 The difference scheme is said to be *stable* around $\mathbf{y}(x)$ if there are positive constants K^*, ρ, and h_0 such that for all meshes π with $h \le h_0$ and all mesh functions $\mathbf{u}_\pi, \mathbf{v}_\pi$ in $S_\rho^\pi(\mathbf{y})$,

$$|\mathbf{u}_i - \mathbf{v}_i| \le K^* \max\{\ |\mathbf{g}(\mathbf{u}_1, \mathbf{u}_{N+1}) - \mathbf{g}(\mathbf{v}_1, \mathbf{v}_{N+1})|, \max_{1 \le j \le N} |\mathbf{N}_\pi \mathbf{u}_j - \mathbf{N}_\pi \mathbf{v}_j|\ \} \qquad 1 \le i \le N+1 \tag{5.49}$$

[cf. (5.31c)] and K^* is of moderate size if the constant κ of (3.34) associated with the variational BVP (5.43) is. □

The definition of *convergence* is the same as in the linear case [see (5.31d)]. Theorem 5.28 once again follows directly, provided $\mathbf{y}_\pi$ exists. To see this, consider the (nonlinear) difference equations which the error mesh function satisfies, and observe that the local truncation error is the corresponding residual vector [cf. (5.34)]. Substituting in the stability bound (5.49), we obtain the convergence bound

$$|\mathbf{y}_i - \mathbf{y}(x_i)| \le K^* \tau[\mathbf{y}] \le K^* ch^p \qquad 1 \le i \le N+1 \tag{5.50}$$

To guarantee existence and stability we make the following assumption on the linearized scheme in the vicinity of the exact solution:

Assumption 5.51 For some $\rho > 0$ and all $h \le h_0$, the linearized finite difference scheme

$$\mathbf{L}_\pi[\mathbf{y}]\mathbf{z}_i = \mathbf{0} \qquad 1 \le i \le N$$

$$B_a \mathbf{z}_1 + B_b \mathbf{z}_{N+1} = \mathbf{0}$$

with B_a and B_b defined in (5.44b), is consistent and stable, with $\left\| \frac{\partial \Phi}{\partial \mathbf{y}_i}(\mathbf{u}_i, \mathbf{u}_{i+1}; x_i, h) \right\|$ and $\left\| \frac{\partial \Phi}{\partial \mathbf{y}_{i+1}}(\mathbf{u}_i, \mathbf{u}_{i+1}; x_i, h) \right\|$ bounded for all $\mathbf{u}_\pi$ in $S_\rho^\pi(\mathbf{y})$. Furthermore, there is a Lipschitz constant K_L such that for all such $\mathbf{u}_\pi$,

$$\|\mathbf{A}[\mathbf{y}] - \mathbf{A}[\mathbf{u}_\pi]\| \le K_L \max_{1 \le i \le N+1} |\mathbf{y}(x_i) - \mathbf{u}_i| \tag{5.51}$$

□

This assumption is quite reasonable for one-step schemes if, for instance, $\mathbf{f}(x, \mathbf{u})$ and $\mathbf{g}(\mathbf{u}, \mathbf{v})$ have continuous second derivatives, as we have assumed. We use it next to show stability.

Theorem 5.52. Under assumption 5.51, the nonlinear scheme (5.41b), (5.22) is stable. Moreover, $\mathbf{A}[\mathbf{u}_\pi]$ has a uniformly bounded inverse for all $\mathbf{u}_\pi \in S_\rho^\pi(\mathbf{y})$ (i.e., the linearized schemes in a tube around $\mathbf{y}(x)$ are stable).

Proof: First we show that $\mathbf{A}[\mathbf{u}_\pi]$ has a uniformly bounded inverse. Assumption 5.51 and (5.40) yield

$$\| \mathbf{A}^{-1}[\mathbf{y}] \| \le K, \qquad K = \kappa + O(h)$$

We now use (5.51) for a contraction argument. For arbitrary $\mathbf{r}$, writing $\mathbf{A}[\mathbf{u}_\pi]\mathbf{z} = \mathbf{r}$ as

$$\{\mathbf{A}[\mathbf{y}] + \mathbf{A}[u_\pi] - \mathbf{A}[\mathbf{y}]\}\mathbf{z} = \mathbf{r}$$

we have

$$\mathbf{A}[\mathbf{y}]\mathbf{z} = \mathbf{r} - \{\mathbf{A}[\mathbf{u}_\pi] - \mathbf{A}[\mathbf{y}]\}\mathbf{z}$$

The bound on $\mathbf{A}^{-1}[\mathbf{y}]$ means that we can write

$$|\mathbf{z}| \le K\{|\mathbf{r}| + \|\mathbf{A}[\mathbf{u}_\pi] - \mathbf{A}[\mathbf{y}]\| \, |\mathbf{z}|\}$$

and using (5.51), (5.48),

$$|\mathbf{z}| \le K\{|\mathbf{r}| + K_L \rho |\mathbf{z}|\}$$

This proves the stability of the linearized schemes for ρ so small that $KK_L\rho < 1$, because then

$$|\mathbf{z}| \le \frac{K}{1 - KK_L\rho} |\mathbf{r}|$$

which means

$$\|\mathbf{A}^{-1}[\mathbf{u}_\pi]\| \le \frac{K}{1 - KK_L\rho} =: K^* \qquad \mathbf{u}_\pi \in S_\rho^\pi(\mathbf{y}) \tag{5.53a}$$

Now write the nonlinear difference scheme (5.41b), (5.22) as a system of nonlinear algebraic equations

$$\mathbf{F}(\mathbf{y}_\pi) := \begin{bmatrix} \mathbf{N}_\pi \mathbf{y}_1 \\ \vdots \\ \mathbf{N}_\pi \mathbf{y}_N \\ \mathbf{g}(\mathbf{y}_1, \mathbf{y}_{N+1}) \end{bmatrix} = \mathbf{0} \tag{5.53b}$$

Using our smoothness assumption on Φ and $\mathbf{g}$, the Mean Value Theorem (for vector-valued functions) can be invoked to yield

$$\mathbf{F}(\mathbf{u}_\pi) - \mathbf{F}(\mathbf{v}_\pi) = \mathbf{A}[\mathbf{u}_\pi, \mathbf{v}_\pi](\mathbf{u}_\pi - \mathbf{v}_\pi) \tag{5.53c}$$

where

$$\mathbf{A}[\mathbf{u}_\pi, \mathbf{v}_\pi] := \int_0^1 \mathbf{A}[t\mathbf{u}_\pi + (1-t)\mathbf{v}_\pi] \, dt$$

But, for $\mathbf{u}_\pi, \mathbf{v}_\pi \in S_\rho^\pi(\mathbf{y})$, (5.51) yields

$$\|\mathbf{A}[\mathbf{u}_\pi, \mathbf{v}_\pi] - \mathbf{A}[\mathbf{y}]\| \le K_L \rho$$

so the bound

$$\|\mathbf{A}^{-1}[\mathbf{u}_\pi, \mathbf{v}_\pi]\| \le K^*$$

is obtained, precisely as in (5.53a). The stability bound (5.49) now follows from (5.53c). □

Since the hypotheses of Theorem 5.52 are quite realistic, we have been able to show stability in a neighborhood of the exact solution $\mathbf{y}(x)$ for virtually any sensible one-step difference scheme. However, this is worthwhile only if we are also able to show that there exists a solution $\mathbf{y}_\pi$ for the numerical scheme in a small neighborhood of the isolated solution. For this, let us recall Newton's method, already introduced in Section 5.1.3, which we rewrite for (5.53b) as

$$\mathbf{A}[\mathbf{y}_\pi^m]\mathbf{w}_\pi = -\mathbf{F}(\mathbf{y}_\pi^m) \tag{5.54a}$$

$$m = 0,1,...$$

$$\mathbf{y}_\pi^{m+1} = \mathbf{y}_\pi^m + \mathbf{w}_\pi \tag{5.54b}$$

[cf. (5.24)]. The Newton-Kantorovich Theorem 2.64 can be used to show *existence* of $\mathbf{y}_\pi$, if we can show that, with $\mathbf{y}_\pi^0$ an initial guess, there are positive constants ρ_0, α, β, and γ such that

$$\alpha\beta\gamma \le \frac{1}{2}, \qquad \rho_0\alpha\gamma \le 1 - \sqrt{1-2\alpha\beta\gamma} \tag{5.55a}$$

$$\|\mathbf{A}^{-1}[\mathbf{y}_\pi^0]\| \le \alpha \tag{5.55b}$$

$$|\mathbf{A}^{-1}[\mathbf{y}_\pi^0]\mathbf{F}(\mathbf{y}_\pi^0)| \le \beta \tag{5.55c}$$

$$\|\mathbf{A}[\mathbf{u}_\pi] - \mathbf{A}[\mathbf{v}_\pi]\| \le \gamma|\mathbf{u}_\pi - \mathbf{v}_\pi| \qquad \text{for all } \mathbf{u}_\pi, \mathbf{v}_\pi \in S_{\rho_0}^\pi(\mathbf{y}_\pi^0) \tag{5.55d}$$

But this is straightforward, using the stability Theorem 5.52, if we choose $\mathbf{y}_\pi^0$ close enough to the vector of exact solution values at mesh points, $(\mathbf{y}(x_1), \ldots, \mathbf{y}(x_{N+1}))^T$. We have

Theorem 5.56 Let $\mathbf{y}(x)$ be an isolated solution of the BVP (5.20) and assume that (5.51) holds for a consistent scheme (5.53b). Then for some $\rho_0 \le \rho$ and h_0 sufficiently small, (5.53b) has a unique solution $\mathbf{y}_\pi \in S_\rho^\pi(\mathbf{y})$ for all meshes π with $h \le h_0$. This solution can be computed by Newton's method (5.54), which converges quadratically. □

The proof of this theorem is left as a useful exercise involving the concepts introduced above.

Remarks

(a) Theorem 5.56 relies on $\mathbf{y}_\pi^0$ being "sufficiently close" to $\mathbf{y}$. The big practical problem of how to choose $\mathbf{y}_\pi^0$ is ignored here. We will return to this in Chapter 8, but note that if $\mathbf{y}_\pi^0$ is not "close" to $\mathbf{y}$ and to $\mathbf{y}_\pi$, then the promised quadratic convergence may not be realized, and even if there is eventually a convergence of the nonlinear iteration, not until near the termination of the iteration process.

(b) The requirement that $\mathbf{y}_\pi^0$ is close enough to $\mathbf{y}(x)$ makes sense only when $\mathbf{y}_\pi$ is also close to $\mathbf{y}(x)$. This is hidden in the requirement that h_0 is "small enough." In practice, the closeness of $\mathbf{y}_\pi$ to $\mathbf{y}(x)$ depends on the mesh chosen. This is discussed in Chapter 9, but note here that if $\mathbf{y}_\pi$ is not close to $\mathbf{y}(x)$, even when $\mathbf{y}_\pi$ does exist, then we do not have the premises of Theorem 5.56.

(c) If ρ is small enough so that $KK_L\rho \ll 1$, then $K^* \approx K \approx \kappa$ [see (5.53a)]. This means that close to the solution, the condition of the difference method for the nonlinear BVP is essentially that of the linearized BVP (5.43), (5.44). This is reflected in the actual error bound (5.50). So, when h is small enough, under mild assumptions everything is under control. However, once again life is sometimes more difficult with certain challenging practical problems. If $KK_L\rho$ is not small (but $KK_L\rho < 1$) for a given $\mathbf{y}_\pi^0$, which is the best available guess in a particular instance, then the conditioning constant K^* may be quite different from K.

5.3 HIGHER-ORDER ONE-STEP SCHEMES

In the previous section, a detailed convergence and stability analysis was given for methods based on general one-step schemes. However, the only actual schemes we have seen so far are the trapezoidal and the midpoint schemes, introduced in Section 5.1. These are simple schemes, but their order is only 2; i. e., the error is $O(h^2)$. If high accuracy is desired, then higher-order schemes can be much more effective. As previously mentioned, there are two ways to obtain higher-order methods. One is to use convergence acceleration, which effectively gives a higher-order method. This is considered in Section 5.5. The other way, considered here, is to use higher-order schemes directly. We concentrate on one-step schemes, because nonuniform meshes are easily handled with them, and discretization near boundaries requires no special treatment.

Recall that a natural way to construct such methods is to integrate the ODE (5.20a)

$$\mathbf{y}' = \mathbf{f}(x, \mathbf{y})$$

over each subinterval (x_i, x_{i+1}) and replace the integral by a quadrature rule [cf. (5.41a, b)]. Thus

$$\mathbf{y}(x_{i+1}) - \mathbf{y}(x_i) = \int_{x_i}^{x_{i+1}} \mathbf{y}'(x)\,dx = \int_{x_i}^{x_{i+1}} \mathbf{f}(x, \mathbf{y}(x))\,dx$$

yields, when we replace the right-hand integral by a quadrature formula and divide by h_i, a scheme

$$h_i^{-1}(\mathbf{y}_{i+1} - \mathbf{y}_i) = \Phi(\mathbf{y}_i, \mathbf{y}_{i+1}; x_i, h_i) \qquad 1 \le i \le N$$

Note that this is precisely the same starting point as when we design one-step schemes for IVPs. The approximation of the BC is always by (5.22) and this avoids boundary truncation errors. The order of the scheme depends on the precision of the quadrature formula.

One choice is to replace $\mathbf{f}$ by its Hermite interpolant of order $p = 2k$ (cf. Section 2.5) and then integrate exactly. For this we need $\mathbf{f}$ and its derivatives up to order $k-1$ at the subinterval ends, x_i and x_{i+1}. We get

$$\mathbf{f}(x, \mathbf{y}(x)) = \sum_{j=0}^{k-1} h_i^j \{\mathbf{f}^{(j)}(x_i, \mathbf{y}(x_i))\phi_j(\frac{x - x_i}{h_i}) + (-1)^j \mathbf{f}^{(j)}(x_{i+1}, \mathbf{y}(x_{i+1}))\phi_j(\frac{x_{i+1} - x}{h_i})\} + O(h_i^p)$$

where the functions $\phi_j(t)$ are polynomials of order $p = 2k$ determined on $[0, 1]$ by interpolation conditions as in Section 2.5. Thus, the numerical method defined by

$$\frac{\mathbf{y}_{i+1} - \mathbf{y}_i}{h_i} = \sum_{j=0}^{k-1} h_i^j \{\alpha_j \mathbf{f}^{(j)}(x_i, \mathbf{y}_i) + \beta_j \mathbf{f}^{(j)}(x_{i+1}, \mathbf{y}_{i+1})\} \tag{5.57a}$$

with

$$\alpha_j := \int_0^1 \phi_j(t)\, dt, \qquad \beta_j = (-1)^j \int_0^1 \phi_j(1-t)\, dt \tag{5.57b}$$

is of order $2k$.

Example 5.4

For $k = 1$ we get $\phi_0(t) = 1 - t$ and so $\alpha_0 = \beta_0 = \frac{1}{2}$ in (5.57b). The resulting method in (5.57a) is the trapezoidal scheme.

For $k = 2$ we get $\phi_0(t) = (2t+1)(1-t)^2$, $\phi_1(t) = t(1-t)^2$ and so $\alpha_0 = \beta_0 = \frac{1}{2}$, $\alpha_1 = -\beta_1 = \frac{1}{12}$. The resulting method of order 4 is

$$\frac{\mathbf{y}_{i+1} - \mathbf{y}_i}{h_i} = \frac{1}{2}(\mathbf{f}(x_i, \mathbf{y}_i) + \mathbf{f}(x_{i+1}, \mathbf{y}_{i+1})) + \frac{h_i}{12}(\mathbf{f}'(x_i, \mathbf{y}_i) - \mathbf{f}'(x_{i+1}, \mathbf{y}_{i+1})) \tag{5.58}$$

□

The obvious problem with (5.58) as well as with higher-order schemes of the form (5.57a) is how to obtain $\mathbf{f}^{(j)}(x_i, \mathbf{y}_i)$ for $j > 0$. This can be achieved by differentiating $\mathbf{f}$ given in the BVP formulation (5.20a), but note that total derivatives are needed, e.g.,

$$\mathbf{f}'(x_i, \mathbf{y}_i) = [\frac{\partial \mathbf{f}}{\partial x} + \frac{\partial \mathbf{f}}{\partial \mathbf{y}} \cdot \mathbf{f}]_{(x_i, \mathbf{y}_i)}$$

If $\mathbf{f}$ is easy to differentiate then schemes like (5.58) can be very efficient. However, this is hard to automate, and for general purposes this family of schemes is rather limited.

More hope lies with another family of higher-order methods, obtained by using values of $\mathbf{f}(x, \mathbf{y}(x))$ at intermediate points of the interval $[x_i, x_{i+1}]$. This leads to Runge-Kutta schemes, and we will concentrate on these for the remainder of this section.

5.3.1 Implicit Runge-Kutta schemes

As for IVPs (Section 2.7) we define a *k-stage Runge-Kutta scheme* for $\mathbf{y}' = \mathbf{f}(x, \mathbf{y})$ by

$$\mathbf{y}_{i+1} = \mathbf{y}_i + h_i \sum_{j=1}^{k} \beta_j \mathbf{f}_{ij} \qquad 1 \le i \le N \tag{5.59a}$$

where

$$\mathbf{f}_{ij} = \mathbf{f}(x_{ij}, \mathbf{y}_i + h_i \sum_{l=1}^{k} \alpha_{jl} \mathbf{f}_{il}) \qquad 1 \le j \le k \tag{5.59b}$$

The points x_{ij} are given by

$$x_{ij} = x_i + h_i \rho_j \qquad 1 \le j \le k, \qquad 1 \le i \le N \tag{5.59c}$$

with

$$0 \le \rho_1 \le \rho_2 \le \cdots \le \rho_k \le 1 \tag{5.59d}$$

the "canonical points." Thus, the points x_{ij}, which are sometimes called *collocation points*, are N scaled translations of the canonical set of k points $\rho_1, \ldots, \rho_k$ into each subinterval of the mesh π.

We have used the subscript i as a mesh interval index, while j and l are stage counters within a particular subinterval. Thus, $1 \le i \le N$ and $1 \le j, l \le k$. We will continue to use the indices in this way throughout this section.

We will use the standard tableau of Fig. 5.2 below to represent Runge-Kutta schemes.

Let us recall from Section 2.7 that the scheme is called *explicit* if $\rho_1 = 0$ and $\alpha_{jl} = 0$, $j \le l$, and *implicit* otherwise. So, for explicit schemes the matrix (α_{jl}) in Fig. 5.2 is strictly lower triangular. For initial value problems, explicit methods have obvious advantages over implicit ones, and so implicit methods are normally used only when there is no other choice (e. g., for very stiff ODEs; cf. Section 2.7). But for boundary value problems, implicitness is inherent in the problem in the sense that solution values on π are obtained simultaneously, so we might as well give up "explicitness" altogether and try to choose the $k(k+2)$ parameters ρ_j, β_l, and α_{jl}, $1 \le j, l \le k$ so as to get the most out of the method in terms of order of accuracy, efficiency, and stiff stability (which is elaborated upon in Section 10.3).

To interpret (5.59a, b) note that we can write

$$\mathbf{f}_{ij} = \mathbf{f}(x_{ij}, \mathbf{y}_{ij}) \tag{5.60a}$$

where

ρ_1	α_{11}	---	α_{1k}
$\vdots$	$\vdots$		$\vdots$
ρ_k	α_{k1}	---	α_{kk}
	β_1	---	β_k

Figure 5.2 A Runge-Kutta scheme

$$\mathbf{y}_{ij} := \mathbf{y}_i + h_i \sum_{l=1}^{k} \alpha_{jl} \mathbf{f}_{il} \qquad 1 \le j \le k \tag{5.60b}$$

is an approximation of $\mathbf{y}(x_{ij})$. Thus, the sum in (5.59a) is a quadrature rule for $\int_{x_i}^{x_{i+1}} \mathbf{f}$ with quadrature weights $\beta_1, \ldots, \beta_k$, while the sum in (5.60b) is a quadrature rule for $\int_{x_i}^{x_{ij}} \mathbf{f}$ with quadrature weights $\alpha_{j1}, \ldots, \alpha_{jk}$. Let us assume that the *precision* of the latter quadrature rule is the same for all j, $1 \le j \le k$, and denote it by s, $1 \le s \le k$. Thus

$$\int_0^{\rho_j} \phi(t)\, dt = \sum_{l=1}^{k} \alpha_{jl} \phi(\rho_l) \qquad \text{all } \phi \in \mathbf{P}_s, \qquad 1 \le j \le k \tag{5.61a}$$

(recall that by $\mathbf{P}_s$ we denote all polynomials of order s, i. e., degree $< s$). Let us further assume that the precision of the quadrature rule in (5.59a) is p, $p \ge s$, i. e.,

$$\int_0^1 \phi(t)\, dt = \sum_{l=1}^{k} \beta_l \phi(\rho_l) \qquad \text{all } \phi \in \mathbf{P}_p \tag{5.61b}$$

We are interested only in schemes which satisfy $p \ge s \ge 1$. Thus, in particular,

$$\sum_{l=1}^{k} \beta_l = 1 \quad \text{and} \quad \sum_{l=1}^{k} \alpha_{jl} = \rho_j, \qquad 1 \le j \le k \tag{5.61c}$$

Example 5.5

Consider the choice $k = 1$, $\rho_1 = \frac{1}{2}$, $\beta_1 = 1$, $\alpha_{11} = \frac{1}{2}$. We want to find the values of s and p for which (5.61) holds. Now, any polynomial of order s, say, can be written as

$$\phi(t) = c_0 + c_1 t + c_2 t^2 + \cdots + c_{s-1} t^{s-1}$$

So, to verify a property like (5.61a) we need to consider only the monomials $1, t, t^2, \ldots, t^{s-1}$. Checking (5.61a) we see that

$$\int_0^{1/2} 1\, dt = \frac{1}{2} = \alpha_{11}, \qquad \int_0^{1/2} t\, dt = \frac{1}{8} \ne \alpha_{11} \cdot \frac{1}{2}$$

so $s = 1$. Checking (5.61b) we see that

$$\int_0^1 1\, dt = 1 = \beta_1, \qquad \int_0^1 t\, dt = \frac{1}{2} = \beta_1 \cdot \frac{1}{2}, \qquad \int_0^1 t^2\, dt = \frac{1}{3} \ne \beta_1 \cdot \frac{1}{4}$$

so $p = 2$.

Let us now take a closer look at this scheme. From (5.59) we have

$$\mathbf{y}_{i+1} = \mathbf{y}_i + h_i \mathbf{f}_{i1}$$

$$\mathbf{f}_{i1} = \mathbf{f}(x_{i+1/2}, \mathbf{y}_{i1}) = \mathbf{f}(x_{i+1/2}, \mathbf{y}_i + \frac{1}{2} h_i \mathbf{f}_{i1})$$

In this case we can eliminate $\mathbf{f}_{i1}$ quite simply, thus obtaining the previous form of a one-step scheme. From the first equation above,

$$\frac{1}{2} h_i \mathbf{f}_{i1} = \frac{1}{2} (\mathbf{y}_{i+1} - \mathbf{y}_i)$$

and substituting in the second equation, we have

$$\mathbf{f}_{i1} = \mathbf{f}(x_{i+1/2}, \frac{1}{2}(\mathbf{y}_i + \mathbf{y}_{i+1}))$$

so the scheme is

$$\mathbf{y}_{i+1} = \mathbf{y}_i + h_i \mathbf{f}(x_{i+1/2}, \frac{1}{2}(\mathbf{y}_i + \mathbf{y}_{i+1}))$$

It is important to note that the above elimination of $\mathbf{f}_{i1}$ can be done *locally*, without relating to any information outside the subinterval $[x_i, x_{i+1}]$.

The reader will identify the last expression as none other than the midpoint scheme (5.23). Thus, we see that the midpoint scheme is a particular instance of an implicit Runge-Kutta method. Another instance is the trapezoidal scheme (see Exercise 5.8). Note also that the precision $p = 2$ agrees with the known second-order accuracy of the midpoint scheme at mesh points. At the midpoint of each subinterval we have $s = 1$ only, which implies first-order accuracy; i. e.,

$$h_i^{-1}(\mathbf{y}(x_{i+1/2}) - \mathbf{y}(x_i)) = \frac{1}{2}\mathbf{f}(x_{i+1/2}, \mathbf{y}(x_{i+1/2})) + O(h_i)$$

It can be shown that this and stability yield

$$\mathbf{y}_{i1} = \mathbf{y}(x_{i+1/2}) + O(h)$$

But an improved, second-order accuracy is later obtained at the midpoints as well [see (5.79b)]. Note from (5.59a) and (5.60b) that

$$\mathbf{y}_{i1} = \frac{1}{2}(\mathbf{y}_i + \mathbf{y}_{i+1})$$

□

Let us now treat the *linear case*, i. e., consider the BVP (5.13). In the Runge-Kutta formulation (5.59) there are two types of unknowns. One is the vector $\mathbf{y}_\pi^T = (\mathbf{y}_1^T, \ldots, \mathbf{y}_{N+1}^T)$ of mesh values. These variables are *global* in that their determination is in some sense done simultaneously, over the entire interval $[a, b]$. The other variables are *local* to $[x_i, x_{i+1}]$, for each i, $1 \le i \le N$. These are

$$\mathbf{f}_i^T := (\mathbf{f}_{i1}^T, \ldots, \mathbf{f}_{ik}^T) \qquad 1 \le i \le N \tag{5.62}$$

Alternatively, we can use (5.60) to express the scheme in terms of the local unknowns $\mathbf{y}_{i1}, \ldots, \mathbf{y}_{ik}$. We then obtain

$$h_i^{-1}(\mathbf{y}_{i+1} - \mathbf{y}_i) = \sum_{l=1}^{k} \beta_l \{A(x_{il})\mathbf{y}_{il} + \mathbf{q}(x_{il})\} \tag{5.63a}$$

$$h_i^{-1}(\mathbf{y}_{ij} - \mathbf{y}_i) = \sum_{l=1}^{k} \alpha_{jl} \{A(x_{il})\mathbf{y}_{il} + \mathbf{q}(x_{il})\} \qquad 1 \le j \le k \tag{5.63b}$$

For both practical and theoretical reasons, we now proceed to *eliminate the local unknowns*. This is sometimes referred to in the literature of finite element methods as *parameter condensation*. We have done it already for Example 5.5. To see how it can be done more generally, we can write (5.59b) for a linear BVP in vector form as

$$W\mathbf{f}_i = V\ \mathbf{y}_i + \mathbf{q}_i \tag{5.64a}$$

where $W \in \mathbf{R}^{nk \times nk}$, $V \in \mathbf{R}^{nk \times n}$ and $\mathbf{q}_i \in \mathbf{R}^{nk}$ are defined by

$$W = I - h_i \begin{bmatrix} \alpha_{11} A(x_{i1}) & \cdots & \alpha_{1k} A(x_{i1}) \\ \vdots & & \\ \alpha_{k1} A(x_{ik}) & \cdots & \alpha_{kk} A(x_{ik}) \end{bmatrix}, \quad V = \begin{bmatrix} A(x_{i1}) \\ \vdots \\ A(x_{ik}) \end{bmatrix}, \quad \mathbf{q}_i = \begin{bmatrix} \mathbf{q}(x_{i1}) \\ \vdots \\ \mathbf{q}(x_{ik}) \end{bmatrix} \tag{5.64b}$$

The obvious dependence of the matrices W and V on the index i is suppressed for notational simplicity. Clearly, for h_i small enough, W is invertible, with

$$W^{-1} = I + O(h_i) \tag{5.64c}$$

Again using linearity of the ODE, we find that substitution in (5.59b) yields, as in (5.36),

$$\mathbf{y}_{i+1} = \Gamma_i \mathbf{y}_i + \mathbf{r}_i \qquad 1 \le i \le N \tag{5.65a}$$

with

$$\Gamma_i := I + h_i D W^{-1} V, \qquad \mathbf{r}_i := h_i D W^{-1} \mathbf{q}_i \tag{5.65b}$$

and $D \in \mathbf{R}^{n \times nk}$ is defined by

$$D := (\beta_1 I, \ \ldots, \ \beta_k I) \tag{5.65c}$$

Combining (5.65a) with the BC (5.14) yields, once again, a linear system of algebraic equations (5.17) with a multiple shooting type matrix **A**. Indeed, the process here resembles multiple shooting even more than before (cf. Section 5.2), because in multiple shooting, the IVP integrations from one shooting point to the next can also be considered as a local elimination process. Alternately, the process here can be considered as multiple shooting where IVP integrations are done by one step of a Runge-Kutta scheme.

Now, with (5.65) our schemes fall within the theory of Section 5.2.1. Thus, all we need for a convergence bound like (5.34d) is to show consistency of order p. This is where the precision requirements (5.61) come in. However, the general case is complicated. We will therefore restrict this aspect of our discussion to a subclass of the Runge-Kutta schemes, one which fortunately contains many schemes of interest.

5.3.2 A subclass of Runge-Kutta schemes

The class of Runge-Kutta schemes considered here is equivalent to the class of collocation schemes discussed in Section 5.4. These schemes are required to satisfy

Assumption 5.66 The points ρ_j are distinct; i. e.,

$$0 \le \rho_1 < \rho_2 < \cdots < \rho_k \le 1$$

and the precision s of (5.61a) equals k (so $p \ge k$). □

Under this assumption we will prove the claimed equivalence to collocation schemes in Section 5.4. Therefore, Theorem 5.79 will apply here. In particular, the error at mesh points is $O(h^p)$. We will delay the convergence proofs until the next section and concentrate here instead on the construction of such schemes. The distinction between them and corresponding collocation schemes will be intentionally blurred.

First, note that given the points $\rho_1, \ldots, \rho_k$, the quadrature weights β_j and α_{jl} are uniquely determined under assumption 5.66. For if (5.61) is to be satisfied for the k monomials $\phi(t) = 1, t, t^2, \ldots, t^{k-1}$ (cf. Example 5.5) then this gives a $k \times k$ linear system of constraints for each k coefficients $\alpha_{j1}, \ldots, \alpha_{jk}$ $(1 \le j \le k)$ or $\beta_1, \ldots, \beta_k$. The matrix of each such system of equations is a nonsingular Vandermonde matrix [see (2.84)], and therefore precisely one solution exists.

Let us now construct the weights β_j and α_{jl}. In the beginning of Section 5.3 we mention the idea of replacing $\mathbf{f}(x, \mathbf{y}(x))$ by a polynomial interpolant in order to obtain a quadrature rule for $\int_{x_i}^{x_{i+1}} \mathbf{f}$. We have already seen what a Hermite interpolant does. Here consider a Lagrange interpolant. Thus, given distinct points $\rho_1, \ldots, \rho_k$ of (5.59d) we define for each mesh subinterval $[x_i, x_{i+1}]$ the collocation points x_{ij} as in (5.59c). We then write the function $\mathbf{y}'(x)$ on $[x_i, x_{i+1}]$ as a sum of its Lagrange interpolant of order k plus a remainder term [recall Section 2.4, eqs. (2.83), (2.91)],

$$\mathbf{y}'(x) = \sum_{l=1}^{k} \mathbf{y}'(x_{il}) L_l\left(\frac{x - x_i}{h_i}\right) + \psi(x) \qquad x_i \le x \le x_{i+1} \tag{5.67a}$$

where

$$L_l(t) := \frac{(t - \rho_1)\ldots(t - \rho_{l-1})(t - \rho_{l+1})\ldots(t - \rho_k)}{(\rho_l - \rho_1)\ldots(\rho_l - \rho_{l-1})(\rho_l - \rho_{l+1})\ldots(\rho_l - \rho_k)} \qquad 1 \le l \le k \tag{5.67b}$$

and the remainder term $\psi(x)$ is expressed in a divided difference form

$$\psi(x) = \mathbf{y}'[x_{i1}, \ldots, x_{ik}, x] \prod_{l=1}^{k} (x - x_{il}) \tag{5.67c}$$

Then, from

$$\mathbf{y}(x) - \mathbf{y}(x_i) = \int_{x_i}^{x} \mathbf{y}'(\xi)\, d\xi = \int_{x_i}^{x} \mathbf{f}(\xi, \mathbf{y}(\xi))\, d\xi$$

we obtain the implicit Runge-Kutta scheme (5.59), with

$$\beta_j = \int_0^1 L_j(t)\, dt \qquad \alpha_{jl} = \int_0^{\rho_j} L_l(t)\, dt \qquad 1 \le j, \quad l \le k \tag{5.68}$$

With this construction, assumption 5.66 clearly holds. As noted before, there is no other Runge-Kutta scheme satisfying (5.66) for the same $\rho_1, \ldots, \rho_k$.

Example 5.6

It is easy to verify that for $k = 1$, $\rho_1 = \frac{1}{2}$, (5.68) gives (in the notation of Fig. 5.2) the midpoint scheme

$$\begin{array}{c|c} 1/2 & 1/2 \\ \hline & 1 \end{array}$$

For the choice $k = 2$, $\rho_1 = 0$, $\rho_2 = 1$, (5.68) gives the trapezoidal scheme

0	0	0
1	1/2	1/2
	1/2	1/2

The scheme resembling Simpson's rule of integration has $k = 3$, $\rho_1 = 0$, $\rho_2 = 1/2$, $\rho_3 = 1$. We get

$$L_1(t) = \frac{(t - 1/2)(t - 1)}{1/2} = 2t^2 - 3t + 1$$

$$L_2(t) = \frac{(t - 0)(t - 1)}{-1/4} = -4t^2 + 4t$$

$$L_3(t) = \frac{(t - 0)(t - 1/2)}{1/2} = 2t^2 - t$$

and this used in (5.68) gives

0	0	0	0
1/2	5/24	1/3	-1/24
1	1/6	2/3	1/6
	1/6	2/3	1/6

We note that $p = 4$ for Simpson's scheme. (Check!) Also, when $\rho_k = 1$, $\alpha_{kl} = \beta_l$ (Why?).

□

We now proceed with the construction (5.68). To obtain schemes with a higher-order of convergence, we need to use families of points which give high-precision quadrature. (See Section 2.6.1, and in particular Theorems 2.126, 2.127 and the following discussion.)

(a) Gauss schemes: The points $\rho_1, \ldots, \rho_k$ are chosen as the zeroes of a Legendre polynomial, (Section 2.6.1). Note that $\rho_1 > 0$, $\rho_k < 1$, so mesh points are not collocation points. For $k = 1$, we have $\rho_1 = \frac{1}{2}$, so the *midpoint scheme* (5.16) is obtained as the simplest instance of this family of schemes. For a k-stage scheme, the accuracy is $O(h^{2k})$.

(b) Radau schemes: The point $\rho_k = 1$ is fixed. Here $\rho_1 > 0$ and mesh points (except a) are particular collocation points. For $k = 1$ we obtain the *backward Euler* scheme

$$h_i^{-1}(\mathbf{y}_{i+1} - \mathbf{y}_i) = \mathbf{f}(x_{i+1}, \mathbf{y}_{i+1}) \tag{5.69}$$

which has some distinguished properties for stiff IVPs and accuracy $O(h)$ in general. For a k-stage scheme, the accuracy is $O(h^{2k-1})$.

(c) Lobatto schemes: The points $\rho_1 = 0$ and $\rho_k = 1$ are fixed. Here mesh points are particular collocation points, and internal ones are in some sense repeated twice. With careful implementation, a k-stage Lobatto scheme is roughly as efficient as a $(k-1)$-stage Gauss scheme. For $k = 2$ the *trapezoidal scheme* is obtained. For a k-stage scheme, the accuracy is $O(h^{2(k-1)})$.

Example 5.7

Let us return to the BVP of Example 5.1. In Table 5.1 we have seen the performance of the one-stage Gauss and the two-stage Lobatto schemes. Below we list numerical results for a two-stage Gauss scheme, given by

$$\begin{array}{c|cc} \frac{1}{2}-\frac{\sqrt{3}}{6} & \frac{1}{4} & \frac{1}{4}-\frac{\sqrt{3}}{6} \\ \frac{1}{2}+\frac{\sqrt{3}}{6} & \frac{1}{4}+\frac{\sqrt{3}}{6} & \frac{1}{4} \\ \hline & 1/2 & 1/2 \end{array}$$

a three-stage Gauss scheme, given by

$$\begin{array}{c|ccc} \frac{1}{2}-\frac{\sqrt{15}}{10} & \frac{5}{36} & \frac{2}{9}-\frac{\sqrt{15}}{15} & \frac{5}{36}-\frac{\sqrt{15}}{30} \\ \frac{1}{2} & \frac{5}{36}+\frac{\sqrt{15}}{24} & \frac{2}{9} & \frac{5}{36}-\frac{\sqrt{15}}{24} \\ \frac{1}{2}+\frac{\sqrt{15}}{10} & \frac{5}{36}+\frac{\sqrt{15}}{30} & \frac{2}{9}+\frac{\sqrt{15}}{15} & \frac{5}{36} \\ \hline & \frac{5}{18} & \frac{4}{9} & \frac{5}{18} \end{array}$$

and a three-stage Lobatto scheme, given in Example 5.6.

The additional accuracy obtained by using higher-order methods is clearly demonstrated when we compare Tables 5.4 and 5.1. Note that the errors in the two-stage Gauss scheme and the three-stage Lobatto scheme are reduced by a factor of 16 when N is doubled (i. e., h halved), because these schemes are of order 4. The three-stage Gauss scheme is of order 6 and the reduction factor for the error is (roughly) $2^6 = 32$ when h is halved. This reduction occurs as long as the discretization error is much larger than the roundoff error. □

TABLE 5.4: Higher-order collocation schemes for Example 5.1

	2-stage Gauss		3-stage Lobatto		3-stage Gauss	
N	$e(y_1)$	$e(y_2)$	$e(y_1)$	$e(y_2)$	$e(y_1)$	$e(y_2)$
2	.20-3	.71-4	.17-4	.11-3	.14-6	.37-6
5	.64-5	.19-5	.57-6	.29-5	.70-9	.17-8
10	.46-6	.12-6	.37-7	.18-6	.13-10	.27-10
20	.33-7	.77-8	.23-8	.11-7	.27-12	.42-12
40	.23-8	.48-9	.15-9	.72-9	.60-14	.71-14
80	.16-9	.30-10	.91-11	.45-10	.13-14*	.94-15*

* These values are mainly roundoff errors.

5.3.3 On the implementation of Runge-Kutta schemes

For the nonlinear BVP (5.20), since $\mathbf{f}(x, \mathbf{y})$ is nonlinear, the Runge-Kutta scheme (5.59) yields a set of nonlinear algebraic equations, for which we consider Newton's method.

In the linear case, we have proceeded by eliminating the local unknowns $\mathbf{f}_{ij}$, defined by (5.59b). This is possible analytically, even in the nonlinear case, when there is only one point ρ_j satisfying $0 < \rho_j < 1$. We have seen one such instance with the midpoint scheme in Example 5.5. Other instances are Simpson's scheme (Lobatto with $k = 3$) and Radau scheme with $k = 2$; see Exercises 10 and 11. For these schemes, the theory of Section 5.2.2 can be applied directly.

Alas, the general picture is not so rosy. In (5.59b) there are nk nonlinear equations which need to be solved if local parameter elimination is to take place. This multiple-shooting-type approach would then yield a one-step scheme falling within the theory of Section 5.2.2.

When (5.59b) is solved for a stiff IVP, assuming that $\mathbf{y}_i$ is known, some variant of Newton's method is usually employed to obtain an accuracy which depends upon the prescribed overall tolerance. The resulting expense incurred is considerable and in fact is the main reason for the original disenchantment with fully implicit Runge-Kutta schemes for IVPs.

The situation is different for BVPs. Here $\mathbf{y}_i$, $\mathbf{y}_{i+1}$, and $\mathbf{f}_i$ are all unknown, the need to solve (5.59b) occurs within each Newton iteration for $\mathbf{y}_\pi$, and it makes little sense to try to solve (5.59b) with an accuracy higher than that produced by one Newton iteration. In other words, we perform Newton iterations on the entire set of unknowns

$$\mathbf{y}_1, \mathbf{f}_1, \mathbf{y}_2, \mathbf{f}_2, \ldots, \mathbf{y}_N, \mathbf{f}_N, \mathbf{y}_{N+1}$$

and carry out the local parameter elimination of the $\mathbf{f}_i$ only *after* the linearization. This process is also the one which the quasilinearization method yields. A quasilinearization algorithm with collocation is described in the next section.

There are a number of methods for solving nonlinear equations which are derived as cheaper approximate alternatives to Newton's method, or the quasilinearization method. Some such methods are discussed in Chapter 8. One way to make an iteration like (5.24) cheaper, which is particular to the Runge-Kutta schemes discussed here, is simply to take the Jacobian constant along each subinterval $[x_i, x_{i+1})$. Thus, the same Jacobian value, $\frac{\partial \mathbf{f}}{\partial \mathbf{y}}(x_i, \mathbf{y}_i)$ say, is used for all k collocation points in the ith subinterval of the mesh. Note that approximating the Jacobian $\mathbf{F}'(\mathbf{s})$ does not alter the value of the root of $\mathbf{F}(\mathbf{s})$ in (5.24). But the convergence properties of the resulting method may change. Still, if $\mathbf{F}'(\mathbf{s})$ does not vary a lot on one mesh subinterval, then this is a cheaper variant of Newton's method which closely approximates its properties.

Other, specially designed Runge-Kutta methods, which do not necessarily satisfy assumption 5.66, are also dicussed in the literature.

Recall that the Runge-Kutta scheme (5.59) gives a discrete solution with values $\mathbf{y}_i$ at mesh points x_i and values $\mathbf{y}_{ij}$ at collocation points x_{ij} satisfying

$$\mathbf{y}_{ij} = \mathbf{y}_i + h_i \sum_{l=1}^{k} \alpha_{jl} \mathbf{f}(x_{il}, \mathbf{y}_{il})$$

[see (5.60)]. Let $\mathbf{y}_\pi(x)$ be a polynomial of order $k+1$ defined on $[x_i, x_{i+1}]$ by the interpolation conditions

$$\mathbf{y}_\pi(x_i) = \mathbf{y}_i \tag{5.70a}$$

$$\mathbf{y}_\pi'(x_{ij}) = \mathbf{f}(x_{ij}, \mathbf{y}_{ij}) \qquad 1 \le j \le k \tag{5.70b}$$

It is simple to verify that such an interpolation process is well-defined. Indeed, if we write $\mathbf{y}_\pi(x)$ in terms of its first derivative, $\mathbf{y}_\pi(x) = \mathbf{y}_i + \int_{x_i}^{x} \mathbf{y}_\pi'(\xi)\, d\xi$, and replace $\mathbf{y}_\pi'(x)$ with its Lagrange interpolant form as in (5.67) [with no remainder because $\mathbf{y}_\pi'(x) \in \mathbf{P}_k$], then we obtain with (5.68)

$$\mathbf{y}_\pi(x_{ij}) = \mathbf{y}_i + \int_{x_i}^{x_{ij}} \sum_{l=1}^{k} \mathbf{f}(x_{il}, \mathbf{y}_{il}) L_l\left(\frac{\xi - x_i}{h_i}\right) d\xi = \mathbf{y}_i + h_i \sum_{l=1}^{k} \alpha_{jl} \mathbf{f}(x_{il}, \mathbf{y}_{il}) = \mathbf{y}_{ij} \tag{5.71a}$$

$$\mathbf{y}_\pi(x_{i+1}) = \mathbf{y}_i + h_i \sum_{l=1}^{k} \beta_l \mathbf{f}(x_{il}, \mathbf{y}_{il}) = \mathbf{y}_{i+1} \tag{5.71b}$$

If we now extend the polynomial to the subinterval $[x_{i+1}, x_{i+2}]$ by using (5.70) with i replaced by $i+1$, the two polynomial pieces will be continuously matched at x_{i+1}, according to (5.71b) and (5.70a). Extending to all i, $1 \le i \le N$, we obtain a *continuous piecewise polynomial function* of order $k+1$ over $[a, b]$, which we still denote by $\mathbf{y}_\pi(x)$. [Note that $\mathbf{y}_\pi(x)$ is a continuous function, whereas $\mathbf{y}_\pi$ is the vector of mesh values $\mathbf{y}_i$. Confusion between the two is not disastrous, however, due to Theorem 5.73 below.]

Combining (5.70b) with (5.71a) we see that the piecewise polynomial function $\mathbf{y}_\pi(x)$ satisfies the ODE at the collocation points, i. e.,

$$\mathbf{y}_\pi'(x_{ij}) = \mathbf{f}(x_{ij}, \mathbf{y}_\pi(x_{ij})) \qquad 1 \le i \le N, \qquad 1 \le j \le k \tag{5.72a}$$

The BC are satisfied as well,

$$\mathbf{g}(\mathbf{y}_\pi(a), \mathbf{y}_\pi(b)) = \mathbf{0} \tag{5.72b}$$

Definition Given a mesh π and points $0 \le \rho_1 < \cdots < \rho_k \le 1$, a *collocation solution* for (5.20) is a continuous, piecewise polynomial function $\mathbf{y}_\pi(x)$ which reduces to a polynomial of degree at most k (order $k+1$) on each mesh subinterval and satisfies (5.72). □

What we have shown above is

Theorem 5.73 Given a mesh π, the unique Runge-Kutta method (5.59) satisfying assumption 5.66 is equivalent to the collocation method (5.72). Moreover,

$$\mathbf{y}_\pi(x_i) = \mathbf{y}_i, \qquad \mathbf{y}_\pi(x_{ij}) = \mathbf{y}_{ij} \qquad 1 \le i \le N, \qquad 1 \le j \le k$$

□

Note that up to this point we have not shown *existence* of the approximate solution in any of its two equivalent forms, the Runge-Kutta or the collocation. We do this next for linear problems; nonlinear problems are considered in Section 5.4.2. After showing existence of the approximate solution we will be interested in the approximation error. We will use the notation

$$\mathbf{e}_\pi(x) := \mathbf{y}_\pi(x) - \mathbf{y}(x) \tag{5.74a}$$

$$\mathbf{e}_{ij} := \mathbf{e}_\pi(x_{ij}), \qquad \mathbf{e}_i := \mathbf{e}_\pi(x_i) \tag{5.74b}$$

5.4.1 Linear problems

First we prove a basic existence and stability result. Then we show higher-order convergence.

Theorem 5.75 The k-stage collocation solution exists for the linear BVP (5.13) and is obtained in a stable way. Moreover, the error and its derivatives satisfy, for $1 \le i \le N$,

$$\max_{x_i \le x \le x_{i+1}} |\mathbf{e}_\pi(x)| = O(h^k) \tag{5.75a}$$

$$\max_{x_i \le x \le x_{i+1}} |\mathbf{e}_\pi^{(j)}(x)| = O(h^{k+1-j})\theta_i^{j-1} \qquad 1 \le j \le k \tag{5.75b}$$

with

$$\theta_i := h/h_i \tag{5.75c}$$

Proof: We view the Runge-Kutta form of the collocation scheme as a one-step scheme. Thus, local parameters are eliminated as in (5.64), and (5.65) is considered. Using (5.61a) with $s = k$ and (5.67), we obtain

$$\mathbf{y}(x_{ij}) - \mathbf{y}(x_i) = \int_{x_i}^{x_{ij}} \mathbf{y}'(x)\,dx = h_i \sum_{l=1}^{k} \alpha_{jl}\mathbf{y}'(x_{il}) + \int_{x_i}^{x_{ij}} \psi(x)\,dx \qquad 1 \le j \le k \tag{5.76a}$$

Similarly,

$$\mathbf{y}(x_{i+1}) - \mathbf{y}(x_i) = h_i \sum_{l=1}^{k} \beta_l \mathbf{y}'(x_{il}) + \int_{x_i}^{x_{i+1}} \psi(x)\,dx \tag{5.76b}$$

In (5.76a, b) the rightmost terms are the residuals, i. e., the amount by which the exact solution fails to satisfy the difference equations, cf. (5.63). Clearly $|\psi(x)| = O(h_i^k)$. Repeating the elimination process which leads to (5.65a), one obtains (Exercise 9)

$$\mathbf{y}(x_{i+1}) = \Gamma_i \mathbf{y}(x_i) + \mathbf{r}_i + h_i \boldsymbol{\tau}_i \tag{5.77a}$$

$$|\tau_i| = O(h_i^k) \tag{5.77b}$$

Since $k \geq 1$, the one-step method is consistent and thus stable by Theorem 5.38. Therefore, the discrete solution exists and satisfies

$$\max_{1 \leq i \leq N+1} |\mathbf{y}_i - \mathbf{y}(x_i)| = O(h^k)$$

Furthermore, using the ODE for $\mathbf{y}'(x_{il})$ in (5.76a) and subtracting (5.63b) yields the error equations

$$\mathbf{e}_{ij} := \mathbf{e}_i + h_i \sum_{l=1}^{k} \alpha_{jl} A(x_{il}) \mathbf{e}_{il} - \int_{x_i}^{x_{ij}} \psi(x)\, dx \tag{5.78a}$$

so

$$\mathbf{e}_{ij} = (1 + O(h_i))\mathbf{e}_i + O(h_i^{k+1}) \tag{5.78b}$$

and this gives a similar bound at collocation points

$$\max_{\substack{1 \leq i \leq N \\ 1 \leq j \leq k}} |\mathbf{y}_{ij} - \mathbf{y}(x_{ij})| = O(h^k)$$

Since $\mathbf{y}_\pi(x)$ and $\mathbf{y}(x)$ satisfy the same ODE at collocation points, we have

$$\mathbf{e}_\pi'(x_{ij}) = A(x_{ij})\mathbf{e}_{ij} \tag{5.78c}$$

so

$$\mathbf{e}_\pi'(x_{ij}) = O(h^k), \qquad 1 \leq j \leq k$$

Next, we express the error derivative $\mathbf{e}_\pi'(x)$ on $[x_i, x_{i+1}]$ as in (5.67). Since $\mathbf{y}_\pi'(x)$ is in $\mathbf{P}_k$, it is equal to its interpolant, so the residual $\psi(x)$ is that of (5.67c),

$$\mathbf{e}_\pi'(x) = \sum_{l=1}^{k} \mathbf{e}_\pi'(x_{il}) L_l\left(\frac{x - x_i}{h_i}\right) + \psi(x) \tag{5.78d}$$

Finally, taking derivatives in (5.78d) and using the estimates of $\mathbf{e}_\pi'(x_{il})$ [and of $\mathbf{e}_i$ for (5.75a)], we obtain the desired results (5.75a, b). □

Theorem 5.75 provides the basic existence and stability result for the collocation solution. However, the orders of the convergence estimates in (5.75a, b) are not sharp. Higher-order estimates can be achieved when $p > k$, as promised in Section 5.3.2 and already demonstrated numerically in Example 5.7. We state the result in terms of the equivalent Runge-Kutta method, relating back to the theory developed in Section 5.2.

Theorem 5.79 If a Runge-Kutta scheme satisfies assumption 5.66 then, as a one-step scheme (5.65), it is accurate of order p for a BVP satisfying $A(x), q(x) \in C^{(p)}[a,b]$. Hence,

$$|\mathbf{y}_i - \mathbf{y}(x_i)| = O(h^p) \qquad 1 \leq i \leq N \tag{5.79a}$$

Also, at collocation points,

$$|\mathbf{y}_{ij} - \mathbf{y}(x_{ij})| = O(h_i^{k+1}) + O(h^p) \qquad 1 \leq j \leq k, \qquad 1 \leq i \leq N \tag{5.79b}$$

Proof: Let $\mathbf{v}(x)$ satisfy

$$\mathbf{L}\,\mathbf{v}(x) = \mathbf{q}(x) \qquad x_i \le x$$

$$\mathbf{v}(x_i) = \mathbf{y}_i$$

Then, from (5.77a) and (5.65a),

$$\boldsymbol{\tau}_i = h_i^{-1}(\mathbf{v}(x_{i+1}) - \mathbf{y}_{i+1}) \tag{5.80}$$

Let $Y(x)$ be the fundamental solution satisfying

$$\mathbf{L}\,Y = 0, \qquad Y(x_i) = I$$

Then

$$\mathbf{v}(x_{i+1}) - \mathbf{y}_{i+1} = \int_{x_i}^{x_{i+1}} Y(x_{i+1})Y^{-1}(t)(\mathbf{q}(t) - \mathbf{L}\,\mathbf{y}_\pi(t))\,dt$$

The integrand can be written, by (5.72a), as

$$Y(x_{i+1})Y^{-1}(t)(\mathbf{q}(t) - \mathbf{L}\,\mathbf{y}_\pi(t)) = -Y(x_{i+1})Y^{-1}(t)(\mathbf{L}\,\mathbf{e}_\pi(t)) =: \mathbf{w}(t)\prod_{l=1}^{k}(t - x_{il})$$

We claim that the function $\theta_i^{-k+1}\mathbf{w}(t)$ has $p - k$ bounded derivatives and can therefore be written, using for instance a local Taylor expansion, as

$$\theta_i^{-k+1}\mathbf{w}(t) = \boldsymbol{\phi}(t) + O(h_i^{p-k})$$

for some $\boldsymbol{\phi}(t) \in \mathbf{P}_{p-k}$, $x_i \le t \le x_{i+1}$. To see this, note that by assumption $A(x)$, $\mathbf{q}(x) \in C^{(p)}[a, b]$, which implies that $\mathbf{y}(x)$, $Y(x) \in C^{(p+1)}[a, b]$. This leaves us to consider $\mathbf{L}\,\mathbf{e}_\pi(t)$. Since $\mathbf{w}(t)$ contains the kth derivative of $\mathbf{L}\,\mathbf{e}_\pi(t)$, we must show that $\theta_i^{-k+1}\mathbf{L}\,\mathbf{e}_\pi(t)$ has p bounded derivatives. This will follow if we show that the derivatives of $\mathbf{e}_\pi(t)$ are appropriately bounded. The latter follows from the estimate (5.75b) of Theorem 5.75, the smoothness of $\mathbf{y}(t)$, and the fact that for any positive integer j

$$\mathbf{y}_\pi^{(k+j)}(t) \equiv \mathbf{0}$$

Now, (5.61b) implies that

$$\int_{x_i}^{x_{i+1}} \boldsymbol{\phi}(t)\prod_{l=1}^{k}(t - x_{il})\,dt = \mathbf{0}$$

so

$$|\,\mathbf{v}(x_{i+1}) - \mathbf{y}_{i+1}\,| = O(h_i^{p+1})\theta_i^{k-1} = O(h_i^{p+2-k}h^{k-1})$$

and (5.79a) is obtained from (5.80). Finally, (5.79b) follows directly from (5.78b). □

From (5.79a, b) we see that, for h small enough, the error at mesh points is particularly small compared to the error at collocation points when $p > k + 1$ and $h = O(h_i)$. The higher-order convergence at mesh points is called *superconvergence*. The proof of superconvergence given above is perhaps less elegant than possible, because we have tied it to the local truncation error and to the theory of Section 5.2.1. Instead, we can write

$$\mathbf{y}_\pi(x) - \mathbf{y}(x) = \int_a^b G(x,t)\,\mathbf{L}(\mathbf{y}_\pi(t) - \mathbf{y}(t))\,dt$$

or

$$\mathbf{y}_\pi(x) - \mathbf{y}(x) = \sum_{j=1}^{N} \int_{x_j}^{x_{j+1}} G(x,t)(\mathbf{L}\,\mathbf{y}_\pi(t) - \mathbf{q}(t))\,dt \tag{5.81a}$$

where $G(x,t)$ is the Green's function, given in (3.32). By the smoothness assumptions of Theorem 5.79, $G(x,t)$ is smooth as a function of t in $[a,x)$ and in $(x,b]$, but it has a jump discontinuity at $t = x$. Thus, if $x \in (x_i, x_{i+1})$ then for $j \neq i$ we obtain, precisely as in the proof of Theorem 5.79,

$$\int_{x_j}^{x_{j+1}} G(x,t)(\mathbf{L}\,\mathbf{y}_\pi(t) - \mathbf{q}(t))\,dt = O(h_j^{p+2-k}h^{k-1}) \qquad x \notin (x_j, x_{j+1}) \tag{5.81b}$$

while

$$\int_{x_i}^{x_{i+1}} G(x,t)(\mathbf{L}\,\mathbf{y}_\pi(t) - \mathbf{q}(t))\,dt = O(h^{k+1}) \qquad x \in (x_i, x_{i+1}) \tag{5.81c}$$

The beauty of this argument is in the way that superconvergence is obtained: When x is a mesh point, there is simply no i such that $x \in (x_i, x_{i+1})$! Hence the sum in (5.81a) involves only terms estimated by (5.81b), and the superconvergence result (5.79a) follows. The estimate (5.79b) is obtained as before from (5.78b). Note that using (5.78c), a similar $O(h_i^{k+1}) + O(h^p)$ estimate is obtained at collocation points for $\mathbf{e}_\pi'(x_{ij})$. This yields, in fact,

$$|\mathbf{y}_\pi(x) - \mathbf{y}(x)| = O(h_i^{k+1}) + O(h^p) \qquad x_i < x < x_{i+1}, \qquad 1 \le i \le N \tag{5.82}$$

5.4.2 Nonlinear problems

For the solution of nonlinear BVPs by collocation, we now describe quasilinearization in algorithmic form.

Algorithm: Collocation with quasilinearization

Input: **(a)** A procedure which returns values of $\mathbf{f} = \mathbf{f}(x, \mathbf{y})$ and $A = \dfrac{\partial \mathbf{f}}{\partial \mathbf{y}}(x, \mathbf{y})$ for given values of x and $\mathbf{y}$.

(b) A procedure which returns values of $\mathbf{g}(\mathbf{u},\mathbf{v})$, $B_1 = \dfrac{\partial \mathbf{g}}{\partial \mathbf{u}}(\mathbf{u},\mathbf{v})$ and $B_2 = \dfrac{\partial \mathbf{g}}{\partial \mathbf{v}}(\mathbf{u},\mathbf{v})$ for given values of $\mathbf{u}$ and $\mathbf{v}$.

(c) An initial solution profile $\mathbf{y}_\pi(x)$ [or, equivalently, the values $\mathbf{y}_1, \mathbf{f}_1, \mathbf{y}_2, \mathbf{f}_2, \ldots, \mathbf{y}_N, \mathbf{f}_N, \mathbf{y}_{N+1}$, where $\mathbf{f}_{ij} = \mathbf{y}_\pi'(x_{ij})$].

(d) A mesh π and a tolerance *TOL*.

Output: Collocation solution which satisfies the collocation equations (5.72) to the extent that the difference between two consecutive Newton iterates is at most *TOL*.

REPEAT

1. {BC equations}
Set B_1 and B_2 at $\mathbf{u} = \mathbf{y}_1$, $\mathbf{v} = \mathbf{y}_{N+1}$;
$\boldsymbol{\beta} := -\mathbf{g}(\mathbf{y}_1, \mathbf{y}_{N+1})$.

2. FOR $i = 1, \ldots, N$ DO
{For each subinterval, assemble collocation equations}
FOR $j = 1, \ldots, k$ DO
{Get relevant quantities at a collocation point}

$$x_{ij} := x_i + h_i \rho_j$$

$$\mathbf{y}_{ij} := \mathbf{y}_i + h_i \sum_{l=1}^{k} \alpha_{jl} \mathbf{f}_{il}$$

$$A(x_{ij}) := \frac{\partial \mathbf{f}}{\partial \mathbf{y}}(x_{ij}, \mathbf{y}_{ij})$$

$$\mathbf{q}(x_{ij}) = \mathbf{f}(x_{ij}, \mathbf{y}_{ij}) - \mathbf{f}_{ij}$$

END FOR {j}

{Local elimination by (5.64)}

$$U_i := W_i^{-1} V_i; \qquad \mathbf{p}_i := W_i^{-1} \mathbf{q}_i$$

3. Set up and solve the linear system
{with Γ_i and $\mathbf{r}_i$ given by (5.65)}

$$\begin{bmatrix} -\Gamma_1 & I & & & \\ & -\Gamma_2 & I & & \\ & & \ddots & & \\ & & & -\Gamma_N & I \\ B_1 & & & & B_2 \end{bmatrix} \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \vdots \\ \vdots \\ \mathbf{w}_{N+1} \end{bmatrix} = \begin{bmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \vdots \\ \mathbf{r}_N \\ \boldsymbol{\beta} \end{bmatrix}$$

4. {Update approximate solution for each subinterval}
FOR $i = 1, \ldots, N$ DO

$$\mathbf{y}_i := \mathbf{y}_i + \mathbf{w}_i;$$

$$\mathbf{f}_i := \mathbf{f}_i + U_i \mathbf{w}_i + \mathbf{p}_i$$

5. {Finish update}

$$\mathbf{y}_{N+1} := \mathbf{y}_{N+1} + \mathbf{w}_{N+1};$$

UNTIL $\| \mathbf{w}_\pi \| \leq TOL$ (or iteration limit exceeded) □

Example 5.8

For the nonlinear Example 3.2, treated numerically also in Examples 5.3 and 4.13, we obtain the results shown in Table 5.5 (with the error e being the maximum error at mesh points).

TABLE 5.5: Higher-order collocation for a simple nonlinear problem

h	e (2-Gauss)	e (3-Lobatto)	e (3-Gauss)	e (4-Lobatto)
.2	.26-6	.16-5	.10-8	.10-9
.1	.18-7	.11-6	.16-10	.14-11
.05	.11-8	.67-8	.26-12	.21-13
.025	.69-10	.42-9	.45-14	.46-15
.0125	.43-11	.26-10	.87-15*	.79-15*

* These values are mainly roundoff errors.

In all cases listed here, we use the initial guess $\mathbf{y}_\pi(x) \equiv \mathbf{0}$ and need two Newton iterations to converge. Comparing these results to those in Table 5.3 or Table 4.4, the advantage of higher-order collocation schemes is clear. Note, though, that we do more work here per step h than for the midpoint scheme, say (cf. Section 5.3.3).

The fourth-order accuracy of the two-stage Gauss and the three-stage Lobatto schemes, and the sixth-order accuracy of the three-stage Gauss and four-stage Lobatto schemes are demonstrated in this nonlinear case. We also observe a difference, by roughly a constant factor, between the errors for schemes of the same order. This is due to some symmetry that holds for this particular example and is of no general importance. □

Let us now turn to the theory supporting the algorithm of collocation and quasilinearization. As mentioned before, the one-step theory of Section 5.2.2 does not quite cover everything here, because the local elimination of $\mathbf{f}_j$, which makes the remaining process a one-step scheme, is done only after linearization in each iteration.

Nevertheless, the modifications required to the one-step theory are not essential, and we discuss them in an informal manner. To simplify notation, we assume that the BC are linear; i. e., we consider the nonlinear problem

$$\mathbf{y}' = \mathbf{f}(x, \mathbf{y}) \qquad a < x < b \tag{5.83a}$$

$$B_a\, \mathbf{y}(a) + B_b \mathbf{y}(b) = \boldsymbol{\beta} \tag{5.83b}$$

We denote by $\mathbf{y}(x)$ an isolated solution of (5.83), on which we concentrate. Recall the linear operator

$$\mathbf{L}[\mathbf{u}]\mathbf{z} \equiv \mathbf{z}' - A[\mathbf{u}]\mathbf{z} \tag{5.84a}$$

where

$$A[\mathbf{u}](x) \equiv \frac{\partial \mathbf{f}}{\partial \mathbf{y}}(x, \mathbf{u}(x)) \tag{5.84b}$$

[cf. (5.45), (5.44a)]. The corresponding collocation discretization (5.59) is denoted by $\mathbf{L}_\pi[\mathbf{u}]$. Thus, $\mathbf{L}_\pi[\mathbf{u}]$ is the collocation operator for the linear differential operator obtained by linearizing (5.83a) about $\mathbf{u}(x)$.

The algorithm of quasilinearization and collocation may be described as follows: Given an initial solution $\mathbf{y}_\pi^0(x)$, successive solution iterates $\mathbf{y}_\pi^{m+1}(x)$ are generated, where $\mathbf{y}_\pi^{m+1}(x)$ is the collocation solution of the linear BVP

$$\mathbf{L}[\mathbf{y}_\pi^m]\,\mathbf{z}(x) = \mathbf{f}(x, \mathbf{y}_\pi^m(x)) - A[\mathbf{y}_\pi^m](x)\,\mathbf{y}_\pi^m(x) \qquad a < x < b \tag{5.85a}$$

$$B_a\,\mathbf{z}(a) + B_b\,\mathbf{z}(b) = \boldsymbol{\beta} \tag{5.85b}$$

$m = 0, 1, \ldots$.

For the linearized BVP, the collocation method with local parameter elimination is a one-step method. It is not difficult to see that those parts of assumption 5.51 and Theorem 5.52 which relate to linearizations hold here. The stability results obtained at mesh points are then extended for the entire collocation solution everywhere, as in Theorem 5.75. Thus, corresponding to $\mathbf{A}[\mathbf{y}]$ in (5.51)–(5.52), we have the collocation discretization for the linear problem

$$\mathbf{L}[\mathbf{y}]\mathbf{z}(x) = \mathbf{f}(x, \mathbf{y}(x)) - A[\mathbf{y}](x)\,\mathbf{y}(x) \tag{5.86}$$

and (5.85b); i. e., the linearization is done about the exact solution $\mathbf{y}(x)$. Note that $\mathbf{y}(x)$ satisfies (5.86), (5.85b). Of course, $\mathbf{y}(x)$ is not known, but we may still consider this linear collocation process, denote its solution by $\mathbf{y}_\pi^0(x)$, and obtain from Theorems 5.75 and 5.79 that, with θ_i given in (5.75c),

$$\max_{x_i \le x \le x_{i+1}} \left| \frac{d^j}{dx^j}(\mathbf{y}_\pi^0(x) - \mathbf{y}(x)) \right| = O(h^{k+1-j})\theta_i^{j-1} \qquad 1 \le j \le k, \qquad 1 \le i \le N \tag{5.87a}$$

$$|\mathbf{y}_\pi^0(x_i) - \mathbf{y}(x_i)| = O(h^p) \qquad 1 \le i \le N+1 \tag{5.87b}$$

$$|\mathbf{y}_\pi^0(x) - \mathbf{y}(x)| = O(h_i^{k+1}) + O(h^p) \qquad x_i \le x \le x_{i+1}, \qquad 1 \le i \le N \tag{5.87c}$$

Using this unknown-but-existing $\mathbf{y}_\pi^0(x)$ as a starting solution profile for the quasi-linearization (5.85), we can apply the Newton-Kantorovich Theorem 2.64, because the bounds in (2.64) [cf. (5.55)] are satisfied for h small enough, with α and γ constants and $\beta = O(h^k)$ according to (5.87a). This yields the existence of a solution $\mathbf{y}_\pi(x)$ to the *nonlinear* collocation equations for (5.83) and the quadratic convergence of the Newton iterates to it. We also obtain the convergence estimate (5.75).

Finally, we write (5.83a) as

$$\mathbf{z}' - A[\mathbf{y}]\mathbf{z} = \mathbf{f}(x, \mathbf{z}) - A[\mathbf{y}]\mathbf{z}$$

and use a Taylor expansion

$$\mathbf{f}(x, \mathbf{z}) = \mathbf{f}(x, \mathbf{y}) + A[\mathbf{y}](\mathbf{z} - \mathbf{y}) + O(|\mathbf{z}-\mathbf{y}|^2)$$

to obtain that $\mathbf{y}_\pi(x)$ is the collocation approximation to the solution of

$$\mathbf{L}[\mathbf{y}]\mathbf{z}(x) = \mathbf{f}(x, \mathbf{y}(x)) - A[\mathbf{y}](x)\,\mathbf{y}(x) + O(|\mathbf{z}-\mathbf{y}|^2)$$

and (5.85b). Comparing this to (5.86), which has the collocation solution $\mathbf{y}_\pi^0(x)$, and using the stability of $\mathbf{L}_\pi[\mathbf{y}]$, we have

$$\max_{a \le x \le b} |\mathbf{y}_\pi^0(x) - \mathbf{y}_\pi(x)| \le c \max_{a \le x \le b} |\mathbf{y}_\pi(x) - \mathbf{y}(x)|^2 = O(h^{2k})$$

So for each x

$$|\mathbf{y}(x) - \mathbf{y}_\pi(x)| \le |\mathbf{y}(x) - \mathbf{y}_\pi^0(x)| + |\mathbf{y}_\pi^0(x) - \mathbf{y}_\pi(x)|$$

$$= |\mathbf{y}(x) - \mathbf{y}_\pi^0(x)| + O(h^{2k})$$

and using (5.87) we retrieve all the results of Theorems 5.75 and 5.79 for the nonlinear problem as well. We summarize this as follows:

Theorem 5.88 Let $\mathbf{y}(x)$ be an isolated solution of the BVP (5.20) with smoothness assumptions as before. Then for each k-stage collocation scheme considered, there are positive constants ρ and h_0 such that the following hold for all meshes with $h \le h_0$.

(a) There is a unique solution $\mathbf{y}_\pi(x)$ to the collocation equations (5.72) in a tube of radius ρ around $\mathbf{y}(x)$.

(b) This solution $\mathbf{y}_\pi(x)$ can be obtained by Newton's method, which converges quadratically provided the initial iterate $\mathbf{y}_\pi^0(x)$ is sufficiently close to $\mathbf{y}(x)$.

(c) The following error estimates hold:

$$|\mathbf{y}_i - \mathbf{y}(x_i)| = O(h^p) \qquad 1 \le i \le N$$

$$|\mathbf{y}_\pi(x) - \mathbf{y}(x)| = O(h_i^{k+1}) + O(h^p) \qquad x_i \le x \le x_{i+1}, \qquad 1 \le i \le N$$

□

5.5 ACCELERATION TECHNIQUES

Acceleration techniques are a very powerful tool for improving the accuracy of a basic, simple finite difference method, such as that based on the trapezoidal scheme or on the midpoint scheme. Recall that these basic schemes, presented in Section 5.1, were only second-order accurate. One way of increasing the order of accuracy, considered in the previous section, is to use higher-order (but per step more expensive) Runge-Kutta schemes. The alternative approach considered here is to use the basic scheme of choice, but in a more involved way. We describe two such ways, extrapolation and deferred corrections.

To understand the underlying principles, let us recall how convergence is shown in Section 5.2. For a one-step scheme (our basic scheme) we have a local truncation error $\boldsymbol{\tau}_i[\mathbf{y}]$ and a stability bound [see, e. g., (5.31)]. We then observe that the error satisfies the difference equations with the local truncation error as the inhomogeneity [see, e. g., (5.34)]. Therefore, substituting into the stability bound, the error is seen to be of the same order in h as the local truncation error.

But suppose now that we know more about $\boldsymbol{\tau}_i[\mathbf{y}]$ than just its order. If we have a way of *estimating* the local truncation error, then perhaps this knowledge can also be translated to a corresponding estimate of the error! Such an estimate for the error can, in turn, be used for an error control in an adaptive code (as discussed in Chapter 9), or it can be added to the approximate solution to improve its accuracy.

In order to be more specific, we consider throughout this section the trapezoidal scheme (5.15) or (5.21) as the *basic scheme,* i. e., the scheme applied in the *basic method.* Note, though, that other symmetric one-step schemes could be used as well. The local truncation error for the trapezoidal scheme is $O(h_i^2)$. This is all that is needed to show $O(h^2)$ error in the solution. But, in fact, there is much more structure in $\boldsymbol{\tau}_i[\mathbf{y}]$. For the general system of ODEs

$$\mathbf{y}' = \mathbf{f}(x, \mathbf{y})$$

the local truncation error can be expanded in a series

$$\boldsymbol{\tau}_i[\mathbf{y}] = \sum_{j=1}^{r} h_i^{2j}\mathbf{T}_j[\mathbf{y}(x_{i+1/2})] + O(h_i^{2r+2}) \qquad 1 \le i \le N \tag{5.89}$$

if $\mathbf{f}$ possesses continuous partial derivatives up to order $2r + 2$ [i. e., the smoothness of $\mathbf{f}$ determines the positive integer r in (5.89)]. A simple exercise with Taylor expansions shows that for the trapezoidal scheme,

$$\mathbf{T}_j[\mathbf{z}(x)] = \frac{-j}{2^{2j-1}(2j+1)!}\mathbf{f}^{(2j)}(x, \mathbf{z}(x)) \tag{5.90}$$

where

$$\mathbf{f}^{(2j)}(x, \mathbf{z}(x)) \equiv \frac{d^{2j}}{dx^{2j}}\mathbf{f}(x, \mathbf{z}(x))$$

Thus, it is natural to ask whether stability yields a series expansion corresponding to (5.89) for the global error, viz.,

$$\mathbf{y}(x_i) - \mathbf{y}_i = \sum_{j=1}^{r} h_i^{2j}\mathbf{d}_j(x_i) + O(h^{2r+2}), \qquad 1 \le i \le N+1 \tag{5.91}$$

where $\mathbf{d}_j(x)$ are bounded functions on [a, b] which are independent of the mesh. In Section 5.5.1 we show that the answer is affirmative, provided that the family of meshes considered is restricted as follows:

Assumption 5.92 There is a constant $K > 0$ independent of h such that for all meshes considered,

$$h/h_i \le K \qquad \text{all } i \tag{5.92}$$

A mesh satisfying (5.92) will be called *quasiuniform.* □

Note that in (5.91), the $O(h^{2r+2})$ remainder term is guaranteed to be smaller than the other terms in the expansion (as $h \to 0$) only under the quasiuniformity assumption. We thus assume throughout this section that (5.92) holds. Also, for convenience of presentation, we assume that any derivatives used are continuous.

Given that an error expansion (5.91) exists, we may attempt to find suitable approximations to the so-called *principal error functions* $\mathbf{d}_j(x)$ and improve the accuracy of the basic scheme. For instance, if we find an $O(h^2)$ approximation $\mathbf{v}_i$ to $\mathbf{d}_1(x_i)$, $1 \le i \le N+1$, then $\mathbf{y}_i + h_i^2\mathbf{v}_i$ is an $O(h^4)$ approximation to $\mathbf{y}(x_i)$. Similarly, if $\mathbf{v}_i - \mathbf{d}_1(x_i) = O(h^4)$ and $\mathbf{w}_i - \mathbf{d}_2(x_i) = O(h^2)$, $1 \le i \le N+1$, then $\mathbf{y}_i + h_i^2\mathbf{v}_i + h^4\mathbf{w}_i$ is an $O(h^6)$ approximation to $\mathbf{y}(x_i)$ (assuming $r \ge 3$), because from (5.91)

$$\mathbf{y}(x_i) - (\mathbf{y}_i + h_i^2\mathbf{v}_i + h_i^4\mathbf{w}_i) = h_i^2(\mathbf{d}_1(x_i) - \mathbf{v}_i) + h_i^4(\mathbf{d}_2(x_i) - \mathbf{w}_i) + O(h^6)$$

$$= O(h^6) \qquad 1 \le i \le N+1$$

An alternative way of using $\mathbf{v}_i$ and $\mathbf{w}_i$ is to obtain a fourth-order method with an *error estimate,* because

$$\mathbf{y}(x_i) - (\mathbf{y}_i + h_i^2\mathbf{v}_i) = h_i^4\mathbf{w}_i + O(h^6) \qquad 1 \le i \le N+1$$

For this latter purpose it would suffice that $\mathbf{v}_i$ be an $O(h^3)$ approximation to $\mathbf{d}_1(x_i)$ and that $\mathbf{w}_i$ be an $O(h)$ approximation to $\mathbf{d}_2(x_i)$.

In Section 5.5.2 we consider one technique to approximate the principal error functions and thereby accelerate convergence. This is the method of *extrapolation*, also known as *Richardson's extrapolation*. It is well-known also in numerical quadrature and in the solution of IVPs. The basic advantage of this method is its simplicity. Another technique, considered in Section 5.5.3 and in Section 5.5.4, is that of *deferred corrections*. The basic advantage with it is that the same mesh is used for all corrections, while with extrapolation finer and finer meshes need to be used in order to obtain the desired increase in the order of convergence.

5.5.1 Error expansion

Before considering the specific techniques for convergence acceleration, we prove that the error in the numerical solution can be expanded in the form (5.91), given that a similar expansion (5.89) holds for the local truncation error. Since the proof is somewhat long, we present it for linear BVPs only and merely state the theorem for the nonlinear case. The more casual reader is asked to at least read the statements of these two theorems. As usual, we consider the linear case first.

Theorem 5.93 Suppose that the linear BVP

$$\mathbf{y}' = \mathbf{f}(x, \mathbf{y}) \equiv A(x)\mathbf{y} + \mathbf{q}(x) \qquad a < x < b$$

$$B_a \mathbf{y}(a) + B_b \mathbf{y}(b) = \boldsymbol{\beta}$$

has a unique solution $\mathbf{y}(x)$ and that $\mathbf{f} \in C^{(2r+2)}[a, b]$. Then the trapezoidal solution $\mathbf{y}_\pi$ on quasiuniform meshes with h small enough satisfies (5.91). The principal error functions are defined as the solutions of the BVPs

$$\mathbf{L}\mathbf{d}_j(x) \equiv \mathbf{d}_j'(x) - A(x)\mathbf{d}_j(x) = \mathbf{T}_j[\mathbf{y}(x)] - \sum_{l=1}^{j-1} \mathbf{T}_l[\mathbf{d}_{j-l}(x)] \qquad a < x < b \tag{5.93a}$$

$$B_a \mathbf{d}_j(a) + B_b \mathbf{d}_j(b) = \mathbf{0} \tag{5.93b}$$

for $1 \le j \le r$.

Proof: Note first that the functions $\mathbf{d}_j(x)$ are (recursively) well-defined in (5.93), independent of any mesh considerations. Given a trapezoidal approximation $\mathbf{y}_\pi$ on a mesh π, we define $\mathbf{v}_\pi = (\mathbf{v}_1, \ldots, \mathbf{v}_{N+1})^T$ by

$$\mathbf{v}_i := \mathbf{y}(x_i) - \mathbf{y}_i - \sum_{j=1}^{r} h_i^{2j} \mathbf{d}_j(x_i) \qquad 1 \le i \le N+1$$

Our task is then to show that $|\mathbf{v}_\pi| = O(h^{2r+2})$. It suffices to show that

$$\mathbf{L}_\pi \mathbf{v}_i = O(h_i^{2r+2}) \qquad 1 \le i \le N \tag{5.94}$$

[using the operator notation of (5.29b)], because

$$B_a \mathbf{v}_1 + B_b \mathbf{v}_{N+1} = \mathbf{0}$$

and the basic scheme is stable. [Indeed, the local step h_i in (5.94) is replaced by the global step h in the remainder term of (5.91) *because* of the use of the stability bound (5.31c).]

To show (5.94) we write

$$\mathbf{L}_\pi \mathbf{v}_i = \mathbf{L}_\pi(\mathbf{y}(x_i) - \mathbf{y}_i) - \sum_{j=1}^{r} h_i^{2j} \mathbf{L}_\pi \mathbf{d}_j(x_i) = \boldsymbol{\tau}_i[\mathbf{y}] - \sum_{j=1}^{r} h_i^{2j} \mathbf{L}_\pi \mathbf{d}_j(x_i)$$

and using the expansion (5.89) for $\boldsymbol{\tau}_i[\mathbf{y}]$,

$$\mathbf{L}_\pi \mathbf{v}_i = \sum_{j=1}^{r} h_i^{2j} \{\mathbf{T}_j[\mathbf{y}(x_{i+1/2})] - \mathbf{L}_\pi \mathbf{d}_j(x_i)\} + O(h_i^{2r+2}) \tag{5.95}$$

For the functions $\mathbf{T}_j$ given by (5.90), $\mathbf{T}_j \in C^{(2r+2(1-j))}$ because $\mathbf{f} \in C^{(2r+2)}$. From (5.93a, b), a simple induction argument shows that $\mathbf{d}_j \in C^{(2(r-j)+3)}[a, b]$. Therefore, we can use the local error expansion (5.89) with $\mathbf{d}_j(x)$ replacing $\mathbf{y}(x)$ to obtain

$$\mathbf{L}_\pi \mathbf{d}_j(x_i) = \mathbf{L}\,\mathbf{d}_j(x_{i+1/2}) + \sum_{l=1}^{r-j} h_i^{2l} \mathbf{T}_l[\mathbf{d}_j(x_{i+1/2})] + O(h_i^{2(r-j+1)})$$

For $\mathbf{L}\,\mathbf{d}_j(x_{i+1/2})$ we use (5.93a), obtaining

$$\mathbf{L}_\pi \mathbf{d}_j(x_i) = \mathbf{T}_j[\mathbf{y}(x_{i+1/2})] - \sum_{l=1}^{j-1} \mathbf{T}_l[\mathbf{d}_{j-l}(x_{i+1/2})] + \sum_{l=1}^{r-j} h_i^{2l} \mathbf{T}_l[\mathbf{d}_j(x_{i+1/2})] + O(h_i^{2(r-j+1)})$$

and substituting into (5.95), we obtain

$$\mathbf{L}_\pi \mathbf{v}_i = \sum_{j=1}^{r} h_i^{2j} \{ \sum_{l=1}^{j-1} \mathbf{T}_l[\mathbf{d}_{j-l}(x_{i+1/2})] - \sum_{l=1}^{r-j} h_i^{2l} \mathbf{T}_l[\mathbf{d}_j(x_{i+1/2})] + O(h_i^{2(r-j+1)}) \}$$

Comparing equal powers of h_i in the last expression, we see that all but the highest powers vanish. The estimate (5.94) is therefore obtained. □

Consider now the nonlinear case.

Theorem 5.96 Let the nonlinear BVP

$$\mathbf{y}' = \mathbf{f}(x, \mathbf{y}) \qquad a < x < b$$

$$\mathbf{g}(\mathbf{y}(a), \mathbf{y}(b)) = \mathbf{0}$$

have an isolated solution $\mathbf{y}(x)$ and suppose that the assumptions of Theorem 5.93 hold. Then the error expansion (5.91) is valid here as well. The principal error functions are solutions of the linearized BVPs

$$\mathbf{L}[\mathbf{y}]\mathbf{d}_j \equiv \mathbf{d}_j' - A(x)\mathbf{d}_j = \mathbf{s}_j(x) \qquad a < x < b \tag{5.96a}$$

$$B_a \mathbf{d}_j(a) + B_b \mathbf{d}_j(b) = \boldsymbol{\gamma}_j \tag{5.96b}$$

for $1 \le j \le r$, where $A(x)$, B_a, and B_b are defined in (5.44); the inhomogeneous terms $\mathbf{s}_j(x)$ and $\boldsymbol{\gamma}_j$ depend on $\mathbf{T}_l$ $(1 \le l \le j)$, $\mathbf{d}_l$ $(1 \le l < j)$, and their derivatives. □

We omit the proof of this theorem because of its technical nature and because the important aspects have already been covered in Theorem (5.93).

Remarks

(a) In the expansions (5.89) and (5.91), the remainder term $O(h^{2r+2})$ appears with an even power. Correspondingly, we have to assume in Theorems 5.93 and 5.96 that $\mathbf{f} \in C^{(2r+2)}$ [a, b]. This can be relaxed to $\mathbf{f} \in C^{(2r+1)}[a, b]$, but then the remainder term in (5.91) is $O(h^{2r+1})$ in general.

(b) The values $\mathbf{y}_i$ of the approximate solution in (5.91) have been tacitly assumed to satisfy the difference equations *exactly*. In practical computation there are also roundoff errors and, in the case of a nonlinear BVP, a nonlinear convergence error. If we want to make use of (5.91) to obtain, say, an $O(h^{2k})$ approximation of $\mathbf{y}(x_i)$, then we must ensure that sufficiently many nonlinear iterations have been performed in obtaining $\mathbf{y}_i$ and that the machine precision constant is sufficiently small that the difference between $\mathbf{y}_i$ and its calculated value is below the $O(h^{2k})$ level.

(c) Let us add a few words regarding the quasiuniformity assumption 5.92. What it says is that the mesh cannot be much coarser in one part of the interval than in another. This is potentially annoying and contrary to the spirit in which we would like to conduct our adaptive mesh selection in Chapter 9 (namely, based on local accuracy needs alone). The practical question is how K of (5.92) appears in the actual error for a given computation. If $K \sim 1$ is necessary for a given method, then we would have to abandon adaptive mesh selection for that method. Note, though, that to obtain an $O(h^{2k})$ approximation ($k \leq r$) using approximations to the principal error functions $\mathbf{d}_1, \ldots, \mathbf{d}_{k-1}$ in (5.91), we can relax (5.92) somewhat to read

$$h/h_i \leq K/h^{r-k+1/k}$$

(see Exercise 16). Thus, if $\mathbf{f}$ is so smooth that r can be taken significantly larger than k, then the mesh restriction becomes much more tolerable. For instance, if we want a sixth-order approximation and $\mathbf{f}$ has 12 continuous derivatives, then we only need $h_i \geq Kh^2$ for all i.

5.5.2 Extrapolation

The essence of the extrapolation technique is the construction of successively higher-order approximate solutions from linear combinations of approximate solutions obtained by using the basic scheme on different meshes. Its success depends upon the existence of an error expansion like (5.91). The principal error functions $\mathbf{d}_j(x)$ are not known (or else we could use them directly to improve upon the accuracy of the approximate solution), but the knowledge of the powers of h_i appearing in the error expansion is sufficient to eliminate them.

Specifically, given any quasiuniform mesh π, we use the basic scheme to compute solutions on π and on refinements of π. These refinements are formed by subdividing each subinterval of π into some fixed number of subintervals (perhaps we should say sub-subintervals).

The simplest example of this is when each subinterval of π is halved. If we denote by $\{\mathbf{y}_i^{(1)}\}_{i=1}^{N+1}$ the trapezoidal solution on $\pi_i \equiv \pi$ and by $\{\mathbf{y}_i^{(2)}\}_{i=1}^{2N+1}$ the trapezoidal solution on $\pi_2 = \{\hat{x}_j : \hat{x}_{2i-1} := x_i, i = 1, \ldots, N+1, \hat{x}_{2i} := x_{i+1/2}, i = 1, \ldots, N\}$, then (5.91) gives

$$\mathbf{y}(x_i) - \mathbf{y}_i^{(1)} = \sum_{j=1}^{r} h_i^{2j} \mathbf{d}_j(x_i) + O(h^{2r+2})$$

$$\mathbf{y}(x_i) - \mathbf{y}_{2i-1}^{(2)} = \sum_{j=1}^{r} \left(\frac{h_i}{2}\right)^{2j} \mathbf{d}_j(x_i) + O(h^{2r+2})$$

Thus, the mesh function $\mathbf{y}_\pi^{(1,2)} = (\mathbf{y}_1^{(1,2)}, \ldots, \mathbf{y}_{N+1}^{(1,2)})^T$ defined by

$$\mathbf{y}_i^{(1,2)} := \frac{1}{3}(4\mathbf{y}_{2i-1}^{(2)} - \mathbf{y}_i^{(1)}) \tag{5.97}$$

is a fourth-order approximate solution, because $\mathbf{d}_1$ has been eliminated and

$$\mathbf{y}(x_i) - \mathbf{y}_i^{(1,2)} = -\frac{h_i^4}{4}\mathbf{d}_2(x_i) - \frac{5h_i^6}{16}\mathbf{d}_3(x_i) + O(h^8) = O(h^4) \tag{5.98}$$

Similarly, an $O(h^{2k})$ approximate solution is obtained on the mesh π (provided that $\mathbf{f} \in C^{(2k)}[a,b]$) by computing k solutions $\{\mathbf{y}_i^{(1)}\}, \ldots, \{\mathbf{y}_i^{(k)}\}$ using the same basic scheme on k different meshes $\pi_1, \ldots, \pi_k$. The mesh π_j is obtained by subdividing each subinterval $[x_i, x_{i+1}]$ of π into p_j equal subintervals ($1 = p_1 < p_2 < \cdots < p_k$). Two possible choices for p_j are

$$p_j = j \quad 1 \le j \le k \tag{5.99a}$$

and

$$p_j = 2^{j-1} \quad 1 \le j \le k \tag{5.99b}$$

As before, we have for each approximate solution $\mathbf{y}^{(j)}$ and each mesh point x_i

$$\mathbf{y}(x_i) - \mathbf{y}_{(i-1)p_j+1}^{(j)} = \sum_{l=1}^{k-1} \left(\frac{h_i}{p_j}\right)^{2l} \mathbf{d}_l(x_i) + O(h^{2k}) \qquad 1 \le j \le k \tag{5.100a}$$

In (5.100a) we have k linear equations for k unknowns, $\mathbf{d}_1(x_i), \ldots, \mathbf{d}_{k-1}(x_i)$ and $\mathbf{y}(x_i)$. We write this as

$$\begin{bmatrix} -p_1^{-2} & -p_1^{-4} & \cdots & -p_1^{-2(k-1)} & 1 \\ -p_2^{-2} & -p_2^{-4} & \cdots & -p_2^{-2(k-1)} & 1 \\ \vdots & & & & \vdots \\ -p_k^{-2} & -p_k^{-4} & \cdots & -p_k^{-2(k-1)} & 1 \end{bmatrix} \begin{bmatrix} h_i^2 \mathbf{d}_1(x_i) \\ h_i^4 \mathbf{d}_2(x_i) \\ \vdots \\ h_i^{2(k-1)} \mathbf{d}_{k-1}(x_i) \\ \mathbf{y}(x_i) \end{bmatrix} = \begin{bmatrix} \mathbf{y}_{(i-1)p_1+1}^{(1)} \\ \mathbf{y}_{(i-1)p_2+1}^{(2)} \\ \vdots \\ \mathbf{y}_{(i-1)p_k+1}^{(k)} \end{bmatrix} + O(h^{2k}) \tag{5.100b}$$

Note that the matrix in (5.100b) is essentially a nonsingular Vandermonde matrix [cf. (2.84]. It is independent of the mesh π or the particular mesh point x_i and depends only on the p_js and on the powers of h_i in the expansion (5.91) (or (5.89)). Thus we can (analytically) apply Gauss elimination once and for all to transform this matrix into an upper triangular one. Then the last row of (5.100b) gives an $0(h^{2k})$ approximation to $\mathbf{y}(x_i)$ as a linear combination of the known values $\mathbf{y}_{(i-1)p_j+1}^{(j)}$.

An alternative use of (5.100) is to both obtain an $O(h^{2(k-1)})$ approximation on π and an *error estimate*. Writing (5.100) for the rearranged vector of unknowns $(h_i^2\mathbf{d}_1(x_i), \ldots, h_i^{2(k-2)}\mathbf{d}_{k-2}(x_i), \mathbf{y}(x_i), h_i^{2(k-1)}\mathbf{d}_{k-1}(x_i))^T$ and using the Gauss elimination procedure to obtain an upper triangular matrix, we obtain from the k^{th} row an estimate for the unknown $\mathbf{d}_{k-1}(x_i)$ expressed in terms of computed $\mathbf{y}^{(j)}$ values. Then an approximation for $\mathbf{y}(x_i)$ is obtained from the $k-1^{\text{st}}$ row in terms of $\mathbf{y}^{(j)}$ values and the computed estimate for $\mathbf{d}_{k-1}(x_i)$.

Example 5.9

Choosing $p_j = j$, $1 \le j \le 3$, $k = 3$, we get (if $\mathbf{f} \in C^{(6)}[a, b]$)

$$\mathbf{y}(x_i) - \mathbf{y}_i^{(1)} = h_i^2\mathbf{d}_1(x_i) + h_i^4\mathbf{d}_2(x_i) + O(h^6)$$

$$\mathbf{y}(x_i) - \mathbf{y}_{2i-1}^{(2)} = \frac{1}{4}h_i^2\mathbf{d}_1(x_i) + \frac{1}{16}h_i^4\mathbf{d}_2(x_i) + O(h^6)$$

$$\mathbf{y}(x_i) - \mathbf{y}_{3(i-1)+1}^{(3)} = \frac{1}{9}h_i^2\mathbf{d}_1(x_i) + \frac{1}{81}h_i^4\mathbf{d}_2(x_i) + O(h^6)$$

Thus, defining

$$\mathbf{y}_i^{(1,2)} := \frac{1}{3}(4\mathbf{y}_{2i-1}^{(2)} - \mathbf{y}_i^{(1)}), \qquad \mathbf{y}_i^{(1,3)} := \frac{1}{8}(9\mathbf{y}_{3(i-1)+1}^{(3)} - \mathbf{y}_i^{(1)})$$

[this corresponds to zeroing the subdiagonal elements in the first column of the matrix in (5.100b)], we get

$$\mathbf{y}(x_i) - \mathbf{y}_i^{(1,2)} = -\frac{1}{4}h_i^4\mathbf{d}_2(x_i) + O(h^6)$$

$$\mathbf{y}(x_i) - \mathbf{y}_i^{(1,3)} = -\frac{1}{9}h_i^4\mathbf{d}_2(x_i) + O(h^6)$$

Similarly, defining

$$\mathbf{y}_i^{(2,3)} := \frac{1}{5}(9\mathbf{y}_i^{(1,3)} - 4\mathbf{y}_i^{(1,2)})$$

[which corresponds to the completion of the Gauss elimination process for (5.100)], we get an approximation of order 6,

$$\mathbf{y}(x_i) - \mathbf{y}_i^{(2,3)} = O(h^6)$$

Alternatively, subtracting the error expressions for $\mathbf{y}_i^{(1,2)}$ and $\mathbf{y}_i^{(1,3)}$ from each other, we have

$$\mathbf{y}_i^{(1,3)} - \mathbf{y}_i^{(1,2)} = (-\frac{1}{4} + \frac{1}{9})h_i^4\mathbf{d}_2(x_i) + O(h^6)$$

and so the error in the fourth-order approximation $\mathbf{y}_i^{(1,3)}$ is estimated by the computable values

$$\mathbf{y}(x_i) - \mathbf{y}_i^{(1,3)} \approx -\frac{1}{9}(-\frac{1}{4} + \frac{1}{9})^{-1}(\mathbf{y}_i^{(1,3)} - \mathbf{y}_i^{(1,2)}) = \frac{4}{5}(\mathbf{y}_i^{(1,3)} - \mathbf{y}_i^{(1,2)})$$

Below we list some calculations made for the linear BVP of Examples 5.1 and 5.7. The mesh π is chosen to be uniform with $N = 10$. Then π_2 has 20 subintervals and π_3 has 30. We list only errors in $u = y_1$, using the exact solution (which is known for this simple example) to find the maximum errors at mesh points. Under *EST* we list the error estimate, computed as described above.

TABLE 5.6 Extrapolation for Example 5.1

j	N	$e(u^{(j)})$	$e(u^{(1,j)})$	$e(u^{(2,j)})$	$EST(u^{(1,j)})$
1	10	.31−3	−	−	−
2	20	.76−4	.34−6	−	−
3	30	.34−4	.16−6	.74−8	.15−6

□

In Example 5.9 we have used some notation which may be generalized. Thus, if $\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(k)}$ are the basic scheme solutions on the meshes $\pi_1, \ldots, \pi_k$, then we may define $\mathbf{y}^{(l,j)}$ recursively, as shown in Fig. 5.3.

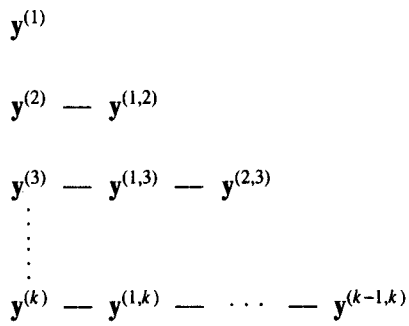

Figure 5.3 Extrapolation

Here, $\mathbf{y}^{(l,j)}$ is an $O(h^{2(l+1)})$ approximation to $\mathbf{y}(x)$, obtained from extrapolating $\mathbf{y}^{(l-1,j)}$ and $\mathbf{y}^{(l-1,j-1)}$ (with $\mathbf{y}^{(0,j)} \equiv \mathbf{y}^{(j)}$) and corresponding to intermediate steps of the Gauss elimination process for (5.100b).

Remarks

(a) We emphasize that the solution of (5.100b) is done as part of *defining* the algorithm, not as part of the solution process for a given BVP. This is apparent in Example 5.9 and Fig. 5.3.

(b) When applying extrapolation for quadrature, or for the solution of IVPs, the choice (5.99b) is usually preferred. This is because (5.99b) is more stable for IVPs and because each mesh π_j is then contained in the next one, π_{j+1}, allowing for savings in the number of evaluations of $\mathbf{f}$ and for a simple adaptive, recursive, local step refinement. For BVPs the situation is different and it is hard to benefit from the advantages of the choice (5.99b). The choice (5.99a) can be preferred

then, because it leads to the fewest number of points in the meshes π_j for a given basic mesh π.

(c) The use of (5.100), whereby an $O(h^{2(k-1)})$ approximate solution with an error estimate is obtained, is particularly attractive. The error estimate can be used to determine whether a given error tolerance has been satisfied, or to refine the mesh in regions where the error is too large (cf. Chapter 9).

(d) From the above description of extrapolation techniques, it should be clear that the use of the trapezoidal scheme for the basic method is not essential. In particular, everything remains the same if we use the midpoint scheme instead. If a higher-order Runge-Kutta scheme is used for the basic method, then the error expansion (5.91) has different powers of h_i. This in turn is reflected in some of the details of the technique, but not in anything essential.

(e) One main criticism of extrapolation is that the computation becomes expensive if many acceleration steps are performed, because the meshes become increasingly fine. For nonlinear problems, this difficulty is somewhat offset because in the majority of applications most iterations are done on the mesh π alone. After $\mathbf{y}_\pi$ is obtained, an $O(h^2)$ approximation of $\mathbf{y}(x)$ is available as an initial solution profile on the next mesh π_2. Generally, after solving for $\{\mathbf{y}_i^{(j)}\}$, $1 \le j \le k$, we may extend it (or $\mathbf{y}_\pi$) by linear interpolation to π_{j+1} and obtain an initial solution profile which is only $O(h^2)$ away from the approximate solution $\{\mathbf{y}_i^{(j+1)}\}$. If this is close enough to give quadratic convergence with Newton's method, then only $\log_2 k + 1$ iterations are needed to find $\{\mathbf{y}_i^{(j+1)}\}$ to $O(h^{2k})$. In fact, a cheaper, modified Newton method may be used here as well (see Chapter 8). Note, however, that the solutions $\{\mathbf{y}_i^{(j)}\}$, $1 \le j \le k$, must all be calculated to accuracy $O(h^{2k})$ [see remark (b) at the end of Section 5.5.1]. Moreover, unlike the method of deferred corrections which is considered next, we cannot reduce the number of Newton iterations by gradually improving the approximation. That is, the use of the $O(h^{2(l+1)})$ solution $\mathbf{y}^{(l,j)}$ as an initial guess for the nonlinear iteration to find $\mathbf{y}^{(j+1)}$ (see Fig. 5.3) has no particular advantage over using $\mathbf{y}^{(j)}$ for that purpose. (Why?)

5.5.3 Deferred corrections

The deferred corrections method is a special case of the *defect correction* method. The basic idea of the latter is to approximate the local truncation error resulting from applying a basic scheme (the "defect" when one is substituting the exact solution into the difference scheme), and then solve for a corrected approximation using this *same* discretization.

Specifically, let $\mathbf{y}_\pi$ be an approximate solution on a mesh π for a nonlinear BVP (5.20) using some basic scheme, so that

$$\mathbf{N}_\pi \mathbf{y}_i = \mathbf{0}, \qquad 1 \le i \le N \tag{5.101a}$$

$$\mathbf{g}(\mathbf{y}_1, \mathbf{y}_{N+1}) = \mathbf{0} \tag{5.101b}$$

Recall that the exact solution satisfies

$$\mathbf{N}_\pi \mathbf{y}(x_i) = \boldsymbol{\tau}_i[\mathbf{y}], \qquad 1 \le i \le N$$

If we know an approximation $\hat{\boldsymbol{\tau}}_i$ to the local truncation error $\boldsymbol{\tau}_i[\mathbf{y}]$ such that

$$\hat{\boldsymbol{\tau}}_i = \boldsymbol{\tau}_i[\mathbf{y}] + O(h^p), \qquad 1 \le i \le N \tag{5.102}$$

then we can compute a corrected solution $\hat{\mathbf{y}}_\pi$ by solving the equations

$$\mathbf{N}_\pi \hat{\mathbf{y}}_i = \hat{\boldsymbol{\tau}}_i, \qquad 1 \le i \le N \tag{5.103a}$$

$$\mathbf{g}(\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_{N+1}) = \mathbf{0} \tag{5.103b}$$

Substitution into the stability bound (5.49) readily yields

$$|\hat{\mathbf{y}}_i - \mathbf{y}(x_i)| \le K^* \max_j |\hat{\boldsymbol{\tau}}_j - \boldsymbol{\tau}_j[\mathbf{y}]| = O(h^p) \tag{5.104}$$

so if p is larger than the order of the basic scheme then we obtain a higher-order method, using the same mesh π and the same difference operator $\mathbf{N}_\pi$.

This defect correction idea is quite general, and (5.89) and (5.91) do not a priori have to hold. But the question is how to obtain approximations $\hat{\boldsymbol{\tau}}_i$ satisfying (5.102) with large p. The method of *(iterated) deferred corrections* gives an adequate answer by using the series expansion (5.89) of the local truncation error and approximating the functions $\mathbf{T}_j$ which appear in that expansion. Here is one reason why we have chosen the trapezoidal scheme as the basic scheme for this section: The functions $\mathbf{T}_j$ given in (5.90) have a particularly simple form, and those of the midpoint scheme are more complicated.

To obtain an $O(h^{2k})$ approximation to $\boldsymbol{\tau}_i[\mathbf{y}]$ we need, by (5.89), an $O(h^{2(k-j)})$ approximation to each $\mathbf{T}_j[\mathbf{y}(x_{i+1/2})]$, $1 \le j \le k-1$. (We assume, of course, that $k \le r$). Suppose we knew the values of $\mathbf{f}(x, \mathbf{y}(x))$ at all mesh points. Then, by passing an interpolating polynomial of order $2k$ through these values at the points $x_{i-k+1}, \ldots, x_{i+k}$ (ignoring for the moment the treatment of near-boundary mesh points) and evaluating its derivatives at $x_{i+1/2}$, we could form the desired approximations to $\mathbf{T}_j$ according to (5.90).

Alas, we do not know $\mathbf{f}(x, \mathbf{y}(x))$, only values of $\mathbf{f}(x_i, \mathbf{y}_i)$, $1 \le i \le N+1$. Still, our goal can be achieved. Initially, $\mathbf{y}_i$ is $O(h^2)$ away from $\mathbf{y}(x_i)$, so we cannot approximate $\mathbf{T}_j[\mathbf{y}(x_{i+1/2})]$ to a higher-order. Fortunately, for $k = 2$ we only need an $O(h^2)$ approximation of $\mathbf{T}_1$ to obtain an $O(h^4)$ method. More generally, if $\mathbf{y}_i$ is $O(h^{2(k-1)})$ away from $\mathbf{y}(x_i)$ for all mesh points, then the interpolating polynomial using these values can yield $O(h^{2(k-j)})$ approximations for $\mathbf{T}_j$, $1 \le j \le k-1$. The approximate solution can therefore be improved, but gradually. Let us summarize this.

Outline: Deferred corrections

Input: A BVP (5.20), a mesh π, and a positive integer k.

Output: Solution values at mesh points, $\mathbf{y}_\pi^{(k)}$, accurate to $O(h^{2k})$.

1. {Obtain $\mathbf{y}_\pi^{(1)}$ using the basic scheme}
   Solve (5.101), denote solution by $\mathbf{y}_\pi^{(1)}$.
2. {Gradually improve accuracy}
   FOR $l = 2, \ldots, k$ DO
   {Find $O(h^{2l})$ approximation for $\boldsymbol{\tau}[\mathbf{y}]$}

2.1 FOR $i = 1, \ldots, N$ DO

Pass a polynomial of order $2l$ through $2l$ neighboring $(x_j, \mathbf{f}(x_j, \mathbf{y}_j^{(l-1)}))$ points; obtain approximations $\mathbf{T}_i^{(j,l)}$ to $\mathbf{T}_j[\mathbf{y}(x_{i+1/2})]$, $1 \le j < l$, by differentiating the polynomial interpolant and evaluating at $x_{i+1/2}$, according to (5.90).

$$\hat{\boldsymbol{\tau}}_i := \sum_{j=1}^{l-1} h_i^{2j} \mathbf{T}_i^{(j,l)}$$

2.2 Solve (5.103), denote solution by $\mathbf{y}_\pi^{(l)}$.

□

Example 5.10

Consider the case where $k = 2$ and the mesh is uniform. Then from (5.90)

$$\mathbf{T}_1[\mathbf{y}(x_{i+1/2})] = -\frac{1}{12}\mathbf{f}''(x_{i+1/2}, \mathbf{y}(x_{i+1/2}))$$

Step 2.1 in the above outline calls for a cubic interpolation of $\mathbf{f}_j = \mathbf{f}(x_j, \mathbf{y}_j)$ where $\mathbf{y}_j = \mathbf{y}_j^{(1)}$ are solution values from the basic trapezoidal method. Thus we interpolate $\mathbf{f}_j$ values at x_{i-1}, x_i, x_{i+1} and x_{i+2} and evaluate the second derivative of the resulting cubic at $x_{i+1/2}$. This gives

$$\hat{\boldsymbol{\tau}}_i = h^2 \mathbf{T}_i^{(1,2)} = \frac{1}{24}(-\mathbf{f}_{i-1} + \mathbf{f}_i + \mathbf{f}_{i+1} - \mathbf{f}_{i+2}) \qquad 2 \le i \le N-1$$

(A more complicated expression would result if the mesh were nonuniform.) For $i = 1$ we need the value of $\mathbf{f}_0$ for this centered formula, so we use instead the interpolant at x_1, x_2, x_3, x_4, evaluate it at $x_{3/2}$, and obtain a noncentered formula

$$\hat{\boldsymbol{\tau}}_1 = \frac{1}{24}(-3\mathbf{f}_1 + 7\mathbf{f}_2 - 5\mathbf{f}_3 + \mathbf{f}_4)$$

Similarly,

$$\hat{\boldsymbol{\tau}}_N = \frac{1}{24}(-3\mathbf{f}_{N+1} + 7\mathbf{f}_N - 5\mathbf{f}_{N-1} + \mathbf{f}_{N-2})$$

In Table 5.7 we list numerical results for this deferred correction method applied to the BVP of Example 5.1, to be compared to those of Examples 5.7 and 5.9. We have computed solutions on three meshes, as in Table 5.6; but unlike Table 5.6, here they are three independent calculations. The error $e(u^{(1)})$ is identical to that in the first column of Table 5.6. (Why?) After calculating $\mathbf{y}_\pi^{(1)}$, $\hat{\boldsymbol{\tau}}_i$ is found and added to the right-hand side of (5.17). The system is solved again, this time for the corrected solution $\mathbf{y}_\pi^{(2)}$. Only forward- and back- substitutions are needed for this, since $\mathbf{A}$ is already decomposed. The resulting maximum error is listed under $e(u^{(2)})$, and the maximum difference between $u^{(1)}$ and $u^{(2)}$ is listed under $EST(u^{(1)})$. Clearly, this is a reasonable estimate for the error in $u^{(1)}$.

TABLE 5.7 Deferred correction for Example 5.1

N	$e(u^{(1)})$	$e(u^{(2)})$	$EST(u^{(1)})$
10	.31−3	.19−5	.31−3
20	.76−4	.16−6	.76−4
30	.34−4	.34−7	.35−4

□

The reason that the above is an "outline," not an algorithm, is that we have avoided specifying a number of important details which we now discuss.

A good deal of the interpolation process in step 2.1 of the outline can be done ahead of time by first computing and storing interpolation (and differentiation) coefficients. (This is called *preprocessing*.) Still, the overhead may be significant. The cost of this step forms an important part of the total cost of the method and involves some nontrivial decisions, as described later on and in Section 5.5.4.

On the other hand, the method also has some beautiful properties. Notice that, as with extrapolation, a similar process can be used to yield, instead of $\mathbf{y}_\pi^{(k)}$, an approximate solution $\mathbf{y}_\pi^{(k-1)}$ with an error estimate (cf. Example 5.10). Moreover, unlike extrapolation, all discretizations are done on the same mesh and using the same basic scheme $\mathbf{N}_\pi$.

For linear BVPs, the linear algebra in steps 1 and 2.2 of the outline involves the same matrix $\mathbf{A}$ of (5.17), (5.18). This matrix is decomposed for the basic scheme (5.101) anyway, so each of the correction solutions (5.103) involves only forward- and back- substitutions. This is economical compared to what is needed to obtain higher-order approximations with extrapolation.

Note that we have not actually used the error expansion (5.91), only the local truncation error expansion (5.89). Still, one useful way to understand the method is to interpret the solution process with corrections as a solution process for the principal error functions $\mathbf{d}_j(x)$ via (5.93). This also helps to explain the recursive nature of the method: In order to define the right-hand side for $\mathbf{d}_j(x)$ in (5.93a), previous $\mathbf{d}_{j-l}(x)$ functions are needed. Thus, a *gradual* buildup of accuracy is at the heart of the process.

For nonlinear BVPs, the discretization $\mathbf{N}_\pi$ in (5.101) and (5.103) involves nonlinear equations that must be solved iteratively. However, we do not use a full Newton iteration. Rather, we relate to the right hand sides in (5.101) and (5.103) as if they were simple inhomogeneities [even though $\hat{\boldsymbol{\tau}}_i$ certainly depends on the unknown solution — see (5.102)]. The quasilinearization process is applied to $\mathbf{N}_\pi$ alone. This gives a fixed-point iteration scheme with Jacobian matrices which have the structure of those in a trapezoidal scheme and can be viewed as approximations for the full Jacobian matrices for the higher-order scheme. The linearized problem (5.25), (5.26) depends, of course, on the previous iterate, but we may still view the solutions of (5.101), (5.103) as one continuous process. Thus, the basic scheme is used as usual for $\mathbf{y}^{(1)}$ and then, in step 2.2 of the outline, additional fixed-point (approximate Newton) iterations are performed. An important question is, how many iterations are needed per value of l in step 2.2?

This is another point where the deferred correction method enjoys an advantage over extrapolation. Here $\mathbf{y}_\pi^{(l-1)}$ may be used as an initial guess for $\mathbf{y}_\pi^{(l)}$, with $|\mathbf{y}_\pi^{(l-1)} - \mathbf{y}_\pi^{(l)}| = O(h^{2(l-1)})$. Also, the error tolerance for the nonlinear convergence for $\mathbf{y}_\pi^{(l)}$ can be just $o(h^{2l})$, not $O(h^{2k})$. Thus we expect one full Newton iteration per correction 2.2 to be sufficient, and we may hope one or two approximate Newton iterations per correction to be sufficient as well! Just what is needed for this expectation to be fulfilled when the approximate Newton's iterations described above are used will be discussed in the next subsection.

While its implications are perhaps not obvious at first, one of the most important considerations for deferred corrections is deciding how to perform the polynomial interpolation (on a generally nonuniform mesh) in step 2.1, and in particular, which

"neighboring points" to choose. Away from boundaries we would certainly use "centered" formulas, i. e., the points $x_{i-k+1}, \ldots, x_{i+k}$. But near boundaries (when $i-k+1 < 1$ or $i+k > N+1$) something special needs to be done. One possibility is to use noncentered formulas as we did in Example 5.10. Another possibility involves obtaining approximate solution values outside the interval $[a, b]$ by stepping outside the boundary with the basic discretization and then using only centered formulas. This requires an *a priori* choice of k. A third possibility is to use extrapolation for values outside $[a, b]$, which (as it turns out) leads to noncentered differences involving more than the minimum number of points. Additional discussion of this question is provided in the next section.

5.5.4 More deferred corrections

All of the alternatives mentioned at the end of Section 5.5.3 for the implementation of deferred correction are somewhat awkward. The main advantage of those which use only centered difference formulas is that it is possible to prove (under appropriate smoothness assumptions) that the method of deferred corrections as described in the outline does indeed give an $O(h^{2k})$ accurate solution $\mathbf{y}_\pi^{(k)}$. The proof utilizes the asymptotic expansion for the error. The main disadvantage in using only centered formulas is the need to find approximations outside the interval. This presents a potentially serious difficulty for BVPs like the one in Example 4.14, which has a singularity just outside $[a, b]$.

Largely for this reason, the major implementations of deferred corrections to date have used noncentered differences near boundaries. In this case the theory using asymptotic expansions is unsatisfactory because, as it turns out, it permits only one correction. We shall now look at deferred corrections from a more detailed and slightly different viewpoint which suggests a more general theoretical framework for proving convergence results.

We begin as before by constructing a high-order finite difference method

$$\mathbf{N}_{\pi,k}\mathbf{y}_i = \mathbf{0} \qquad 1 \le i \le N \tag{5.105a}$$

$$\mathbf{g}(\mathbf{y}_1, \mathbf{y}_{N+1}) = \mathbf{0} \tag{5.105b}$$

with local truncation error

$$\mathbf{N}_{\pi,k}\mathbf{y}(x_i) = \boldsymbol{\tau}_i^k[\mathbf{y}] = O(h^{2k}) \qquad 1 \le i \le N \tag{5.106}$$

A way to achieve a scheme like this is to approximate $\mathbf{f}(x, \mathbf{y}(x))$ by a polynomial of order $2k$ which interpolates $\mathbf{f}$ at $2k$ points centered (to the extent possible) at $x_{i+1/2} := \dfrac{x_i + x_{i+1}}{2}$, and use this in

$$\frac{1}{h_i}\int_{x_i}^{x_{i+1}} (\mathbf{y}'(x) - \mathbf{f}(x, \mathbf{y}(x)))\, dx = 0$$

obtaining (see Exercise 5.23)

$$\mathbf{N}_{\pi,k}\mathbf{y}(x_i) = \frac{1}{h_i}(\mathbf{y}(x_{i+1}) - \mathbf{y}(x_i)) - \sum_{j=1}^{2k} \beta_{ij}\mathbf{f}(x_{p_i+j}, \mathbf{y}(x_{p_i+j})) \tag{5.107}$$

$$= \mathbf{N}_\pi \mathbf{y}(x_i) - \sum_{j=1}^{2k} \alpha_{ij}\mathbf{f}(x_{p_i+j}, \mathbf{y}(x_{p_i+j})) \qquad 1 \le i \le N$$

Here

$$p_i = \begin{cases} 0 & 1 \le i \le k-1 \\ i-k & \text{if} \quad k \le i \le N-k+1 \\ N-2k+1 & N-k+2 \le i \le N \end{cases}$$

the coefficients $\{\alpha_{ij}\}_{j=1}^{2k}$ satisfy the Vandermonde system

$$\sum_{j=1}^{2k} \left(\frac{x_{p_i+j} - x_{i+1/2}}{h_i}\right)^p \alpha_{ij} = \begin{cases} \dfrac{p}{2^p(p+1)} & p = 0, 2, \ldots, 2k-2 \\ 0 & p = 1, 3, \ldots, 2k-1 \end{cases} \tag{5.108}$$

and from (5.89)[2]

$$\mathbf{C}_k(\mathbf{y}(x_i)) := \sum_{j=1}^{2k} \alpha_{ij} \mathbf{f}(x_{p_i+j}, \mathbf{y}(x_{p_i+j})) = \sum_{j=1}^{k-1} h_i^{2j} \mathbf{T}_j[\mathbf{y}(x_{i+1/2})] + O(h_i^{2k}) \tag{5.109}$$

The scheme (5.105) is consistent of order $2k$; however, it is computationally expensive, since the Jacobian matrix has bandwidth $2kn$.

Rather than try to solve (5.105) with the standard approach, we could solve as follows: Given an initial iterate $\mathbf{y}_\pi^{(1)}$, let

$$\mathbf{N}_\pi \mathbf{y}_i^{(l)} = \mathbf{C}_k(\mathbf{y}_i^{(l-1)}) \qquad 1 \le i \le N$$

$$\mathbf{g}(\mathbf{y}_1^{(l)}, \mathbf{y}_{N+1}^{(l)}) = \mathbf{0}$$

for $l = 2, 3, \ldots, k$ or even

$$\mathbf{N}_\pi \mathbf{y}_i^{(l)} = \mathbf{C}_l(\mathbf{y}_i^{(l-1)}) \qquad 1 \le i \le N \tag{5.110a}$$

$$\mathbf{g}(\mathbf{y}_1^{(l)}, \mathbf{y}_{N+1}^{(l)}) = \mathbf{0} \tag{5.110b}$$

for $l = 2, \ldots, k$ (where $\mathbf{C}_l$ is defined in the obvious way). But if $\mathbf{y}_\pi^{(1)}$ is the solution of (5.101) using the basic scheme, then (5.110) is just a particular way to carry out the deferred corrections algorithm outlined before, using (5.103). Indeed, under appropriate assumptions on $\mathbf{y}_\pi^{(l-1)}$ (as we discuss below),

$$\hat{\boldsymbol{\tau}}_i := \mathbf{C}_l(\mathbf{y}_i^{(l-1)}) = \mathbf{C}_l(\mathbf{y}(x_i)) + O(h^{2l}) = \mathbf{N}_\pi \mathbf{y}(x_i) + O(h^{2l})$$

$$= \boldsymbol{\tau}[\mathbf{y}(x_i)] + O(h^{2l}) \tag{5.111}$$

Thus, by constructing higher-order finite difference schemes of the form $\mathbf{N}_{\pi,l} \equiv \mathbf{N}_\pi - \mathbf{C}_l$, we are able to bootstrap our way up to successively higher-order approximations $\mathbf{y}_\pi^{(l)}$ and corresponding truncation error approximations. It *is* a deferred corrections process, since all solution steps use the same basic discretization $\mathbf{N}_\pi$. Note that the method is stable. The stability property is inherited from that of the basic scheme and does not depend on the stability of the higher-order scheme $\mathbf{N}_{\pi,k}$.

[2] Notice that our notation $\mathbf{C}_k(\mathbf{y}_i)$, or for that matter $\mathbf{N}_\pi \mathbf{y}_i$ or $\mathbf{N}_{\pi,k}\mathbf{y}_i$, does not relate the natural dependence upon the entire vector $\mathbf{y}_\pi = (\mathbf{y}_1, \ldots, \mathbf{y}_{N+1})$, but we use it for simplicity.

As previously mentioned, the deferred corrections algorithm needs modification, since the use of noncentered differences is not without consequence. The problem is that the iterates $\mathbf{y}_\pi^{(l-1)}$ in (5.110) are not "smooth" enough (in a sense explained later) for (5.111) to be valid. Instead, some inner iteration with $\mathbf{C}_l$ fixed is necessary. Let I_l be the number of such inner iterations. Then we have

Algorithm: Deferred corrections using noncentered differences

Input: A BVP (5.20), a mesh π, and a positive integer k.

Output: A solution $\mathbf{y}_\pi^{(k)}$ accurate to $O(h^{2k})$.

1. Solve (5.101); denote the solution by $\mathbf{y}_\pi^{(1)}$.

2. FOR $l = 2, \ldots, k$ DO

2.1 $\mathbf{z}_\pi^{l,1} := \mathbf{y}_\pi^{(l-1)}$

2.2 FOR $i = 1, \ldots, N$ DO

Solve a Vandermonde system [like (5.108)] for $\mathbf{C}_l$.

2.3 FOR $j=1, \ldots, I_l$ DO

Solve

$$\mathbf{N}_\pi \mathbf{z}_i^{l,j+1} = \mathbf{C}_l(\mathbf{z}_i^{l,j}) \qquad 1 \le i \le N \tag{5.112}$$

$$\mathbf{g}(\mathbf{z}_1^{l,j+1}, \mathbf{z}_{N+1}^{l,j+1}) = \mathbf{0}$$

2.4 $\mathbf{y}_\pi^{(l)} := \mathbf{z}_\pi^{l,I_l}$

□

The following theorem describes the convergence of this strategy. The proof is fairly technical and is omitted.

Theorem 5.113 Suppose that the hypotheses of Theorem 5.96 are satisfied. Then the above deferred corrections algorithm generates $\{\mathbf{y}_\pi^l\}_{l=1}^k$ such that $\mathbf{y}_\pi^{(l)}$ has accuracy $O(h^{2l})$ for $1 \le l \le k$, if one of the following holds:

(i) $I_2 = 1$ and $I_3 = \cdots = I_k = 2$; or

(ii) $I_2 = I_3 = 1, I_4 = \cdots = I_k = 2$, and [in addition to (5.92)]

$$\sum_{j=1}^{N-1} \left| \frac{h_{j+1}}{h_j} - 1 \right| \le K \tag{5.113a}$$

or

(iii) $I_2 = I_3 = I_4 = 1, I_5 = \cdots = I_k = 2$ and [in addition to (5.92)]

$$\max_{1 \le j \le N-1} \left| \frac{h_{j+1}}{h_j} - 1 \right| \le Kh \tag{5.113b}$$

□

Remarks

(a) We see from the theorem that two inner iterations are always sufficient to achieve the desired orders of accuracy. Only one inner iteration is sufficient for the first few iterations if the additional mesh restriction (5.113a) or (5.113b) is imposed.

Of course, the constant K here is assumed to be independent of the mesh used. Note that (5.113a) is not too severe and holds, for instance (with K not large), for many piecewise uniform meshes. On the other hand, (5.113b) is a rather severe restriction, requiring the mesh to be almost uniform.

(b) In describing the method, we have been imprecise in assuming that the nonlinear equations in step 2.3 are solved exactly. In practice one would generally discretize (5.112) in the standard way to obtain, say, $\mathbf{z}_\pi^{l,j+1}$ from $\mathbf{z}_\pi^{l,j}$ (cf. Section 5.1.3). However, if $\mathbf{z}_\pi^{l,j}$ is sufficiently close to $\mathbf{y}_\pi^{(l)}$ [a reasonable assumption, since $\mathbf{z}^{l,1}$ is an $O(h^{2l-2})$ approximation to $\mathbf{y}_\pi^{(l)}$], then the rapid convergence of Newton's method assures us that one iteration in (5.112) gives essentially $\mathbf{z}_\pi^{l,j+1}$. Thus it is not surprising that, as it turns out, the theorem can be extended to assume that the nonlinear equations are solved sufficiently accurately (but not exactly), without modifying the conclusions.

(c) It is easy to obtain an *a posteriori* error estimate $\mathbf{e}$ for $\mathbf{y}_\pi^{(k)}$: Compute the coefficients of $\mathbf{C}_k(\mathbf{y}_i^{(k)})$, solve the linearized version of

$$\mathbf{N}_\pi \mathbf{z}_i = \mathbf{C}_k(\mathbf{y}_i^{(k)}) \qquad 1 \le i \le N$$

$$\mathbf{g}(\mathbf{z}_1, \mathbf{z}_{N+1}) = \mathbf{0}$$

once, and let $\mathbf{e} := \mathbf{y}_\pi^{(k)} - \mathbf{z}$. Again, the linearized system inherits the simple matrix structure of the basic scheme.

(d) It can be shown that, if a deferred corrections method with noncentered differences is implemented by using (5.110) instead of the iteration in steps 2.3 and 2.4 of the algorithm, then for the three cases in Theorem 5.113, $\mathbf{y}_\pi^{(l)}$ has accuracy $O(h^{p_l})$, where the sequence $\{p_l\}_{l=1}^k$ is

(i) $\{2, 3, 4, 5, \ldots, k+1\}$, or

(ii) $\{2, 4, 5, 6, \ldots, k+2\}$, or

(iii) $\{2, 4, 6, 7, 8, \ldots, k+3\}$, respectively.

□

We demonstrate the convergence properties given in Theorem 5.113 and in this last remark with a numerical example.

Example 5.11

The simple nonlinear BVP

$$y_1' = y_2$$

$$0 < x < 1$$

$$y_2' = -y_2 - y_1^2 + e^{2x}$$

$$y_1(0) = 1, \qquad y_1(1) = e^{-1}$$

has the solution $y_1(x) = -y_2(x) = e^{-x}$. We solve for it on a uniform mesh π using the deferred corrections algorithm, first with all $I_l = 1$ and then with all $I_l = 2$. The resulting maximum errors in π, corresponding to case (iii) for remark (d) and for Theorem 5.113, respectively, are given in Table 5.8 below, along with the measured rates of convergence.

TABLE 5.8 Deferred corrections for Example 5.11

	One iteration						Two iterations					
$k\backslash N$	12		24		48		12		24		48	
0	.39-3		.96-4	2.00	.24-4	2.00	x		x	x	x	x
1	.82-6	3.64	.56-7	3.87	.36-8	3.95	x	x	x	x	x	x
2	.25-8	5.92	.42-10	5.92	.69-12	5.92	.22-8	5.89	.37-10	5.88	.61-12	5.93
3			.45-12	6.91	.37-14	6.92			.45-13	7.64	.21-15	7.74
4			.32-14	8.57	.11-16	8.22			.49-16	9.73	.58-19	9.73
5			.91-17	10.35	.21-19	8.74			.60-19	11.87	.19-22	11.62
6					.45-22	10.89					.61-26	13.61
7					.16-24	11.73					.20-29	15.66

□

The above observations show the method of deferred corrections to be a strong candidate for actual implementation. With the error estimation capabilities, it is straightforward to do adaptive mesh selection. Indeed, by comparing $\mathbf{y}_i^{(l)} - \mathbf{y}_i^{(l-1)}$ on successive meshes [as $\mathbf{y}(x_i) - \mathbf{y}_i^{(l-1)}$ was compared in Table 5.8 above], we can see if the theoretical orders of convergence are being realized and thus adapt the order k to our particular solution's smoothness.

Remark

It is useful at this point to make a superficial comparison of some aspects of the deferred corrections method and a high-order collocation method, two of our favorite global methods. But first, a word of caution: It is very difficult to "compare" methods broadly (i. e., it is difficult to make definitive statements), especially for solving BVPs, because of the large number of factors involved. For example, the class of BVPs selected for comparison, the type of linear system solver used and the way it is implemented, the storage requirements, the output (say, a continuous versus discrete solution and the number of points where solution values are desired), and the ease of implementation are but a few of the factors which are difficult to assess.

For deferred corrections, steps 2.2 and 2.3 are required at each intermediate step l, $2 \le l \le k$, while the fixed-order collocation method requires nonlinear iteration on the one large system. For BVPs with smooth solutions, Example 4.11 above shows how easily we can obtain extremely high accuracy with the acceleration method, so deferred correction would generally be preferable for nontrivial but simple problems. For less simple BVPs, the situation is not so straightforward. A potential difficulty of all acceleration methods is that the initial mesh π has to be sufficiently fine for the basic method to make sense on it. (Recall that our theory requires that h be "small enough".) For some applications this implies a dense mesh π (e.g., see Section 10.3.2) and the collocation solution could be obtained more cheaply than even the basic solution $\mathbf{y}_\pi^{(1)}$. If mesh selection is used, collocation requires no mesh restrictions like (5.92) [and certainly not (5.113)] because of its local approximation nature. The deferred corrections scheme above uses $2k$ adjacent mesh points for its difference formulas, and so has difficulties in a transition region where the mesh changes from coarse to fine. Finally, we mention that in the nonlinear BVP case, a situation similar to that for deferred corrections occurs for a high-order collocation scheme with mesh selection. Namely,

most Newton iterations are usually done on a coarse initial mesh, and for refined meshes a good initial guess for the nonlinear iteration is available from the solution on the previous mesh. One full Newton iteration plus a few fixed Jacobian corrections (cf. Section 8.2) are normally sufficient on all but the coarsest mesh when using collocation. □

The basic idea of the approach for showing convergence of the deferred corrections method is fairly simple: Suppose that $\mathbf{y}_\pi^{(l-1)}$ is accurate to $O(h^{2l-2})$ and that $\mathbf{y}_\pi^l$ is defined from (5.110). Stability of the basic scheme $\mathbf{N}_\pi$ implies

$$\begin{aligned} |\mathbf{y}_i^{(l)} - \mathbf{y}(x_i)| &\le K \max_{1\le i\le N} |\mathbf{N}_\pi \mathbf{y}_i^{(l)} - \mathbf{N}_\pi \mathbf{y}(x_i)| \\ &\le K \max_{1\le i\le N} \{ |\mathbf{C}_l(\mathbf{y}_i^{(l-1)}) - \mathbf{C}_l(\mathbf{y}(x_i))| + |\mathbf{C}_l(\mathbf{y}(x_i)) - \mathbf{N}_\pi \mathbf{y}(x_i)| \} \\ &\le K \max_{1\le i\le N} |\mathbf{C}_l(\mathbf{y}_i^{(l-1)}) - \mathbf{C}_l(\mathbf{y}(x_i))| + K \max_{1\le i\le N} |\boldsymbol{\tau}_i^l(\mathbf{y})| \end{aligned} \tag{5.114}$$

from (5.106) and (5.107). Thus $\mathbf{y}_\pi^{(l)}$ is potentially accurate to $O(h^{2l})$, depending upon how well $\mathbf{C}_l(\mathbf{y}_i^{(l-1)})$ approximates the truncation error [see (5.111)] and how large $|\boldsymbol{\tau}_i^l(\mathbf{y})|$ is. The actual analysis is complicated by the fact that the accuracy of $\mathbf{C}_l(\mathbf{y}_i^{(l-1)})$ depends upon the "smoothness" of $\mathbf{y}_\pi^{(l-1)}$. This smoothness is a discrete analogue of continuity of the function and some of its derivatives. Using divided differences of mesh values, it is adversely affected when the difference formula at a point near the boundary switches from centered to noncentered. A similar situation holds for $\boldsymbol{\tau}_i^l(\mathbf{y})$. A careful description of these smoothness conditions becomes quite technical, and we give references at the end of the chapter. One attractive feature of the method of proof which uses (5.112) and (5.114) (instead of asymptotic expansions) is its "robustness"—the only restriction placed upon an initial iterate $\mathbf{z}^{l,1}$ at any stage is that it has to be sufficiently close to the solution. This is realistic, for example, if $\mathbf{z}^{l,1}$ is an interpolate of a solution from a different *basically unrelated* mesh, as happens with mesh selection. Assuming an asymptotic expansion (5.91) for the error, on the other hand, is less natural in such a case. Indeed, we are not really using a sequence of related meshes as we did for extrapolation.

Remarks

(a) While we have described deferred corrections for the particular scheme (5.107), it should be appreciated that the method and convergence approach (5.114) are applicable for virtually any higher-order scheme. In particular, one could use deferred corrections with one of the implicit Runge-Kutta schemes of Section 5.3 as a basic scheme. Efficient implementation and analysis are far from straightforward, especially for nonlinear BVPs, but nevertheless such an approach has a number of possible merits. In addition to the opportunity for utilizing the advantages of each, the corrections are now local. Thus, one would expect only one iteration (5.112) per step (i. e., $I_l = 1$ for all l).

(b) Deferred corrections (5.112) can be interpreted as a matrix fixed-point iteration (cf. also Section 2.3.2) under fairly general circumstances. For instance, for a linear problem (with appropriate boundary conditions) if the basic scheme looks like

$$\mathbf{L}_\pi \mathbf{y}_\pi^{(1)} = \mathbf{q}_\pi \tag{5.115}$$

and the higher-order scheme is

$$(\mathbf{L}_\pi - \mathbf{L}_{\pi,l})\mathbf{y}_\pi^{(k)} = \mathbf{q}_\pi \tag{5.116}$$

then the deferred corrections iteration can be written as

$$\mathbf{z}^{l,j+1} = \mathbf{z}^{l,j} - \mathbf{L}_\pi^{-1}[(\mathbf{L}_\pi - \mathbf{L}_{\pi,l})\mathbf{z}^{l,j} - \mathbf{q}_\pi] \tag{5.117}$$

(check!). Letting $\mathbf{v}^{l,j} := [(\mathbf{L}_\pi - \mathbf{L}_{\pi,k})\mathbf{z}^{l,j} - \mathbf{q}_\pi]$, this is

$$\mathbf{v}^{l,j+1} = \mathbf{L}_{\pi,k}\mathbf{L}_\pi^{-1}\mathbf{v}^{l,j} \tag{5.118}$$

Now, $\mathbf{L}_\pi^{-1}\mathbf{v}^{l,j}$ has the same magnitude as $\mathbf{v}^{l,j}$ and can be shown to be in some sense "smoother" (as $\mathbf{L}^{-1}\mathbf{v}$ is smoother than $\mathbf{v}$, cf. Section 2.8). It can be shown that applying $\mathbf{L}_{\pi,k}$ makes $\mathbf{v}^{l,j+1}$ smaller. □

5.6 HIGHER-ORDER ODEs

A brief glance at the examples in Chapter 1 suggests that in many applications ODEs appear in the form of mixed-order systems rather than as first-order systems. It is therefore natural to look for numerical methods to approximate BVPs for higher-order ODEs. A simple finite difference method for a second-order ODE has already been introduced in Section 5.1.1. The tridiagonal linear system of equations which results has a very attractive form, accounting for the popularity of the scheme and its extensions to PDEs. But it is not immediately apparent how to generalize the scheme for a nonuniform mesh, and this is discussed in Section 5.6.1.

Usually a BVP appears in a more involved form than one to which the scheme in Section 5.6.1 can be applied. At the same time, a transformation from a general mixed-order form to a first order form is rather straightforward (see Section 1.1.2), and for the latter some very powerful techniques have already been introduced in this chapter. Thus, the rest of the section focuses on methods for a higher-order ODE which are sufficiently flexible and robust to be of general-purpose use.

Under these restrictions, few methods remain. We choose to consider a collocation method which is a generalization of the method introduced in Section 5.4 for first-order ODEs. In Section 5.6.2 we introduce the method for a higher-order ODE and show that, indeed, some efficiency can be gained by applying it directly to the higher-order ODE, rather than to the transformed first-order system. In Section 5.6.3 we consider other good implementations for the same collocation method, and in Section 5.6.4 we discuss the conditioning of the various implementations introduced.

Let us introduce the general form of the BVP considered here. The differential equation of order m comes in two flavours: linear,

$$Lu \equiv u^{(m)} - \sum_{l=1}^{m} c_l(x)u^{(l-1)} = q(x), \qquad a < x < b \tag{5.119a}$$

and nonlinear,

$$Nu \equiv u^{(m)} - f(x, u, u', \ldots, u^{(m-1)}) = 0, \qquad a < x < b \tag{5.120a}$$

Denoting

$$\mathbf{y}(x) := (u(x), u'(x), \ldots, u^{(m-1)}(x))^T \tag{5.121}$$

(this is the vector of unknowns for the corresponding first-order system—see Section 1.1.2) we can write (5.119a) as

$$Lu \equiv u^{(m)} - \mathbf{c}^T(x)\,\mathbf{y} = q(x)$$

and (5.120a) as

$$Nu \equiv u^{(m)} - f(x, \mathbf{y}) = 0$$

As usual, the linear form of the two-point BC is

$$B_a \mathbf{y}(a) + B_b \mathbf{y}(b) = \boldsymbol{\beta} \tag{5.119b}$$

while the nonlinear form is

$$\mathbf{g}(\mathbf{y}(a), \mathbf{y}(b)) = \mathbf{0} \tag{5.120b}$$

It is assumed that the coefficients $c_l(x)$ of the linear operator L are smooth and that there is a unique solution u for (5.119), $u(x)$ generally has m more continuous derivatives than the inhomogeneity $q(x)$. Thus, if $q(x)$ is a piecewise continuous function then $u \in C^{(m-1)}[a, b]$.

5.6.1 More on a simple second-order scheme

In Section 5.1.1 we present a simple scheme for a scalar, second-order Dirichlet BVP. The scheme has the advantage of simplicity, being second-order accurate and yielding a tridiagonal system of linear algebraic equations. But in addition to being defined only for a special BVP form, it is defined only on uniform meshes. Here we generalize it to an arbitrary mesh π of (5.1), retaining the second-order accuracy. The way to go about this is not obvious because the usual 3-point formula for u'' gives

$$\frac{2}{h_i + h_{i-1}}\left[\frac{u(x_{i+1}) - u(x_i)}{h_i} - \frac{u(x_i) - u(x_{i-1})}{h_{i-1}}\right] = u''(x_i) + O(h_i^l)$$

where $l = 2$ if $h_i = h_{i-1}$, but $l = 1$ if $h_i \neq h_{i-1}(1 + O(h_i))$. Indeed, if we use this approximation for $u''(x_i)$ in (5.4a), considering a case where $p \equiv 0$ and leaving everything else in (5.4) unchanged, then for an arbitrary mesh a first-order scheme results (i.e., the truncation error grows significantly as compared to a uniform mesh). Still, by a generalization the second-order accuracy can be salvaged, as we show below.

It is convenient to restrict the BVP (5.2) to the form

$$u'' = (p(x)u)' + r(x)u + q(x) \qquad a < x < b \tag{5.122a}$$

$$u(a) = \beta_1, \qquad u(b) = \beta_2 \tag{5.122b}$$

To derive a second-order formula we first integrate (5.122a) and write it as

$$u' = p(x)u + v \tag{5.123a}$$

$$v' = r(x)u + q(x) \tag{5.123b}$$

Then we introduce a *staggered mesh* (see Fig. 5.4) and define approximants $u_i \sim u(x_i)$ and $v_{i+1/2} \sim v(x_{i+1/2})$, $(x_{i+1/2} := x_i + \frac{1}{2}h_i)$ using the midpoint rule.

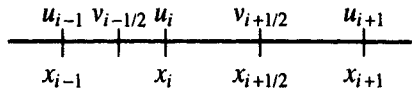

Figure 5.4 Staggered mesh and unknowns

For (5.123a) the midpoint rule on $[x_i, x_{i+1}]$ yields

$$h_i^{-1}[u_{i+1} - u_i] = \frac{1}{2}p_{i+1/2}(u_i + u_{i+1}) + v_{i+1/2} \tag{5.124a}$$

where $p_{i+1/2} := p(x_{i+1/2})$. For (5.123b) the midpoint rule is less standard, being defined on $[x_{i-1/2}, x_{i+1/2}]$. The midpoint of this subinterval is

$$\tilde{x}_i = x_i + \frac{1}{4}(h_i - h_{i-1}) \tag{5.125a}$$

If we approximate $u(\tilde{x}_i)$ in terms of $u(x_{i-1})$, $u(x_i)$ and $u(x_{i+1})$ using quadratic interpolation, this yields

$$\tilde{u}_i := \alpha_i u_{i-1} + \beta_i u_i + \gamma_i u_{i+1} \tag{5.125b}$$

$$\beta_i = \frac{1}{16}\frac{(3h_i + h_{i-1})(h_i + 3h_{i-1})}{h_{i-1}h_i} \tag{5.125c}$$

$$\alpha_i = \frac{1}{16}\frac{(3h_i + h_{i-1})(h_{i-1} - h_i)}{h_{i-1}(h_i + h_{i-1})}, \qquad \gamma_i = \frac{1}{16}\frac{(h_i + 3h_{i-1})(h_i - h_{i-1})}{h_i(h_i + h_{i-1})}$$

Then the midpoint rule approximation for (5.123b) becomes

$$\frac{v_{i+1/2} - v_{i-1/2}}{\frac{1}{2}(h_i + h_{i-1})} = r(\tilde{x}_i)\tilde{u}_i + q(\tilde{x}_i) \tag{5.124b}$$

We may now use (5.124a) to define $v_{i+1/2}$, and substitution into (5.124b) gives

$$\frac{2}{h_{i-1} + h_i}[h_i^{-1}(u_{i+1} - u_i) - h_{i-1}^{-1}(u_i - u_{i-1}) - \frac{1}{2}p_{i+1/2}(u_{i+1} + u_i) + \frac{1}{2}p_{i-1/2}(u_{i-1} + u_i)]$$

$$= r(\tilde{x}_i)\tilde{u}_i + q(\tilde{x}_i) \tag{5.126}$$

This is a 3-point formula involving only u-values and hence generalizing (5.4a).

Remark

This simple derivation actually brings up two curious points not encountered hitherto. One is the staggered mesh (grid): While a tool commonly used in deriving schemes in numerical fluid mechanics, it is rarely needed for the material covered in this book (but see Section 10.3.3).

The other point is that the global error seems at first sight to be of higher-order than the local truncation error! Substituting the exact solution into (5.124a), we obtain the local truncation error term

$$\frac{h_i^2}{24}u'''(x_{i+1/2}) - p_{i+1/2}\frac{h_i^2}{16}u''(x_{i+1/2}) + O(h_i^4)$$

Substituting this into (5.126) yields, corresponding to (5.6), an $O(h)$ local truncation error for an arbitrary mesh. Still, the error is expected to be $O(h^2)$, because the difference scheme (5.124a, b) as a discretization for (5.123a, b) is second-order and stable, and (5.126) (in exact arithmetic) is equivalent to (5.124). The second-order accuracy cannot be directly obtained with the approach in Section 5.1.1. This suggests that a less mysterious way to interpret (5.126) is as a clever implementation of (5.124). □

Substituting (5.125) into (5.126) we obtain the second-order 3-point formula

$$a_i u_{i-1} + b_i u_i + c_i u_{i+1} = g_i \qquad 2 \le i \le N \tag{5.127a}$$

$$a_i = \frac{2}{h_{i-1}(h_i + h_{i-1})} + \frac{p_{i-1/2}}{h_i + h_{i-1}} - r(\tilde{x}_i)\alpha_i \tag{5.127b}$$

$$c_i = \frac{2}{h_i(h_i + h_{i-1})} - \frac{p_{i+1/2}}{h_i + h_{i-1}} - r(\tilde{x}_i)\gamma_i \tag{5.127c}$$

$$b_i = -\frac{2(h_i^{-1} + h_{i-1}^{-1})}{h_i + h_{i-1}} + \frac{p_{i-1/2} - p_{i+1/2}}{h_i + h_{i-1}} - r(\tilde{x}_i)\beta_i \tag{5.127d}$$

$$g_i = q(\tilde{x}_i) \tag{5.127e}$$

Note that if $h_i = h_{i-1}$ then

$$\tilde{x}_i = x_i,\ \tilde{u}_i = u_i,\ \alpha_i = \gamma_i = 0,\ \beta_i = 1, \qquad \frac{h_i + h_{i-1}}{2} = h_i$$

and the scheme reduces to the usual one.

5.6.2 Collocation

The method we describe below for one ODE (5.119) or (5.120) can be directly extended to handle mixed-order systems of ODEs with multipoint BCs.

The basic idea of collocation is quite general: An approximate solution is sought in the form

$$u_\pi(x) = \sum_{j=1}^{M} \alpha_j \phi_j(x) \qquad a \le x \le b \tag{5.128}$$

where $\phi_j(x)$ are known linearly independent functions defined on $[a, b]$, and α_j are parameters. These parameters are determined by requiring $u_\pi(x)$ to satisfy M conditions—the m BC and the ODE at $M-m$ points (the *collocation points*) in $[a, b]$. It is sometimes convenient to say that $u_\pi(x)$ is an element in a *linear space* of *dimension* M which is *spanned* by *basis functions* $\phi_1(x), \ldots, \phi_M(x)$.

For the remainder of this chapter we choose this linear space to consist of piecewise polynomial functions. Thus, there is a partition π of $[a, b]$ as in (5.1) [justifying the notation u_π in (5.128)] such that any linear combination of the basis functions reduces to a polynomial on each subinterval $[x_i, x_{i+1}]$, $1 \le i \le N$. In this section we further restrict the functions $\phi_j(x)$, and therefore any of their linear combinations, to be in $C^{(m-1)}[a, b]$ (like the exact solution). Further, the order of the polynomial pieces is restricted to be $k+m$, for some $k \ge m$. Recall that $v(x)$ is in $\mathbf{P}_{k+m,\pi}$ if $v(x)$ is a piecewise polynomial of order $k+m$ on π.

The dimension M of the resulting approximation space is easy to calculate: Each polynomial piece has $k+m$ parameters and there are m matching constraints for y across each interior mesh point, so

$$M = N(k+m) - (N-1)m = Nk + m \tag{5.129}$$

The M conditions imposed on $u_\pi(x)$ of (5.128) are:

(a) That u_π satisfy the m BC (5.119b) or (5.120b).

(b) That u_π satisfy the ODE (5.119a) or (5.120a) at k points in each of the N subintervals of the mesh π. These points are the collocation points x_{ij} of (5.59c, d) with the canonical points ρ_j being distinct.

Let us summarize this for the general BVP (5.120), denoting (similarly to (5.121))

$$\mathbf{y}_\pi(x) := (u_\pi(x), u_\pi'(x), \ldots, u_\pi^{(m-1)}(x))^T \tag{5.130}$$

Given a mesh π of (5.1) and k distinct points $\rho_1, \ldots, \rho_k$ of (5.59d), the collocation method determines an approximate solution $u_\pi(x)$ defined on $[a, b]$ such that

$$\mathbf{y}_\pi(x) \in C[a, b]; \qquad u_\pi(x) \in \mathbf{P}_{k+m,\pi} \tag{5.131a}$$

$$\mathbf{g}(\mathbf{y}_\pi(a), \mathbf{y}_\pi(b)) = \mathbf{0} \tag{5.131b}$$

$$Nu_\pi(x_{ij}) \equiv u_\pi^{(m)}(x_{ij}) - f(x_{ij}, \mathbf{y}_\pi(x_{ij})) = 0 \qquad 1 \le j \le k, \qquad 1 \le i \le N \tag{5.131c}$$

One important question is how to choose the basis functions $\phi_j(x)$ so as to obtain an efficient, stable method. Choices of Hermite-type bases and of B-splines (recall Section 2.5) are discussed in Section 5.6.3. With these we specify basis functions $\phi_j(x)$ with local support, and the continuity conditions on u_π are already imbedded in the basis functions, while the collocation equations are satisfied only later. But here we prefer not to explicitly specify any basis functions. Instead, we consider local representations of the polynomial pieces. It enables us to relate more directly to the collocation method for first-order systems already discussed, and to see that the method introduced here is just a fancy finite difference method. We first impose the collocation equations,

followed by local parameter elimination, and only then connect to the action in adjacent subintervals. This is a multiple-shooting-type approach which allows us to capitalize on previous theoretical results and to avoid introducing heavier functional analysis machinery.

Let us focus on one subinterval $[x_i, x_{i+1}]$ and express the polynomial $u_\pi(x)$ in terms of its Taylor series about x_i,

$$u_\pi(x) = \sum_{j=1}^{k+m} \frac{(x-x_i)^{j-1}}{(j-1)!} u_\pi^{(j-1)}(x_i) \qquad x_i \le x \le x_{i+1}$$

This can be written as

$$u_\pi(x) = \sum_{j=1}^{m} \frac{(x-x_i)^{j-1}}{(j-1)!} y_{ij} + h_i^m \sum_{j=1}^{k} \psi_j\left(\frac{x-x_i}{h_i}\right) z_{ij} \tag{5.132}$$

where, corresponding to (5.130),

$$\mathbf{y}_\pi(x_i) \equiv \mathbf{y}_i = (y_{i1}, \ldots, y_{im})^T$$

and

$$z_{ij} = h_i^{j-1} u_\pi^{(m+j-1)}(x_i), \qquad \mathbf{z}_i := (z_{i1}, \ldots, z_{ik})^T$$

The functions $\psi_j(t)$ are therefore defined as

$$\psi_j(t) = \frac{t^{m+j-1}}{(m+j-1)!} \qquad 0 \le t \le 1, \qquad 1 \le j \le k \tag{5.133}$$

Note that $\psi_1(t), \ldots, \psi_k(t)$ are linearly independent polynomials of order $k+m$ on [0, 1] satisfying

$$\psi_j^{(l-1)}(0) = 0 \qquad 1 \le l \le m, \qquad 1 \le j \le k \tag{5.134}$$

These functions are independent of i and can be used for each subinterval as in (5.132).

Now, we can write down the constraints (5.131) which define the approximate solution $u_\pi(x)$ in terms of the parameters $\mathbf{y}_i$ and $\mathbf{z}_i$ of the representation (5.132). We do this for a linear problem first.

Linear problems For the linear problem (5.119) we write

$$Lu_\pi(x) = h_i^m \sum_{j=1}^{k} z_{ij} L\left[\psi_j\left(\frac{x-x_i}{h_i}\right)\right] - \sum_{l=1}^{m} c_l(x) \sum_{j=l}^{m} \frac{y_{ij}(x-x_i)^{j-l}}{(j-l)!}$$

so the collocation conditions (5.131c) give

$$V\mathbf{y}_i + W\mathbf{z}_i = \mathbf{q}_i \qquad 1 \le i \le N \tag{5.135a}$$

where $\mathbf{q}_i = (q(x_{i1}), \ldots, q(x_{ik}))^T$, V is a $k \times m$ matrix with entries

$$V_{rj} = -\sum_{l=1}^{j} \frac{c_l(x_{ir})(h_i \rho_r)^{j-l}}{(j-l)!} \qquad 1 \le r \le k, \qquad 1 \le j \le m \tag{5.136a}$$

and W is a $k \times k$ matrix with entries

$$W_{rj} = \psi_j^{(m)}(\rho_r) - \sum_{l=1}^{m} c_l(x_{ir}) h_i^{m+1-l} \psi_j^{(l-1)}(\rho_r) \qquad 1 \le r, j \le k \tag{5.136b}$$

The continuity conditions in (5.131a) are even easier to write down. We evaluate $u_\pi(x)$ and its first $m-1$ derivatives at $x = x_{i+1}$ by (5.132) and equate to $\mathbf{y}_{i+1}$, the corresponding values at the $(i+1)$st subinterval. This yields

$$\mathbf{y}_{i+1} = C\mathbf{y}_i + D\mathbf{z}_i \qquad 1 \le i \le N \tag{5.135b}$$

where C is an $m \times m$ upper triangular matrix with entries

$$C_{rj} = \frac{h_i^{j-r}}{(j-r)!} \qquad j \ge r \tag{5.136c}$$

and D is an $m \times k$ matrix with entries

$$D_{rj} = h_i^{m+1-r} \psi_j^{(r-1)}(1) \qquad 1 \le r \le m, \qquad 1 \le j \le k \tag{5.136d}$$

Note that the obvious dependence of C, D, W, and V on i has been suppressed in the notation.

The specification of the collocation constraints (5.131) for the linear problem is completed by writing for (5.131b)

$$B_a \mathbf{y}_1 + B_b \mathbf{y}_{N+1} = \boldsymbol{\beta} \tag{5.135c}$$

The reader should realize that, despite the unfortunate number of subscripts which we have to use here, there is no significant conceptual difference between this process and (5.64). Our next step is to eliminate the local unknowns, in a similar way to that used to obtain (5.65). We note that as $h \to 0$ in (5.136b),

$$W_{rj} \to \rho_r^{j-1}/(j-1)! =: W_{rj}(0)$$

The matrix $W(0)$ is a nonsingular Vandermonde matrix, so for h_i small enough W is nonsingular and

$$W^{-1} = W(0)^{-1} + O(h_i)$$

Eliminating $\mathbf{z}_i$ from (5.135a) and substituting in (5.135b), we arrive at the form

$$\mathbf{y}_{i+1} = \Gamma_i \mathbf{y}_i + \mathbf{r}_i \qquad 1 \le i \le N \tag{5.137a}$$

with

$$\Gamma_i := C - DW^{-1}V, \qquad \mathbf{r}_i := DW^{-1}\mathbf{q}_i \tag{5.137b}$$

The similarity between (5.137) and (5.65) is quite clear, noting that $C = I + O(h_i)$ and $D = O(h_i)$.

In (5.137), (5.135c) we have, once again, a linear system of equations of the form (5.17) for $u_\pi(x)$ and its first $m-1$ derivatives at mesh points. After obtaining the values of $\mathbf{y}_i$ we can easily obtain $\mathbf{z}_i$ from (5.135a), and hence $u_\pi(x)$ from (5.132), if we store the values of $W^{-1}V$ and $W^{-1}\mathbf{q}_i$ which are computed while assembling each Γ_i, $1 \le i \le N$.

We summarize the collocation procedure outlined above in an algorithm. For the efficient assembly of Γ_i and $\mathbf{r}_i$ we compute and store in advance all mesh independent quantities like $\psi_j^{(l)}(\rho_r)$, $\psi_j^{(l)}(1)$, ρ_r^j and $j!$. The distinct points $\rho_1, \ldots, \rho_k$ are therefore determined at preprocessing time. Then we obtain the following algorithm.

Algorithm: Collocation for a linear higher-order BVP

Input: A BVP (5.119) and a mesh π.

Output: Solution values $\mathbf{y}_1, \ldots, \mathbf{y}_{N+1}$ consisting of $u_\pi(x)$ and its first $m-1$ derivatives at mesh points.

1. {Assemble the difference equations}
   FOR $i = 1, \ldots, N$ DO
   Assemble V, W, C and D according to (5.136a–d).
   Find $U_i := W^{-1}V$ and $\mathbf{p}_i := W^{-1}\mathbf{q}_i$.
   Set $\Gamma_i := C - DU_i$; $\mathbf{r}_i := D\mathbf{p}_i$.
2. Set up and solve the linear system

$$\begin{bmatrix} -\Gamma_1 & I & & & \\ & -\Gamma_2 & I & & \\ & & \ddots & \ddots & \\ & & & -\Gamma_N & I \\ B_1 & & & & B_2 \end{bmatrix} \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_N \\ \mathbf{y}_{N+1} \end{bmatrix} = \begin{bmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \vdots \\ \mathbf{r}_N \\ \boldsymbol{\beta} \end{bmatrix}$$

3. Optionally, obtain $\mathbf{z}_i$ values from

$$\mathbf{z}_i := \mathbf{p}_i - U_i \mathbf{y}_i \qquad 1 \le i \le N$$

□

Example 5.15

The BVP presented in Example 5.1 is a scalar second-order problem. In Example 5.7 we have computed approximate solutions, using various collocation schemes for the transformed first-order BVP, and listed the results in Table 5.4. We repeat these calculations here, using the corresponding collocation schemes for the original problem formulation. The result is that the obtained maximum errors at mesh points for $u_\pi(\equiv y_1)$ and $u_\pi'(\equiv y_2)$ are *identical* (to the number of digits shown) to those listed in Table 5.4. □

Example 5.16

The following BVP describes, under suitable assumptions, a uniformly loaded beam of variable stiffness, simply supported at both ends:

$$(x^3 u'')'' = 1 \qquad 1 < x < 2$$

$$u(1) = u''(1) = u(2) = u''(2) = 0$$

The exact solution is

$$u(x) = \frac{1}{4}(10\ln 2 - 3)(1-x) + \frac{1}{2}\left[\frac{1}{x} + (3+x)\ln x - x\right]$$

so this BVP too is well-behaved and has a rather smooth solution. Here we demonstrate the power of high-order methods for very smooth examples. Using Gauss points on uniform meshes, the maximum error in u at mesh points is $.24-13$ for $k = 4$, $N = 16$, and $.96-14$ for $k = 6$, $N = 4$. □

Remark The representation for the polynomial $u_\pi(x)$ in (5.132) is not unique. In particular, other choices for $\psi_j(t)$ satisfying (5.134) can be considered. Two such choices are as follows: Requiring

$$\psi_j^{(m)}(\rho_r) = \delta_{jr} \quad 1 \le j, r \le k \tag{5.138a}$$

implies $z_{ij} = u_\pi^{(m)}(x_{ij})$. (Why?). It is easy to see that $\psi_j(t)$ is well-defined by (5.138a) and (5.134). Moreover, $W(0) = I$, and for $m = 1$ we have

$$\psi_j(\rho_r) = \alpha_{rj}, \qquad \psi_j(1) = \beta_j$$

with α_{rj}, β_j given by (5.68). We call this *a Runge-Kutta representation.* Another choice is

$$\psi_j^{(r-1)}(1) = \delta_{j-k+m,r} \qquad 1 \le j, r \le k \tag{5.138b}$$

The importance of this is that D of (5.136d) becomes very simple. But for a mixed-order system of ODEs, the Runge-Kutta representation is preferred, because the same functions may be used for more than one order (see Exercise 21). □

The results in Example 5.15 indicate a rather strong connection between the collocation method for the higher-order BVP (5.119) and the corresponding collocation method using the same k points ρ_j ($k \ge m$) for the transformed first-order system

$$\mathbf{y}' = \begin{bmatrix} 0 & 1 & & & \\ & & \ddots & & \\ & & & 0 & 1 \\ c_1(x) & c_2(x) & \cdots & & c_m(x) \end{bmatrix} \mathbf{y} + \begin{bmatrix} 0 \\ \vdots \\ 0 \\ q(x) \end{bmatrix} \tag{5.139}$$

and (5.119b). Indeed, the linear equations (5.137a) form a *one-step difference scheme* for (5.139). Two questions immediately arise: (a) Have we gained anything in the treatment here; indeed, is anything different from the corresponding collocation methods of Section 5.3? and (b) Can we capitalize on the theory of Sections 5.2 and 5.3?

Clearly we should answer question (a) before spending time on (b). We first point out that while here $u_\pi(x) \in \mathbf{P}_{k+m,\pi} \cap C^{(m-1)}[a, b]$, the corresponding collocation approximation of y_1 in (5.139) is in $\mathbf{P}_{k+1,\pi} \cap C[a, b]$, so the approximations are generally not the same when $m > 1$. Next, note that in (5.132) we have $k + m$ parameters per mesh subinterval, whereas the corresponding method for a first-order system would have $m(k+1)$, i. e., $(m-1)k$ additional parameters. Of course the treatment of the parameters is slightly more cumbersome here; still, the matrix W of (5.64) for (5.139) is $mk \times mk$ and A_i is $mk \times m$, whereas in (5.136) W is merely $k \times k$ and V is $k \times m$. If $m = 4$ as in Example 5.16, for instance, then the saving in computation and storage is rather substantial. We can view (5.137) as *a sophisticated high-order one-step scheme for (5.139), which takes advantage of the special structure of the coefficients in the transformed first-order system.* We hasten to point out, however, that this interpretation is beneficial only for mesh point values.

The answer to question (b) is affirmative, as indicated by the proof to the following.

Theorem 5.140 Assume that there are integers $p \geq k \geq m$ such that

(a) the linear BVP of order m (5.119) is well-posed, in the sense that the equivalent first-order BVP (5.139), (5.119b) has a conditioning constant κ of moderate size; (5.119) has coefficients in $C^{(p)}[a, b]$ and has a unique solution $u(x)$ in $C^{(p+m)}[a, b]$; and

(b) the k canonical collocation points $\rho_1, \ldots, \rho_k$ satisfy the orthogonality conditions

$$\int_0^1 \phi(t) \prod_{l=1}^{k} (t - \rho_l)\, dt = 0 \qquad \phi \in \mathbf{P}_{p-k} \tag{5.140a}$$

(note that $p \leq 2k$).

Then for h small enough the collocation method given in the above algorithm is stable with stability constant $c\kappa N$, c a constant of moderate size, and it has a unique solution $u_\pi(x)$. Furthermore, the following error estimates hold: At mesh points

$$|u^{(j)}(x_i) - u_\pi^{(j)}(x_i)| = O(h^p) \qquad 0 \leq j \leq m-1, \qquad 1 \leq i \leq N+1 \tag{5.140b}$$

while elsewhere, for $1 \leq i \leq N$,

$$|u^{(j)}(x) - u_\pi^{(j)}(x)| = O(h_i^{k+m-j}) + O(h^p) \qquad x_i \leq x \leq x_{i+1}, \qquad 0 \leq j \leq k+m-1 \tag{5.140c}$$

Proof: First we show that the analogue of Theorem 5.75 holds. It is easy to see, by looking at a local Taylor expansion, that $u(x)$ has the representation (5.132) up to $O(h_i^{k+m})$ terms. From this we obtain (Exercise 18) that $\mathbf{y}(x)$ of (5.121), which is the exact solution of the first-order BVP (5.139), (5.119b), satisfies

$$\mathbf{y}(x_{i+1}) = \Gamma_i \mathbf{y}(x_i) + \mathbf{r}_i + O(h_i^{k+1})$$

This means that the finite difference scheme (5.137a), (5.135c) is a one-step scheme for (5.139), (5.119b) which is consistent of order k. The sound construction of Γ_i of (5.137) allows application of Theorem 5.38. Thus we obtain existence of a unique collocation solution and stability with the claimed stability bound. Moreover, the error at mesh points satisfies

$$|\mathbf{y}_i - \mathbf{y}(x_i)| = O(h^k) \qquad 1 \leq i \leq N+1 \tag{5.141a}$$

Let us use the notation $d_\pi(x) := u_\pi(x) - u(x)$ and write as in (5.132)

$$d_\pi(x) = \sum_{j=1}^{m} \frac{(x - x_i)^{j-1}}{(j-1)!} e_{ij} + h_i^m \sum_{j=1}^{k} \psi_j\left(\frac{x - x_i}{h_i}\right) g_{ij} - \mu(x) \tag{5.141b}$$

with

$$(e_{i1}, \ldots, e_{im})^T = \mathbf{e}_i = \mathbf{y}_i - \mathbf{y}(x_i)$$

and $\mu^{(j)}(x) = O(h_i^{k+m-j})$, $0 \leq j \leq k+m-1$. Equations (5.135a) for the error give the estimate

$$|g_{ij}| = O(h^k)$$

so taking derivatives in the expression (5.141b) for $d_\pi(x)$, we obtain

$$|u_\pi^{(j)}(x) - u^{(j)}(x)| = \begin{cases} O(h^k) & 0 \le j \le m \\ & x_i \le x \le x_{i+1} \\ O(h^{k+m-j})\theta_i^{j-m} & m \le j \le k+m-1 \end{cases} \tag{5.141c}$$

with θ_i defined in (5.75c).

We now proceed as in (5.81). Let $G(x,t)$ be Green's function for (5.119) and denote $G_j(x,t) \equiv \dfrac{\partial^j}{\partial x^j} G(x,t)$. Then from the assumption on the problem coefficients, $G_j(x,t)$ is smooth as a function of t in $[a,x)$ and in $(x,b]$; but at $t=x$, $G_j(x,t)$ has only $m-1-j$ derivatives. Writing for $0 \le j \le m-1$

$$u_\pi^{(j)}(x) - u^{(j)}(x) = \int_a^b G_j(x,t)\, L(u_\pi(t) - u(t))\, dt = \sum_{i=1}^{N} E_i(x)$$

with

$$E_i(x) = \int_{x_i}^{x_{i+1}} G_j(x,t)(Lu_\pi(t) - q(t))\, dt$$

we note that $Lu_\pi(t) - q(t) = 0$ at k collocation points $t = x_{il}$. Thus, if $x \notin (x_i, x_{i+1})$ then by (5.140a) and (5.140d)

$$E_i(x) = O(h_i^{p+2-k} h^{k-1}) \le \text{const}\, h_i h^p \tag{5.141d}$$

while if $x \in (x_i, x_{i+1})$ then the limited smoothness of $G_j(x,t)$ allows us to conclude only that

$$E_i(x) = O(h_i^{k+m-j})\theta_i^{k-1} \tag{5.141e}$$

The conclusion (5.140b) follows, because for a mesh point each $E_i(x)$ satisfies (5.141d) and so $\sum_{i=1}^{N} E_i(x) = O(h^p)$. But to obtain (5.140c) without imposing any mesh restriction, we cannot just use (5.141e). Instead, knowing that (5.140b) holds, we again use (5.135a) for the error, obtaining

$$|g_{ij}| = O(h^p) + O(h_i^k)$$

and substitute into (5.141b). This yields (5.140c) and completes the proof. □

Remarks

(a) The result (5.140c) shows that, when x is not a mesh point, $u_\pi(x)$ is a *better approximation* to $u(x)$ than the corresponding collocation approximation for the equivalent first-order system.

(b) The approximation space to which $u_\pi(x)$ belongs is a linear space of piecewise polynomials of order $k+m$. A result from approximation theory states that, unless $u(x)$ itself is in the approximation space, we cannot get a global approximation order of more than $O(h^{k+m})$. In other words, for any piecewise polynomial $v_\pi(x)$ of order $k+m$, we cannot guarantee a better result than

$$\max_{a \le x \le b} | v_\pi(x) - u(x) | = O(h^{k+m})$$

If $p \ge k+m$ then the estimate (5.140c) shows that the collocation solution achieves the *optimal* global convergence order. Furthermore, if $p > k+m$ then at mesh points we obtain a *superconvergence* order, i. e., an order of convergence higher than the best possible global order. It is interesting to note that the superconvergence result, which (as the name suggests) is perhaps less natural from the point of view of approximation spaces, is most natural from the point of view of one-step difference schemes.

(c) A restriction on the mesh under which the convergence results of Theorem 5.140 hold is remarkably absent. We have not assumed that the mesh is quasiuniform (recall Section 5.5). Also, Theorem 5.38 which was used in the proof of (5.140) indicates that the condition number of the matrix **A** involved is $O(N)$. In many methods for higher-order BVPs one finds condition numbers which depend on $\frac{h}{\min h_i}$ (see Section 5.6.4, for instance) and/or on N^m (see Exercise 1).

(d) The proof of Theorem 5.140 yields an $O(N)$ conditioning for the approximate solution $\mathbf{y}_\pi(x)$ of the first-order system (5.139). To obtain from this a similar conditioning for the approximate solution $u_\pi(x)$ of a given BVP (5.119), note that it is easy to bound

$$\| u_\pi^{(j)} \| \le K \| u_\pi \|, \qquad 1 \le j \le m-1$$

where K is a constant which depends on κ and on $\frac{\| q \| + |\boldsymbol{\beta}|}{\| u \|}$, but not on the mesh (for $h > 0$ small enough).

(e) The requirements of Theorem 5.140 include fairly strong (but not unusual) smoothness assumptions on the BVP coefficients. These assumptions can be relaxed to allow a finite number of jump discontinuities, *provided* that mesh points are placed at each such discontinuity. Then $u_\pi(x)$ will be a polynomial in regions where $u(x)$ is highly smooth, but merely in $C^{(m-1)}[a, b]$ globally, just like $u(x)$. For instance, Example 4.7 can be solved in this way. The method cannot handle δ-functions in the coefficients, though. (Why?) □

While the superconvergence in (5.140b) is of a high-order, the global convergence result in (5.140c) has a *local* nature (at least as long as $h_i^{k+m} \gg h^p$). This is important for mesh selection. In fact, if $p > k+m$ then we can refine (5.140c) by explicitly giving the leading term of the error:

Corollary 5.142 Let $p > k+m$. With the assumptions of Theorem 5.140, the collocation error satisfies

$$u^{(j)}(x) - u_\pi^{(j)}(x) = h_i^{k+m-j} u^{(k+m)}(x_i) P^{(j)}\left(\frac{x - x_i}{h_i}\right) + O(h_i^{k+m-j+1}) + O(h^p) \quad (5.142a)$$

$$x_i \le x \le x_{i+1}, \qquad 1 \le i \le N, \qquad 0 \le j \le k+m-1$$

where

$$P(\xi) = \frac{1}{k!(m-1)!} \int_0^{\xi} (t-\xi)^{m-1} \prod_{l=1}^{k} (t-\rho_l)\, dt \tag{5.142b}$$

Proof: We will show (5.142a) for $j = 0$; the general case follows by taking derivatives in (5.141b). First, since both u_π and u satisfy the ODE at collocation points, it follows from (5.140c) that

$$u^{(m)}(x_{ir}) - u_\pi^{(m)}(x_{ir}) = O(h_i^{k+1}) + O(h^p) \qquad 1 \le r \le k \tag{5.143}$$

It is therefore useful to consider here the Runge-Kutta choice (5.138a) for the functions $\psi_j(t)$, because the leading term of the error representation (5.141b) is then $\mu(x)$. The latter is the residual of $u(x)$ substituted in (5.132), so

$$\mu^{(m)}(x) = u^{(m)}[x_{i1}, \ldots, x_{ik}, x] \prod_{l=1}^{k} (x - x_{il})$$

$$\mu(x_i) = \cdots = \mu^{(m-1)}(x_i) = 0$$

Noting that $u^{(m)}[x_{i1}, \ldots, x_{ik}, x] = \frac{1}{k!} u^{(k+m)}(x_i) + O(h_i)$ and integrating m times for $\mu(x)$ yields the required result. □

The above corollary indicates that, if we are willing to "give up" the superconvergence order, then we get a *localized representation for the global error.* This is unlike the usual case, where the error can only be bounded; also, the leading error term depends on local quantities in (5.142). An estimate of $u^{(k+m)}(x_i)$ would produce an estimate of the error locally. This is utilized in Section 9.3 for mesh selection.

Another consequence of (5.142) is that there are additional "superconvergence" points, i. e., points other than mesh points where the order of convergence is higher than what is possible everywhere. In particular, any roots of $P^{(j)}(\xi)$ in (5.142b) correspond to such points for the jth derivative of the error in (x_i, x_{i+1}), $1 \le i \le N$. We have already mentioned the roots of $P^{(m)}$ in (5.143). Also, $P^{(k+m-1)}(\bar{t}) = 0$ at

$$\frac{\bar{x}_i - x_i}{h_i} \equiv \bar{t} := \frac{1}{k} \sum_{l=1}^{k} \rho_l \tag{5.144a}$$

so the piecewise constant function $u_\pi^{(k+m-1)}$ satisfies

$$u_\pi^{(k+m-1)}(x) = u^{(k+m-1)}(\bar{x}_i) + O(h_i^2) \qquad x_i \le x \le x_{i+1}, \qquad 1 \le i \le N \tag{5.144b}$$

If the collocation points are symmetric then $\bar{x}_i = x_{i+1/2}$.

Nonlinear problems For the nonlinear BVP (5.120) we once again consider the method of quasilinearization. The principle is precisely the same as in Section 5.3.2, but the linearized differential system appears as a higher-order ODE. We denote, as in (5.43)–(5.45),

$$L[u]z(x) \equiv z^{(m)}(x) - \sum_{l=1}^{m} c_l(x) z^{(l-1)}(x) \qquad a < x < b \tag{5.145a}$$

$$c_l(x) := \frac{\partial f(x, \mathbf{y}(x))}{\partial y_l} \tag{5.145b}$$

where u and $\mathbf{y}$ are related by (5.121). The linearization of the BC is as in Section 5.2.2.

Recall how the method is derived: Given a known solution profile $\hat{u}(x)$, we formally expand

$$u^{(m)}(x) = f(x, \mathbf{y}(x)) \approx f(x, \hat{\mathbf{y}}(x)) + \sum_{l=1}^{m} \frac{\partial f}{\partial y_l}(x, \hat{\mathbf{y}}(x))(u^{(l-1)}(x) - \hat{u}^{(l-1)}(x))$$

where the vector $\hat{\mathbf{y}}(x)$ has components $\hat{y}_l := \hat{u}^{(l-1)}$. This yields an approximate *linear* ODE for the correction $z(x) \equiv u(x) - \hat{u}(x)$. The method of collocation with quasilinearization then reads as follows: Given an initial approximate solution $u_\pi(x)$, *repeat* (i) solving by collocation the linearized problem

$$L[u_\pi]z(x) = -[u_\pi^{(m)}(x) - f(x, \mathbf{y}_\pi(x))] \qquad a < x < b \tag{5.146a}$$

$$B_a \mathbf{w}(a) + B_b \mathbf{w}(b) = -\mathbf{g}(\mathbf{y}_\pi(a), \mathbf{y}_\pi(b)) \tag{5.146b}$$

where $\mathbf{w}(x) := (z(x), z'(x), \ldots, z^{(m-1)}(x))$, and (ii) improving the approximate solution by

$$u_\pi(x) := u_\pi(x) + z_\pi(x) \qquad a \le x \le b \tag{5.146c}$$

until $\| z_\pi \|$ is below an error tolerance.

The quasilinearization method outlined above can be seen to be equivalent to Newton's method for solving (5.131).

The basic convergence results for the method of collocation and quasilinearization are a combination of Theorems 5.88 and 5.140: The nonlinear analysis is the same as for Theorem 5.88, while Theorem 5.140 applies for the linearized problem. We state the results but omit the proof.

Theorem 5.147 Let $u(x)$ be an isolated solution of the BVP (5.120), where f and $\mathbf{g}$ have continuous second partial derivatives, and assume that the hypotheses of Theorem 5.140 hold for u and $L[u]$. Consider a k-stage collocation method, satisfying (5.140a), for (5.131). Then there are positive constants ρ and h_0 such that for all meshes with $h \le h_0$:

(a) There is a unique solution $u_\pi(x)$ to the collocation equations (5.131) in a tube $S_\rho(u)$ of radius ρ around $u(x)$.

(b) This solution $u_\pi(x)$ can be obtained by Newton's method, which converges quadratically provided that the initial guess for $u_\pi(x)$ is sufficiently close to $u(x)$.

(c) The error estimates (5.140b, c) and (5.142a) hold. □

Example 5.17

We have solved the nonlinear part of application 1.4 using a 5-stage collocation method with Gaussian points. The BVP (1.10c, b) is rewritten here,

$$u'''' = R[u'u'' - uu''']$$

$$u(0) = u'(0) = 0, \qquad u(1) = 1, \qquad u'(1) = 0$$

so $m = 4$, $k = 5$, $k + m = 9$, and $p = 10$.

This problem is slightly more difficult, when the Reynolds number R is large, than those in other examples in this chapter. The solution profile of $u'(x)$, shown in Fig. 5.5, exhibits a narrow region of fast change near $x = 0$, which gets narrower as R is increased.

Two observations come to mind. The first is that the mesh chosen should reflect this solution behaviour. As in Example 5.2, we make the mesh fine near $x = 0$ and coarse elsewhere. For R = 10,000, 40 mesh points were sufficient to obtain an accuracy of more than 6 significant digits in all components of $\mathbf{y}(x)$.

The second observation is that if we know nothing in advance about the solution profile, then our initial guess may be far from the actual solution, and Newton's method may not converge. This turns out not to be a serious problem here (at least for $R \leq 10{,}000$), but for other examples it is a rather acute difficulty. Figure 5.5 suggests one way to proceed: Solve a number of similar BVPs, gradually increasing R, and using the obtained solution and mesh for one R to start the Newton iteration for the next, larger R. This is a process of simple *continuation*, which we discuss further in Section 8.3. □

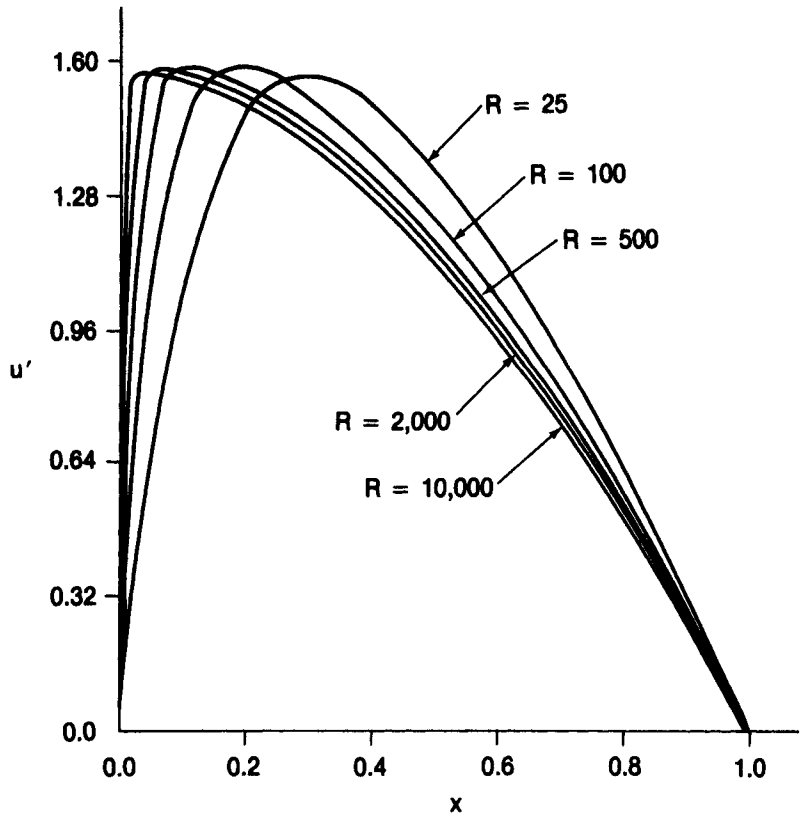

Figure 5.5 Example 5.17, tangential velocity u'

5.6.3 Collocation implementation using spline bases

The collocation algorithm of the previous subsection uses the local representation (5.132), but this is not the only possibility for implementing the method. Here we consider a different idea which arises naturally from an approximation point of view, when the numerical solution is viewed as an element from an approximation space. Thus we wish to construct good basis functions for the piecewise polynomial space in which the collocation solution lies.

The spline collocation method for solving (5.119) or (5.120) can be viewed as follows: We seek a function $u_\pi(x)$ in the space of piecewise polynomial functions with $m-1$ continuous derivatives $\mathbf{P}_{k+m,\pi} \cap C^{(m-1)}[a,b] =: \mathbf{P}_{k+m,\pi,m}$. After choosing a basis $\phi_1(x), \ldots, \phi_M(x)$ for this space, we write $u_\pi(x)$ as a linear combination of these functions as in (5.128) and determine the coefficients $\alpha_1, \ldots, \alpha_M$ by requiring $u_\pi(x)$ to satisfy (5.131b,c). Thus, collocation is a generalization of function interpolation, where we now "interpolate" (i. e., collocate) an ODE. Relative to the other global methods, collocation is in a sense midway between finite difference methods and finite element methods (see Section 5.7), since it may be considered a finite difference method as shown in Section 5.6.2, and at the same time involves finding a solution in a spline approximation space.

A natural approach is to try to compute the smooth solution $u_\pi(x)$ of (5.128) directly (and not only solution and derivative values $\mathbf{y}_i$ at mesh points). Thus, we must choose some basis functions $\phi_1(x), \ldots, \phi_M(x)$. With this approach we can naturally evaluate $u_\pi(x)$ for any $x \in [a,b]$, which is useful for nonlinear iterations or for just plotting the solution. Some popular implementations of collocation have used this approach. We shall see that while it offers some minor advantages over the local representation approach of the previous section, it also has drawbacks (which in our opinion usually outweigh the advantages).

When choosing the functions $\phi_j(x)$, we have to pay attention to considerations of efficiency as well as stability. A choice of spline basis functions which have *local support* (see Sections 2.5.1, 2.5.2) answers both of these concerns, particularly that of efficiency, because the resulting collocation matrix is banded. (Such a choice also enables us to again view the collocation method as a finite difference method.) The Hermite-type basis (well-defined if $k \geq m$) and the B-spline basis, both described in Section 2.5, turn out to have a minimal local support, and we consider them for the case $u_\pi(x) \in \mathbf{P}_{k+m,\pi,m}$, $k \geq m$. We consider only linear BVPs (5.119), since quasilinearization for (5.120) is done precisely as in Section 5.6.2.

We first consider using B-splines to represent $u_\pi(x)$, so let $\phi_j(x) = B_{j,k+m}(x)$, $1 \leq j \leq M := kN + m$. Algorithm 2.89 together with (2.90) describes how $\phi_j(x)$ and its derivatives are evaluated. The collocation matrix $\mathbf{A} \in \mathbf{R}^{M \times M}$ is formed by imposing the kN collocation conditions

$$Lu_\pi(x_{ij}) = q(x_{ij}) \qquad 1 \leq j \leq k, \qquad 1 \leq i \leq N \tag{5.148a}$$

and the BC

$$B_a \mathbf{y}_\pi(a) + B_b \mathbf{y}_\pi(b) = \boldsymbol{\beta} \tag{5.148b}$$

where $\mathbf{y}_\pi(x) := (u_\pi(x), u_\pi'(x), \ldots, u_\pi^{(m-1)}(x))^T$.

Since the multiplicity of an interior mesh point x_i in the knot sequence $\{t_j\}_{j=1}^{m+k+m}$ is k (see Section 2.5.2), m of the $k+m$ nonzero B-splines on (x_i, x_{i+1}) have support (x_{i-1}, x_{i+1}), $k-m$ have support (x_i, x_{i+1}), and m have support (x_i, x_{i+2}). The resulting collocation matrix **A** has the form

$$A = \begin{bmatrix} V_1 & & & \\ & V_2 & & \\ & & \cdot & \\ & & & V_N \\ W_1 & & & W_2 \end{bmatrix} \tag{5.149}$$

where $V_1, \ldots, V_N \in \mathbf{R}^{k \times k+m}$ and $W_1, W_2 \in \mathbf{R}^{m \times (k+m)}$. Each V_i corresponds to the k collocation equations in the ith subinterval (x_i, x_{i+1}) and overlaps with V_{i-1} and V_{i+1} (for $V_1, \ldots, V_N$) in m columns. An example is depicted in Fig. 5.6. As before, the reader should picture the structure of **A** with the 3×5 blocks as indicated in the figure, but with N large.

$$\mathbf{A} = \begin{bmatrix} \times & \times & \times & \times & \times & & & \\ \times & \times & \times & \times & \times & & & \\ \times & \times & \times & \times & \times & & & \\ & & & \times & \times & \times & \times & \times \\ & & & \times & \times & \times & \times & \times \\ & & & \times & \times & \times & \times & \times \\ \times & \times & & & & & \times & \times \\ \times & \times & & & & & \times & \times \end{bmatrix}$$

Figure 5.6 Structure of **A** using B-splines with $m = 2$, $k = 3$ and $N = 2$

If the BC are separated, the equations should be reordered so that those involving only information at $x = a$ are listed first, and then **A** is banded.

A second choice is to use a Hermite-type basis instead of a B-splines basis to represent $u_\pi(x)$. Then $\phi_j(x)$, $1 \le j \le M$, coincide with the functions described in Section 2.5.1. It is easy to see that these basis functions have the same support as the B-splines do, so that the resulting collocation matrix **A** again has the structure (5.149).

The numerical method used to solve the resulting linear system of collocation equations should utilize the sparseness structure of **A** as much as possible in order to provide economy both in terms of storage and in terms of computational effort. This issue is discussed in Chapter 7.

We have seen by now three implementations of the collocation method for a higher-order ODE. The first, presented in Section 5.6.2 and based on local solution representation, is markedly different from the other two, just introduced. A comparison of these implementations would have to take into account considerations of

conditioning, computational effort, storage, and ease of implementation. As it turns out, the Hermite-type basis is simpler and cheaper to implement than the B-spline basis, and they both have essentially the same conditioning, as discussed in Section 5.6.4. We later comment on the (more intricate) comparison between the local solution representation and the Hermite-type basis. But it should be remarked that all three approaches produce reasonable, practical implementations.

Let us consider in some detail how the collocation matrix **A** is constructed in the case of a Hermite-type basis. From (2.83), the derivatives of the basis functions $\{\phi_j(x)\}_{j=1}^{M}$ involve subinterval-dependent factors times the terms $\{\psi_j^{(l-1)}(\rho_r)$: $1 \le r \le k-m, 1 \le l \le m+1, 1 \le j \le k\}$. These latter terms are mesh-independent and need only to be evaluated once (which is why the work required to construct the collocation matrix **A** is considerably less than for the B-splines). The approximate solution $u_\pi(x)$ of (5.128) and its derivatives are evaluated at collocation points by

$$u_\pi^{(l-1)}(x_{ir}) = \sum_{j=1}^{k+m} \alpha_{(i-1)k+j}\, \phi_{(i-1)k+j}^{(l-1)}(x_{ir}) \qquad 1 \le r \le k, \quad 1 \le l \le m+1, \quad 1 \le i \le N$$

$$= \sum_{j=1}^{k+m} \alpha_{(i-1)k+j}\, w_{ij}\, \psi_j^{(l-1)}(\rho_r)/h_i^{l-1}$$

where from (2.83)

$$w_{ij} = \begin{cases} \left(\dfrac{h_{i-1}}{h_i}\right)^{j-1} & \text{if } 1 \le j \le m \quad \text{and} \quad 2 \le i \le N \\ 1 & \text{otherwise} \end{cases} \tag{5.150}$$

The block $V_i = (V_{rj}^i)$ for **A** corresponding to the k collocation conditions in $[x_i, x_{i+1}]$ has the form

$$V_{rj}^i = w_{ij}\{h_i^{-m}\psi_j^{(m)}(\rho_r) - \sum_{l=1}^{m} c_l(x_{ir}) h_i^{1-l}\psi_j^{(l-1)}(\rho_r)\}$$

From (5.150)

$$w_{ij}\alpha_{(i-1)k+j} = h_i^{j-1} y_{ij} \qquad 1 \le j \le m, \qquad 1 \le i \le N$$

where y_{ij} are solution and derivative values at mesh points, as before. Thus, the (scaled) coefficients $\{\alpha_{(i-1)k+j}\}_{j=1;i=1}^{m;N+1}$ correspond to the unknowns $\{\mathbf{y}_i\}_{i=1}^{N+1}$ of the local representation (5.132). The other unknown coefficients correspond to basis functions which are nonzero over only one subinterval and can therefore be locally eliminated. Thus, $k-m$ Gaussian elimination steps (with row partial pivoting) per subinterval i can be carried out in order to locally eliminate some of the unknowns, leaving a system of equations of the form (5.17) for unknowns directly related to nodal solution values. This latter process is referred to as *parameter condensation*. Note that the Hermite-type basis implementation with condensation resembles (but is not the same as) the process in Section 5.6.2, where also some unknowns $\mathbf{z}_i$ are locally eliminated to produce the difference scheme (5.137a), yielding a linear system of equations with the same size and structure.

After the nodal solution values are obtained (when condensation is used), we may need to evaluate $u_\pi(x)$ in (x_i, x_{i+1}), particularly if the BVP is nonlinear. This can be done by (i) forming a local polynomial approximation by interpolating the nodal values, or (ii) recovering the eliminated unknowns by local back substitution for each subinterval and using (5.128). The first of these methods has the advantages that it is more efficient, both in terms of speed and in terms of storage (the upper triangular matrices resulting from the parameter condensation may be discarded), and that it uses only superconvergent nodal values [see Theorem 5.140]. This method applies equally well here and for the local solution representation (5.132). Its disadvantage is that it is possibly less robust in a general-purpose implementation. The second method above requires the same amount of effort as applying the usual Gaussian elimination to **A**; it simply does things in a different order.

Condensation of parameters could also be used in the B-spline case, whereby unknowns corresponding to B-splines having support over only one subinterval are eliminated. The nodal values $\mathbf{y}_i$ can be determined from the remaining unknowns (see Exercise 22). Still, condensation of parameters is not as natural in this case because these nodal values are not determined directly.

The use of condensation of parameters also allows us to examine relative merits of the Hermite-type basis and the local solution representation of Section 5.6.2. (The B-spline implementation is a somewhat distant third.) Whereas $k-m$ local elimination steps per subinterval are made here, k steps are needed in Section 5.6.2. Thus, if the first method above is used for solution evaluation, then the Hermite-type basis has a storage advantage. It is also cheaper in operation count, but only by a slight amount. The ease of carrying out the two implementations is comparable. But the local solution representation has a significant conditioning advantage, which in our opinion generally outweighs the slight disadvantages when compared to the Hermite-type basis. This is discussed next.

5.6.4 Conditioning of collocation matrices

We are considering the BVP (5.119) for an mth order ODE on a finite interval, assuming that as a first-order system it has a conditioning constant κ of moderate size. Using collocation as described above, it turns out that both the Hermite-type basis and the B-spline basis yield linear systems of equations with condition numbers which, as functions of the mesh π, are less than optimal. An indication of this is obtained if we consider $L\phi_j(x)$ for *any* (scaled) basis function of $\mathbf{P}_{k+m,\pi,m}$ with compact support on a uniform mesh π. When h is small enough, we obtain

$$L\phi_j(x) \sim \phi_j^{(m)}(x) \sim h^{-m}$$

and this factor h^{-m} appears in **A** when imposing the collocation equations (5.148a). It is not difficult to see that this yields

$$\text{cond } (\mathbf{A}) \sim h^{-m}$$

at best. (N.B. $N \sim h^{-1}$.) In general, when the mesh π is not necessarily uniform, the ith block of **A** corresponding to the collocation equations on the ith subinterval contains the factor h_i^{-m}. It makes sense to attempt to row-scale blocks of **A** differently, because they involve different scaling factors, and because the BC equations involve different

derivatives of $u_\pi(x)$ than the collocation equations. If we do the equilibration so that the largest element in each row has magnitude 1, and call the scaled matrix $\hat{\mathbf{A}}$, then it can be shown that the Hermite-type basis (and, under stronger assumptions, the B-spline basis) yields

$$\text{cond}(\hat{\mathbf{A}}) \sim \max_{1\le i\le N+1} \sum_{j=1}^{N} \sum_{l=0}^{m-1} h_j^{-m} \hat{h}_i^{\,l} \int_{x_j}^{x_{j+1}} |G_l(x_i, t)|\, dt \tag{5.151}$$

where $\hat{h}_i := \max(h_i, h_{i+1})$ and as before $G(x,t)$ is the (scalar) Green's function of (5.119) and $G_l(x,t) \equiv \dfrac{\partial^l}{\partial x^l} G(x,t)$. It can also be shown that the parameter condensation procedure does not affect the conditioning of the algorithm significantly, so (5.151) is satisfied for this variant too. From (5.151) we can write

$$\text{cond}\ (\hat{\mathbf{A}}) \lesssim \kappa N \underline{h}^{-m+1} \tag{5.152}$$

where

$$\underline{h} := \min_{1\le i\le N} h_i$$

The approximate bound (5.152) is sharp when $\kappa \approx \|G\|$, and when there are $O(N)$ subintervals of size $h_i \sim \underline{h}$.

This is to be contrasted with the $\sim\kappa N$ condition number obtained with the local solution representation of Section 5.6.2 ! We emphasize that we are discussing different *implementations* of the *same* collocation method. Thus, the various implementations would all yield the same approximate solution $u_\pi(x)$ if exact arithmetic were used. However, as the condition numbers indicate, the roundoff error effects for the implementations considered in Section 5.6.3 differ considerably from those of Section 5.6.2. This is demonstrated in the following example.

Example 5.18

Consider the simple BVP

$$u'' - 4u = 16x + 12x^2 - 4x^4, \qquad 0 < x < 1$$

$$u(0) = u'(1) = 0$$

which has the unique solution

$$u(x) = x^4 - 4x.$$

We solve this problem using collocation at Gaussian points in $\mathbf{P}_{6,\pi,2}$. Since the exact solution lies in the approximation space, there is no discretization error, and we can examine pure roundoff error effects. The spline solution is represented using (i) a B-spline basis, (ii) a Hermite-type basis, and (iii) a Runge-Kutta local representation. For uniform meshes, doubling N should quadruple the condition number for (i) and (ii), but only double it for (iii), according to the above estimates. To demonstrate the effect of highly nonuniform meshes in (5.151), (5.152), the solution is also computed on

$$\pi_1\colon\ 0 < 10^{-4} < .25 < .5 < .75 < 1$$

$$\pi_2: 0 < 10^{-6} < .25 < .5 < .75 < 1$$

$$\pi_3: 0 < .25 < .5 < .75 < 1-10^{-4} < 1$$

$$\pi_4: 0 < .25 < .5 < .75 < 1-10^{-6} < 1$$

Since the Green's function for this example is

$$G(x,t) = \begin{cases} \sinh 2x \cosh 2(1-t)/(2 \cosh 2), & x \le t \\ \sinh 2t \cosh 2(1-x)/(2 \cosh 2), & x > t \end{cases}$$

we have that for the first two meshes $G(x,t) \sim h$ for $x \in [0, x_2]$, so the expression in (5.151) (with $m=2$) yields a condition number estimate of $\sim N$. Thus, the condition number should be roughly the same for π_1 and π_2, if all three implementations are used. On the other hand, $G(1,t)$ is not small, and since there is only one very small subinterval near $x=1$ in the last two meshes, the condition number should increase by a factor of ~100 when going from π_3 to π_4, using implementations (i) or (ii). No such deterioration is expected with (iii).

The calculated condition numbers and solution errors, when floating-point arithmetic with 56 binary digits in the mantissa is used, are listed in Table 5.9. Note that a fairly long wordlength has been used. If a short wordlength is used, the loss of precision from using the first two implementations (namely those of Section 5.6.3) may be significant. More severe effects for implementations (i) and (ii) [but not (iii)] can also occur if higher-order ODEs are encountered.

TABLE 5.9 Condition numbers and errors for Example 5.18

		Condition numbers			Errors in u		
	N	(i)	(ii)	(iii)	(i)	(ii)	(iii)
Uniform meshes	10	.42 + 3	.22 + 3	.20 + 2	.50 − 13	.21 − 13	.24 − 14
	20	.17 + 4	.87 + 3	.34 + 2	.26 − 12	.11 − 12	.33 − 14
	40	.69 + 4	.34 + 4	.64 + 2	.23 − 11	.72 − 13	.82 − 14
	80	.28 + 5	.14 + 5	.12 + 3	.68 − 11	.18 − 11	.13 − 13
Nonuniform meshes	π_1	.68 + 2	.43 + 2	.17 + 2	.10 − 13	.62 − 14	.67 − 15
	π_2	.68 + 2	.43 + 2	.12 + 2	.40 − 14	.58 − 14	.18 − 14
	π_3	.96 + 5	.54 + 5	.12 + 2	.41 − 10	.94 − 11	.18 − 14
	π_4	.96 + 7	.54 + 7	.12 + 2	.38 − 8	.15 − 8	.18 − 14

□

It is important to realize that truncation error usually dominates roundoff error when one is solving BVPs using collocation, regardless of the choice of implementation from amongst the ones introduced in this chapter. (Note also that it is extremely rare to find an ODE of order higher than 4 in practice.) Nevertheless, it is not unreasonable to expect the collocation method to perform well even if the mesh is extremely fine or highly nonuniform, and in such situations the local solution representation is considerably more secure. The issue becomes more important if use of a short wordlength is contemplated.

The different conditioning occurring with the various collocation implementations raises more global theoretical issues, particularly if we recall from Section 5.6.2 that the local representation implementation can be considered as a sophisticated discretization of the corresponding first-order ODE. Note that in Section 5.6.1 we also have essentially the same discretization for a second-order ODE and for a corresponding first-order system [viz. (5.126) and (5.124), respectively], and there too the condition numbers differ: that of (5.126) for a uniform mesh is $\sim N^2$, while that of (5.124) is only $\sim N$. Still, we would expect the *stability constant* of the algorithm not to change.

Indeed, recall the discussion in Section 2.8.2 on the fundamental difference between the "smooth" truncation error and the "rough" ("high frequency") roundoff error. The truncation error, controlled by the stability constant, is the same for the three collocation implementations [or for (5.124) and (5.126)]. It satisfies a discrete analogue of the smoothing property of the inverse of the differential operator (cf. Section 5.5.4), and it is bounded (in fact, vanishes) as $h \to 0$.

Not so for roundoff error. Consider a stable finite difference or collocation discretization of a BVP involving a first-order ODE system, and denote for simplicity the differential operator together with the boundary operator by $\mathbf{L}$ [cf. (2.167), (2.168)]. We have a system of linear algebraic equations

$$\mathbf{Au} = \mathbf{v} \tag{5.153}$$

where $\mathbf{A}$ is a discretization of $\mathbf{L}$, $\mathbf{u}$ relates to solution values of $\mathbf{y}(x)$ at mesh points, say, and $\mathbf{v}$ relates to inhomogeneity values of $\mathbf{q}(x)$ and $\boldsymbol{\beta}$. The stability of the scheme yields that $\| \mathbf{A}^{-1} \| \le K$, where the constant K is of moderate size because κ is. The qualitative size of the condition number of $\mathbf{A}$ is then essentially $\| \mathbf{A} \|$. When assessing the effect of *roundoff errors*, the vectors $\mathbf{u}$ and $\mathbf{v}$ in (5.153) must be considered as belonging to the same normed space, so the smoothing effect of $\mathbf{L}^{-1}$ is in this sense lost. Correspondingly, $\mathbf{A}$ is an approximation of *an unbounded differential operator* $\mathbf{L}$. Indeed, reasonable one-step schemes yield matrices which can be equilibrated, as we have seen, to have condition numbers

$$\text{cond}\,(\mathbf{A}) \sim \kappa N$$

where $N \to \infty$ as $h \to 0$. In practice values of κN are quite tolerable for well-conditioned BVPs (i. e., with κ not large), but as we ask for a diminishing truncation error for a BVP, the roundoff error generally grows unboundedly (regardless of the numerical method and implementation used).

Now, consider the higher-order differential operator L. Its inverse L^{-1} is bounded, in an appropriate norm, by the conditioning constant κ of its equivalent first-order system form. (Let us assume for simplicity that $\kappa = 1$.) This says nothing about roundoff error accumulation resulting from (the discretization of) the unbounded operator L. If L is discretized directly, by evaluating $L\phi_j(x)$ at collocation points [as in (5.148), (5.128)], then a numerical differentiation of an mth order derivative is encountered, leading to the estimate (5.151). This condition number *cannot be improved* by any method of implementation which forms basis functions $\phi_j(x)$ with compact support and then differentiates them m times.

Now we can better understand also the conditioning of the local (or Runge-Kutta) representation of the collocation solution. Here the unfavourable condition number (5.152) (for $m > 1$) is avoided by never constructing basis functions $\phi_j(x)$ explicitly. Instead, we can view the implementation as starting from the mth *derivative* of the approximate solution (which is generally discontinuous at mesh points), representing it on each mesh subinterval i as a linear combination

$$u_\pi^{(m)}(x) = \sum_{j=1}^{k} z_{ij} \psi_j^{(m)}\left(\frac{x - x_i}{h_i}\right), \qquad x_i \le x \le x_i + 1 \tag{5.154}$$

where $\psi_j^{(m)}(t)$ are k linearly independent polynomials of order k (k being the number of collocation points per subinterval). Then $u_\pi(x)$ and its derivatives are formed by *integration* of (5.154), thus avoiding a high-order numerical differentiation. The k unknowns $\mathbf{z}_i$ are local in each subinterval i and can be locally eliminated upon imposing the k collocation equations, thereby eliminating $u_\pi^{(m)}$. The inevitable factor N in the condition number appears when we patch the solution segments together to form a global solution $u_\pi(x) \in C^{(m-1)}[0, 1]$. [When we attempt to simulate this argument for the Hermite-type basis of Section 5.6.3, the representation corresponding to (5.154) has more than k parameters, which are therefore not completely local to the ith subinterval alone, and the argument does not work.]

5.7 FINITE ELEMENT METHODS

Finite element methods are among the most extensively studied methods for solving differential equations. They have enjoyed widespread use in solving BVPs in partial differential equations (PDEs), originally steady state problems in structural engineering. While such problems can be described as BVPs, they also satisfy an equivalent variational formulation. Specifically, an integral relationship in the unknown function variable (a *functional)* is optimized over an appropriate class of functions by the solution of the BVP.

The variational formulation incorporates some of the BC information and involves lower derivatives than the BVP does. Although not so critical for ODEs, both features are important for PDEs because lower derivatives are much easier to handle and because satisfying the BC for irregularly shaped regions can pose difficulty. More importantly for the latter case, the variational formulation allows geometrically flexible discretizations of domains, e. g., using triangles (in two independent variables) which do not necessarily lie along rectangular grid lines. Again, this feature is dormant in the ODE case.

The basic idea of the classical *Ritz method* for elliptic problems is to take a convenient space of functions which approximates the space where the exact solution lies and to choose the approximate solution such that it minimizes the functional over this approximation space of functions. For the finite element method, these functions are normally chosen as piecewise polynomials (splines) defined on some partition of the domain in the independent variable(s). Basis functions with local support for the piecewise polynomial space (which give rise to the name "finite element") yield sparse matrix equations for the unknown parameters. Moreover, the matrices are symmetric

positive definite, which allows an easier application of sparse matrix techniques for their solution.

For ODEs the task of solving the linear system is much easier and therefore not as important as it is for PDEs. Furthermore, special geometrical flexibility is not needed and, as we have seen in the last section, construction of a high-order spline basis is more or less straightforward. Indeed, the collocation method turns out to usually be more efficient than the traditional finite element methods. Collocation also applies to quite general BVPs for which no variational formulation is available. As a result, these traditional finite element methods which are so important for solving PDEs do not play a correspondingly central role in solving ODEs, except in special situations, and we only mention them briefly.

Some may say that for a method to be called a finite element method it should be derived by using a variational formulation. Others take the broader interpretation and call any method which gives a spline solution a finite element method, so technically a (spline) collocation method is one.

5.7.1 The Ritz method

The Ritz method operates directly on the variational formulation of a BVP. We shall just motivate the method through the simple second-order problem

$$Lu(x) \equiv -u''(x) + r(x)u(x) = q(x) \qquad a < x < b \tag{5.155a}$$

$$u(a) = 0, \qquad u(b) = 0 \tag{5.155b}$$

Here $r(x)$ is nonnegative and (5.155a, b) has the solution $u(x)$. Defining the functional

$$I(v) := \frac{1}{2} \int_a^b [(v'(x))^2 + r(x)(v(x))^2 - 2v(x)q(x)]\, dx \tag{5.156}$$

for any $v(x) \in C^{(1)}[a, b]$, a direct substitution of $d(x) = v(x) - u(x)$ into (5.156) gives

$$I(v) = I(u) + \int_a^b [u'(x)\, d'(x) + r(x)u(x)\, d(x) - d(x)q(x)]\, dx \tag{5.157}$$

$$+ \frac{1}{2} \int_a^b [(d'(x))^2 + r(x)(d(x))^2]\, dx$$

Using integration by parts and (5.155a, b) (see Exercise 24) one can show that the first integral in (5.157) is equal to zero if $v(a) = v(b) = 0$. Thus, $I(v) \geq I(u)$, and the variational formulation for (5.155a, b) is

$$I(u) = \min_{v \in C_0^1[a,b]} I(v) \tag{5.158}$$

where $C_0^1[a, b] := \{v \in C^{(1)}[a, b] : v(a) = v(b) = 0\}$. Given a mesh π and a spline space of piecewise polynomials (of order k and in $C^{(l-1)}[a, b]$) which are also zero at the end points, $\mathbf{P}^0_{k,\pi,l} := \{s \in \mathbf{P}_{k,\pi,l} : s(a) = s(b) = 0\}$, the corresponding *Ritz solution* $u_\pi(x)$ is defined to satisfy

$$I(u_\pi) = \min_{s \in \mathbf{P}^0_{k,\pi,l}} I(s) \tag{5.159}$$

Motivated by the BVP (5.155), one would choose $k \geq 3$, but the Ritz solution is defined, following (5.158), even if $u_\pi(x)$ is only linear and piecewise continuous (so $k = 2$). The fact that a Ritz solution may have global continuity and order which are too low for the BVP to make sense leads to the necessity of talking about *weak*, or *generalized solutions*, but this is outside the scope of our brief description.

To solve (5.159), we would first choose a basis $\{\phi_j(x)\}_{j=1}^J$ for $\mathbf{P}^0_{k,\pi,l}$ (e. g., B-splines appropriately modified at the ends so as to satisfy the BC, or a local representation), and write, say, $v(x) = \sum_{j=1}^{J} \alpha_j \phi_j(x)$ for any $v \in \mathbf{P}^0_{k,\pi,l}$. The necessary first-order conditions for $u_\pi(x) = \sum_{j=1}^{J} \hat{\alpha}_j \phi_j(x)$ to be the minimizer of (5.159) are

$$\frac{\partial I(v)}{\partial \alpha_j} = 0 \qquad 1 \leq j \leq N+1$$

The solution $\hat{\boldsymbol{\alpha}} = (\hat{\alpha}_1, \ldots, \hat{\alpha}_J)^T$ can be found by solving

$$A\,\hat{\boldsymbol{\alpha}} = \hat{\mathbf{q}} \tag{5.160}$$

where the elements of A and $\hat{\mathbf{q}}$ are

$$a_{ij} = \int_a^b [\phi_i'(x)\phi_j'(x) + r(x)\phi_i(x)\phi_j(x)]\,dx \tag{5.161a}$$

$$=: E(\phi_i, \phi_j) \qquad 1 \leq i, j \leq J$$

and

$$\hat{q}_i = \int_a^b q(x)\phi_i(x)\,dx \tag{5.161b}$$

Notice that determining these coefficients requires evaluating integrals, and the most efficient way to do this is by using a quadrature method (unless the coefficients are exceedingly simple). Observe also that A is symmetric.

Example 5.19

Taking $k = 2$, and $l = 1$, we find that the B-spline basis is simply the "roof functions"

$$\phi_j(x) = \begin{cases} \dfrac{x - x_j}{h_j} & \text{if } x_j \leq x \leq x_{j+1} \\ \dfrac{x_{j+2} - x}{h_{j+1}} & \text{if } x_{j+1} \leq x \leq x_{j+2} \\ 0 & \text{otherwise} \end{cases}$$

and $\hat{\alpha}_j$ is $u_\pi(x_{j+1})$ $(1 \leq j \leq J := N - 1)$. From (5.161a, b),

$$a_{ij} = \begin{cases} \dfrac{1}{h_i} + \dfrac{1}{h_{i+1}} + \displaystyle\int_{x_i}^{x_{i+1}} r(x)\left(\frac{x-x_i}{h_i}\right)^2 dx + \int_{x_{i+1}}^{x_{i+2}} r(x)\left(\frac{x_{i+2}-x}{h_{i+1}}\right)^2 dx & \text{if } j = i \\ -\dfrac{1}{h_i} + \displaystyle\int_{x_i}^{x_{i+1}} r(x)\left(\frac{x_{i+1}-x}{h_i}\right)\left(\frac{x-x_i}{h_i}\right) dx & \text{if } j = i-1 \\ 0 & \text{if } j < i-1 \end{cases}$$

$a_{ij} = a_{ji}$ for $j > i$, and

$$\hat{q}_i = \int_{x_i}^{x_{i+1}} q(x)\left(\frac{x-x_i}{h_i}\right) dx + \int_{x_{i+1}}^{x_{i+2}} q(x)\left(\frac{x_{i+2}-x}{h_{i+1}}\right) dx$$

The matrix A is tridiagonal and symmetric positive definite. If the trapezoidal rule is used to approximate the integrals in its coefficients then the resulting matrix A is *identical* to the one resulting from using the simple finite difference method (5.4), (5.5). But note that the Ritz method has been defined on an arbitrary mesh and is essentially no more difficult to apply on a nonuniform mesh than on a uniform one. □

Assuming appropriate smoothness on the coefficients of the ODE (5.155a) [and recalling that $r(x) \geq 0$], we can show for instance

Theorem 5.162 For any integer $n \geq 1$, if $l \geq n$ then the Ritz solution $u_\pi(x) \in \mathbf{P}^0_{2n,\pi,l}$ satisfies

$$\| u_\pi - u \|_2 := \left[\int_a^b (u_\pi(x) - u(x))^2 dx\right]^{1/2} = O(h^{2n}) \tag{5.162}$$

Proof outline: For arbitrary $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_j)$, $v(x) = \sum_{j=1}^{J} \alpha_j \phi_j(x) \in \mathbf{P}^0_{2n,\pi,l}$ satisfies

$$\boldsymbol{\alpha}^T A \boldsymbol{\alpha} = \int_a^b [(v'(x))^2 + r(x)(v(x))^2]\, dx =: \| v \|_A \geq 0$$

Since $\| v \|_A = 0$ iff $v \equiv 0$, A is positive definite, so $u_\pi(x)$ is uniquely defined from (5.160). From (5.157),

$$\| u_\pi - u \|_A = \min_{v \in \mathbf{P}^0_{2n,\pi,l}} \| v - u \|_A$$

and since spline approximation theory guarantees the existence of a $v_\pi \in \mathbf{P}^0_{2n,\pi,l}$ satisfying

$$\| v_\pi - u \|_2 = O(h^{2n})$$

we also have

$$\| v_\pi - u \|_A = O(h^{2n-1})$$

and hence

$$\| u_\pi - u \|_A = O(h^{2n-1}) \tag{5.163}$$

To improve the error estimate (5.163), we use the so-called "Nitsche trick": Let $w(x)$ solve the BVP

$$-w''(x) + r(x)w(x) = (u_\pi(x) - u(x))/\| u_\pi - u \|_2$$

$$w(a) = w(b) = 0$$

Multiplying this ODE by $u_\pi(x) - u(x)$ and integrating, we have

$$\| u_\pi - u \|_2 = \int_a^b (-w''(x) + r(x)w(x))(u_\pi(x) - u(x))\, dx$$

and use of integration by parts and (5.157) show that

$$\| u_\pi - u \|_2 = E(w, u_\pi - u) = E(w - v, u_\pi - u)$$

for any $v \in \mathbf{P}^0_{2n,\pi,l}$. [The quadratic form E is defined in (5.161a).] Choosing for v an appropriate piecewise linear interpolate $\hat{w}$ to w, the Cauchy-Schwarz inequality gives

$$\| u_\pi - u \|_2 \le \| w - \hat{w} \|_A \| u_\pi - u \|_A = O(h^{2n})$$

□

A linear differential operator L has an *adjoint operator* L^* such that

$$\int_a^b (Lv(x))w(x)\, dx = \int_a^b v(x)(L^* w(x))\, dx$$

for any smooth $v(x)$ and $w(x)$, and L is called *self-adjoint* if $L = L^*$. In the latter case the Green's function $G(x,t)$ is symmetric, satisfying $G(x,t) = G(t,x)$ for $a \le x, t \le b$ [cf. Theorem 3.40]. A BVP involving a self-adjoint operator and appropriate BC has an equivalent variational formulation. Generalizing the above, the BVP

$$Lu(x) \equiv \sum_{j=0}^{m/2} (-1)^j (c_{2j}(x)u^{(j)}(x))^{(j)} = q(x) \qquad a < x < b \tag{5.164a}$$

$$u^{(j)}(a) = u^{(j)}(b) = 0 \qquad 0 \le j \le m/2 - 1 \tag{5.164b}$$

where m is an even integer, $c_{2j}(x) \ge 0$ $(0 \le j \le m/2 - 1)$, and $c_m(x) > 0$, has an equivalent variational formulation with

$$I(v) := \int_a^b [\sum_{j=0}^{m/2} c_{2j}(x)(v^{(j)}(x))^2 - 2q(x)v(x)]\, dx \tag{5.165}$$

It is straightforward to show that if $u_\pi(x)$ is the Ritz solution for this problem in $\mathbf{P}^0_{k,\pi,l}$, the subspace of $\mathbf{P}_{k,\pi,l}$ consisting of splines satisfying (5.164b) (for appropriate k and l), and if $\{\phi_j(x)\}_{j=1}^J$ is a basis for $\mathbf{P}^0_{k,\pi,l}$, then

$$\int_a^b [Lu_\pi(x) - q(x)]\phi_j(x)\, dx = 0 \qquad 1 \le j \le J \tag{5.166}$$

5.7.2 Other finite element methods

For a BVP consisting of a smooth but otherwise arbitrary linear differential operator L and homogeneous BC, the *Galerkin solution* $v_\pi(x) = \sum_{j=1}^{J} \beta_j \phi_j(x)$ in $\mathbf{P}^0_{k,\pi,l}$ is defined by requiring that

$$\int_a^b [Lv_\pi(x) - q(x)]\phi_j(x)\,dx = 0 \qquad 1 \le j \le J \tag{5.167}$$

Integration by parts may be used, repeatedly if necessary, to lower the highest derivative appearing in (5.167) and thus relax the minimum smoothness requirement on the approximation space. If L is self-adjoint then the Galerkin solution is the same as the Ritz solution, but the Galerkin solution may be defined for a linear BVP even if there is no corresponding variational formulation. The coefficients $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_J)^T$ for $v_\pi(x)$ are determined by solving the linear system of equations resulting from (5.167). Usually the matrix coefficients $\int_a^b (L\phi_i(x))\phi_j(x)\,dx$ cannot be computed exactly, so a quadrature rule is used to approximate them. The solution to the resulting linear system is called a *discrete Galerkin solution.*

Instead of (5.167), we can solve for $w_\pi(x) \in \mathbf{P}^0_{k,\pi,l}$ defined by

$$\int_a^b [Lw_\pi(x) - q(x)]L\phi_j(x)\,dx = 0 \qquad 1 \le j \le J \tag{5.168}$$

Both (5.167) and (5.168) may be viewed as *orthogonality conditions.* The solution $w_\pi(x)$ is called the *least squares solution* because (5.168) is equivalent to solving

$$\min_{v \in \mathbf{P}^0_{k,\pi,l}} \int_a^b [Lv(x) - q(x)]^2\,dx$$

As with the Galerkin method, in practice one must generally compute a *discrete least squares solution.* Interestingly, the collocation solution $u_\pi(x) \in \mathbf{P}^0_{k,\pi,l}$ is related naturally to the Galerkin and least squares solutions, where instead of (5.167) or (5.168) we require that the residual $Lu_\pi(x) - q(x)$ satisfy

$$\int_a^b [Lu_\pi(x) - q(x)]\psi_j(x)\,dx = 0 \qquad 1 \le j \le J \tag{5.169}$$

with $\psi_j(x)$ being a δ–function corresponding to a collocation point z_j [so $\int_a^b f(x)\psi_j(x)dx = f(z_j)$]. Using other choices of $\{\psi_j(x)\}$ from *different* spline spaces $\hat{\mathbf{P}}_{\hat{k},\pi,\hat{l}}$ defines a general class of methods called *Petrov-Galerkin methods.* Note that if integration by parts cannot be (or is not) used in (5.169) or (5.167) then the minimum smoothness and order requirements on the approximate solution space are not different from those for the least squares method (5.168), the collocation method of Section 5.6, or other difference schemes.

EXERCISES

1. Consider the approximation (5.4) to the problem (5.2).

(a) Show that if $h \leq 2/\|p\|$ and $r \geq 0$ then **A** of (5.5) is diagonally dominant.

(b) Show that if in addition, $r(x) \geq \delta > 0$ for $x \in [0, 1]$, then (5.9) is satisfied with $K = \max\{1, \delta^{-1}\}$

(c) Show that

$$\text{cond } (\mathbf{A}) \approx 4(1 + \frac{1}{\delta h^2}) = O(N^2)$$

2. Describe the application of a finite difference scheme like (5.4) to a nonlinear problem

$$y'' = f(x, y, y')$$

subject to the BC (5.2b). How would the scheme look for Example 3.1?

3. Consider the BVP (5.13a) subject to separated BC

$$B_{a1}\mathbf{y}(a) = \boldsymbol{\beta}_1, \qquad B_{b2}\mathbf{y}(b) = \boldsymbol{\beta}_2$$

with $B_{a1} \in \mathbf{R}^{(n-\nu)\times n}$ and $B_{b2} \in \mathbf{R}^{\nu\times n}$. Show that the linear system of equations (5.17) can be rearranged so that **A** is banded with bandwidth $3n - 1$. What would the bandwidth be after Gauss elimination with partial row-pivoting is applied to **A**?

4. Implement the trapezoidal-Newton algorithm described in Section 5.1.3 for nonlinear first-order BVPs. Your data structures will depend on the linear system solver used. For the latter, use a band solver (Exercise 3), or consult Chapter 7, or treat **A** as full (as a last resort). Try your code on Example 5.3. Devise initial guesses which would yield approximations to both exact solutions.

5. Show that both the midpoint scheme and the trapezoidal scheme for the BVP (5.20) are consistent of order 2.

6. Consider the ODE (5.2a) under the BC

$$u(0) = \beta_1, \qquad u'(1) = \beta_2$$

Derive appropriate difference approximations for the BC so that, when used with (5.4a), the resulting method is consistent of order 2.

7. Prove Theorem 5.56. (*Hint:* Verify the premises of the Newton-Kantorovich Theorem 2.64. Note that α is bounded by stability, while β can be made small for h and ρ_0 small).

8. Show that the trapezoidal scheme (5.21) is an implicit Runge-Kutta method (5.59), with $k = 2$, $\rho_1 = 0$, $\rho_2 = 1$, $\alpha_{11} = \alpha_{12} = 0$, $\beta_1 = \beta_2 = \alpha_{21} = \alpha_{22} = 1/2$ and that the precision is $p = 2$.

9. Show that (5.76) yields (5.77).

10. **(a)** Show that the collocation scheme at $k = 3$ Lobatto points per subinterval, derived in Example 5.6, can be written in the one-step form

$$\frac{\mathbf{y}_{i+1}-\mathbf{y}_i}{h_i} = \frac{1}{6}\{\mathbf{f}(x_i, \mathbf{y}_i)+\mathbf{f}(x_{i+1}, \mathbf{y}_{i+1}) + 4\mathbf{f}(x_{i+1/2}, \frac{1}{2}(\mathbf{y}_i + \mathbf{y}_{i+1})$$

$$+\frac{h}{8}(\mathbf{f}(x_i, \mathbf{y}_i)-\mathbf{f}(x_{i+1}\mathbf{y}_{i+1})))\}$$

(b) Show that, applied to the nonlinear problem (5.20), this scheme is convergent of order 4.

11. Consider Radau collocation with $k = 2$ for $\mathbf{y}' = \mathbf{f}(x, \mathbf{y})$. The canonical collocation points are given by $\rho_1 = 1/3$, $\rho_2 = 1$.

(a) Derive the method coefficients α_{jl}, β_l.

(b) Obtain the method in a one-step form, as in the previous exercise.

This is a third-order, nonsymmetric scheme with good stability properties for stiff IVPs.

12. Consider a collocation method (5.72) for the linear first-order ODEs $\mathbf{y}' = A(x)\mathbf{y} + \mathbf{q}(x)$ with order $p > k + 1$. Show that the error for $x_i < x < x_{i+1}$ can be written as

$$\mathbf{y}(x) - \mathbf{y}_\pi(x) = \frac{h_i^{k+1}}{k!}\mathbf{y}^{(k)}(x_i) \int_0^{\frac{x-x_i}{h_i}} \prod_{l=1}^{k} (t-\rho_l)\, dt + O(h_i^{k+2}) + O(h^p)$$

This is a rather strong result, explicitly expressing the leading term of the error. [*Hint:* Write $\mathbf{e}_\pi(x) = \mathbf{e}_i + \int_{x_i}^{x} \mathbf{e}_\pi'$ and use (5.67), (5.78).]

15. **(a)** Show that the local truncation error for the trapezoidal scheme can be expanded as in (5.89), (5.90).

(b) Show that (5.89) holds for the midpoint scheme as well. What are the $\mathbf{T}_j$ in this case?

(c) Derive the corresponding series expansion for the fourth-order Simpson scheme given in Example 5.6.

16. Show that if the assumptions of Theorem 5.93 hold, except that the quasiuniformity assumption is relaxed to

$$h_i/h \geq Kh^{(r-k+1)/k}$$

then the trapezoidal solution error can be written as

$$\mathbf{y}(x_i) - \mathbf{y}_i = \sum_{j=1}^{k-1} h_i^{2j}\mathbf{d}_j(x_i) + O(h^{2k}) \qquad 1 \leq i \leq N+1$$

17. Consider the simple finite difference method (5.4).

(a) Derive a series expansion like (5.89) for the local truncation error.

(b) Describe a deferred correction method for this basic discretization scheme.

(c) Write a program implementing a method of order 6, using either extrapolation or deferred correction to accelerate the convergence of (5.4) for the BVP (5.2). Try your program on the problem of Example 5.1 and compare results to those listed in Tables 5.6 and 5.7. [Note that the linear systems of equations (5.5) are of a rather simple, tri-diagonal form.]

18. Show that the local truncation error of the difference scheme (5.137) is $O(h_i^k)$.

19. Prove Theorem 5.147.

20. State and prove a theorem corresponding to Theorem 5.147 for a mixed-order system of ODEs with multipoint BC (1.9a, b).

21. Let us denote the functions ψ_j appearing in (5.132) as ψ_{mj}, bringing out explicitly their dependence on the ODE order m. Show that for the choices (5.133) and (5.138a),

$$\psi_{rj} = \psi_{mj}^{(m-r)}, \qquad 1 \le r \le m, \qquad 1 \le j \le k$$

This does not hold for the choice (5.138b).

22. Construct an algorithm for parameter condensation when using collocation with B-splines, as described in Section 5.6.3. Show that the mesh values $\mathbf{y}_i$ can be retrieved from the B-spline parameters remaining after condensation.

23. Constructing the finite difference method (5.107) as outlined, show that the coefficients $\{\alpha_{ij}\}_{j=1}^{2k}$ satisfy (5.108).

24. For the BVP (5.155a, b), show that the solution $u(x)$ satisfies (5.158), for the functional $I(v)$ defined in (5.156).

25. Show that for any $v \in \mathbf{P}_{k,\pi,I}^0$,

$$E(v, u_\pi - u) = 0$$

with u_π defined by (5.159) and E by (5.161a).