Numerical Methods for Two–Point Boundary Value Problems

Graeme Fairweather and Ian Gladwell

1 Finite Difference Methods

1.1 Introduction

Consider the second order linear two–point boundary value problem

$$Lu(x) \equiv -u'' + p(x)u' + q(x)u = f(x), \quad x \in I, \tag{1.1}$$

$$u(0) = g_0, \quad u(1) = g_1, \tag{1.2}$$

where $I = [0, 1]$. We assume that the functions p, q and f are smooth on I, q is positive, and p^* and q_* are positive constants such that

$$|p(x)| \leq p^*, \ 0 < q_* \leq q(x), \quad x \in I. \tag{1.3}$$

Let $\pi = \{x_j\}_{j=0}^{N+1}$ denote a uniform partition of the interval I such that $x_j = jh, j = 0, 1, \ldots, N+1$, and $(N+1)h = 1$. On this partition, the solution u of (1.1)–(1.2) is approximated by the mesh function $\{u_j\}_{j=0}^{N+1}$ defined by the finite difference equations

$$L_h u_j \equiv -\frac{u_{j+1} - 2u_j + u_{j-1}}{h^2} + p_j \frac{u_{j+1} - u_{j-1}}{2h} + q_j u_j = f_j, \quad j = 1, \ldots, N, \tag{1.4}$$

$$u_0 = g_0, \quad u_{N+1} = g_1, \tag{1.5}$$

where

$$p_j = p(x_j), \quad q_j = q(x_j), \quad f_j = f(x_j).$$

Equations (1.4) are obtained by replacing the derivatives in (1.1) by basic centered difference quotients.

We now show that under certain conditions the difference problem (1.4)–(1.5) has a unique solution $\{u_j\}_{j=0}^{N+1}$, which is second order accurate; that is,

$$|u(x_j) - u_j| = O(h^2), \quad j = 1, \ldots, N.$$

1.2 The Uniqueness of the Difference Approximation

From (1.4), we obtain

$$h^2 L_h u_j = -\left(1+\frac{h}{2}p_j\right)u_{j-1} + \left(2+h^2 q_j\right)u_j - \left(1-\frac{h}{2}p_j\right)u_{j+1} = h^2 f_j, \quad j=1,\ldots,N. \tag{1.6}$$

The totality of difference equations (1.6), subject to (1.5), may be written in the form

$$A\mathbf{u} = \mathbf{b}, \tag{1.7}$$

where

$$\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{N-1} \\ u_N \end{bmatrix}, A = \begin{bmatrix} d_1 & e_1 & & & \\ c_2 & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & e_{N-1} \\ & & & c_N & d_N \end{bmatrix}, \mathbf{b} = h^2 \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_{N-1} \\ f_N \end{bmatrix} - \begin{bmatrix} c_1 g_0 \\ 0 \\ \vdots \\ 0 \\ e_N g_1 \end{bmatrix},$$

and, for $j=1,\ldots,N$,

$$c_j = -\left(1+\frac{1}{2}hp_j\right), \quad d_j = 2+h^2 q_j, \quad e_j = -\left(1-\frac{1}{2}hp_j\right). \tag{1.8}$$

We prove that there is a unique $\{u_j\}_{j=1}^N$ by showing that the tridiagonal matrix A is strictly diagonally dominant and hence nonsingular.

Theorem 1.1. *If $h < 2/p^*$, then the matrix A is strictly diagonally dominant.*

Proof — If $h < 2/p^*$ then

$$|c_j| = 1+\frac{h}{2}p_j, \quad |e_j| = 1-\frac{h}{2}p_j,$$

and

$$|c_j| + |e_j| = 2 < d_j, \quad j=2,\ldots,N-1.$$

Also,

$$|e_1| < d_1, \quad |c_N| < d_N,$$

which completes the proof. □

Corollary If $p=0$, the matrix A is positive definite, with no restriction on the mesh spacing h.

Proof — If $p=0$, then A is strictly diagonally dominant with no restriction on the mesh spacing h. In addition, A is symmetric and has positive diagonal entries and is therefore positive definite. □

1.3 Consistency, Stability and Convergence

To study the accuracy and the computability of the difference approximation $\{u_j\}_{j=0}^{N+1}$, we introduce the concepts of consistency, stability and convergence of finite difference methods. The basic result proved in this section is that, for a consistent method, stability implies convergence.

Definition 1.1 (Consistency)**.** *Let*

$$\tau_{j,\pi}[w] \equiv L_h w(x_j) - Lw(x_j), \quad j = 1, \ldots, N,$$

where w is a smooth function on I. Then the difference problem (1.4)–(1.5) is consistent with the differential problem (1.1)–(1.2) if

$$|\tau_{j,\pi}[w]| \to 0 \text{ as } h \to 0.$$

The quantities $\tau_{j,\pi}[w]$, $j = 1, \ldots, N$, are called the local truncation (or local discretization) errors.

Definition 1.2. *The difference problem (1.4)–(1.5) is locally p^{th}– order accurate if, for sufficiently smooth data, there exists a positive constant C, independent of h, such that*

$$\max_{1 \le j \le N} |\tau_{j,\pi}[w]| \le Ch^p.$$

The following lemma demonstrates that the difference problem (1.4)–(1.5) is consistent with (1.1)–(1.2) and is locally second–order accurate.

Lemma 1.1. *If $w \in C^4(I)$, then*

$$\tau_{j,\pi}[w] = -\frac{h^2}{12}[w^{(4)}(\nu_j) - 2p(x_j)w^{(3)}(\theta_j)],$$

where ν_j and θ_j lie in (x_{j-1}, x_{j+1}).

Proof — By definition

$$\tau_{j,\pi}[w] = -\left[\frac{w(x_{j+1}) - 2w(x_j) + w(x_{j-1})}{h^2} - w''(x_j)\right] + p_j\left[\frac{w(x_{j+1}) - w(x_{j-1})}{2h} - w'(x_j)\right], \; j = 1, \ldots,$$

(1.9)

It is easy to show using Taylor's theorem that

$$(1.10)\quad \frac{w(x_{j+1}) - w(x_{j-1})}{2h} - w'(x_j) = \frac{h^2}{6} w^{(3)}(\theta_j), \quad \theta_j \in (x_{j-1}, x_{j+1}).$$

Also,

$$(1.11)\quad \frac{w(x_{j+1}) - 2w(x_j) + w(x_{j-1})}{h^2} - w''(x_j) = \frac{h^2}{12} w^{(4)}(\nu_j), \quad \nu_j \in (x_{j-1}, x_{j+1}).$$

The desired result now follows on substituting (1.10) and (1.11) in (1.9). □

Definition 1.3 (Stability)**.** *The linear difference operator L_h is stable if, for sufficiently small h, there exists a constant K, independent of h, such that*

$$|v_j| \le K\{\max(|v_0|, |v_{N+1}|) + \max_{1\le i\le N} |L_h v_i|\} \quad j = 0, \ldots, N+1,$$

for any mesh function $\{v_j\}_{j=0}^{N+1}$.

We now prove that, for h sufficiently small, the difference operator L_h of (1.4) is stable.

Theorem 1.2. *If the functions p and q satisfy (1.3), then the difference operator L_h of (1.4) is stable for $h < 2/p^*$, with $K = \max\{1, 1/q_*\}$.*

Proof — If

$$|v_{j*}| = \max_{0\le j\le N+1} |v_j|, \quad 1 \le j* \le N,$$

then, from (1.6), we obtain

$$d_{j*} v_{j*} = -e_{j*} v_{j*+1} - c_{j*} v_{j*-1} + h^2 L_h v_{j*}.$$

Thus,

$$d_{j*} |v_{j*}| \le (|e_{j*}| + |c_{j*}|)\, |v_{j*}| + h^2 \max_{1\le j\le N} |L_h v_j|.$$

If $h < 2/p^*$, then

$$d_{j*} = |e_{j*}| + |c_{j*}| + h^2 q_{j*},$$

and it follows that

$$h^2 q_{j*} |v_{j*}| \le h^2 \max_{1\le j\le N} |L_h v_j|,$$

or

$$|v_{j*}| \leq \frac{1}{q_*} \max_{1\leq j\leq N} |L_h v_j|.$$

Thus, if $\max_{0\leq j\leq N+1} |v_j|$ occurs for $1 \leq j \leq N$ then

$$\max_{0\leq j\leq N+1} |v_j| \leq \frac{1}{q_*} \max_{1\leq j\leq N} |L_h v_j|,$$

and clearly

$$\max_{0\leq j\leq N+1} |v_j| \leq K\{\max(|v_0|, |v_{N+1}|) + \max_{1\leq j\leq N} |L_h v_j|\} \tag{1.12}$$

with $K = \max\{1, 1/q_*\}$. If $\max_{0\leq j\leq N+1} |v_j| = \max\{|v_0|, |v_{N+1}|\}$, then (1.12) follows immediately. □

An immediate consequence of stability is the uniqueness (and hence existence since the problem is linear) of the difference approximation $\{u_j\}_{j=0}^{N+1}$ (which was proved by other means in earlier), for if there were two solutions, their difference $\{v_j\}_{j=0}^{N+1}$, say, would satisfy

$$\begin{aligned} &L_h v_j = 0, \qquad j = 1, \ldots, N, \\ &v_0 = v_{N+1} = 0. \end{aligned}$$

Stability then implies that $v_j = 0$, $j = 0, 1, \ldots, N+1$.

Definition 1.4 (Convergence)**.** *Let u be the solution of the boundary value problem (1.1)–(1.2) and $\{u_j\}_{j=0}^{N+1}$ the difference approximation defined by (1.4)–(1.5). The difference approximation converges to u if*

$$\max_{1\leq j\leq N} |u_j - u(x_j)| \to 0,$$

as $h \to 0$. The difference $u_j - u(x_j)$ is the global truncation (or discretization) error at the point x_j, $j = 1, \ldots, N$.

Definition 1.5. *The difference approximation $\{u_j\}_{j=0}^{N+1}$ is a p^{th}–order approximation to the solution u of (1.1)–(1.2) if, for h sufficiently small, there exists a constant C independent of h, such that*

$$\max_{0\leq j\leq N+1} |u_j - u(x_j)| \leq Ch^p.$$

The basic result connecting consistency, stability and convergence is given in the following theorem.

Theorem 1.3. *Suppose $u \in C^4(I)$ and $h < 2/p^*$. Then the difference solution $\{u_j\}_{j=0}^{N+1}$ of (1.4)–(1.5) is convergent to the solution u of (1.1)–(1.2). Moreover,*

$$\max_{0\le j\le N+1} |u_j - u(x_j)| \le Ch^2.$$

Proof — Under the given conditions, the difference problem (1.4)–(1.5) is consistent with the boundary value problem (1.1)–(1.2) and the operator L_h is stable.

Since

$$L_h[u_j - u(x_j)] = f(x_j) - L_h u(x_j) = Lu(x_j) - L_h u(x_j) = -\tau_{j,\pi}[u],$$

and $u_0 - u(x_0) = u_{N+1} - u(x_{N+1}) = 0$, the stability of L_h implies that

$$|u_j - u(x_j)| \le \frac{1}{q_*} \max_{1\le j\le N} |\tau_{j,\pi}[u]|.$$

The desired result follows from Lemma 1.1. □

It follows from this theorem that $\{u_j\}_{j=0}^{N+1}$ is a second–order approximation to the solution u of (1.1).

1.4 Experimental Determination of the Asymptotic Rate of Convergence

If a norm of the global error is $O(h^p)$ then an estimate of p can be determined in the following way. For ease of exposition, we use a different notation in this section and denote by $\{u_j^h\}$ the difference approximation computed with a mesh length of h. Also let

$$e^h \equiv \|u - u^h\| = \max_j |u(x_j) - u_j^h|.$$

If $e^h = O(h^p)$, then, for h sufficiently small, $h < h_0$ say, there exists a constant C, independent of h, such that

$$e^h \approx Ch^p.$$

If we solve the difference problem with two different mesh lengths h_1 and h_2 such that $h_2 < h_1 < h_0$, then

$$e^{h_1} \approx Ch_1^p$$

and

$$e^{h_2} \approx C h_2^p$$

from which it follows that

$$\ln e^{h_1} \approx p \ln h_1 + \ln C$$

and

$$\ln e^{h_2} \approx p \ln h_2 + \ln C.$$

Therefore an estimate of p can be calculated from

$$p \approx \ln(e^{h_1}/e^{h_2}) / \ln(h_1/h_2) \tag{1.13}$$

In practice, one usually solves the difference problem for a sequence of values of h, $h_0 > h_1 > h_2 > h_3 > \ldots$, and calculates the ratio on the right hand side of (1.13) for successive pairs of values of h. These ratios converge to the value of p as $h \to 0$.

1.5 Higher–Order Finite Difference Approximations

1.5.1 Richardson extrapolation and deferred corrections

Commonly used methods for improving the accuracy of approximate solutions to differential equations are Richardson extrapolation and deferred corrections. These techniques are applicable if there exists an expansion of the local truncation error of the form

$$\tau_{j,\pi}[u] = h^2 \tau[u(x_j)] + O(h^4). \tag{1.14}$$

As is shown in the next theorem, if (1.14) holds, then the stability of the difference operator L_h ensures that there exists a function $e(x)$ such that

$$u(x_j) = u_j + h^2 e(x_j) + O(h^4), \qquad j = 0, 1, \ldots, N+1. \tag{1.15}$$

In Richardson extrapolation, the difference problem (1.4)–(1.5) is solved twice with mesh spacings h and $h/2$ to yield difference solutions $\{u_j^h\}_{j=0}^{N+1}$, and $\{u_j^{h/2}\}_{j=0}^{2(N+1)}$. Then at a point $\hat{x} = jh = (2j)(h/2)$ common to both meshes we have, from (1.15),

$$u(\hat{x}) = u_j^h + h^2 e(\hat{x}) + O(h^4)$$

and

$$u(\hat{x}) = u_{2j}^{h/2} + \frac{h^2}{4} e(\hat{x}) + O(h^4),$$

from which it follows that

$$u(\hat{x}) = u_{2j}^{h/2} + \frac{1}{3}(u_{2j}^{h/2} - u_j^h) + O(h^4).$$

Thus

$$u_j^{(1)} \equiv u_{2j}^{h/2} + \frac{1}{3}(u_{2j}^{h/2} - u_j^h)$$

is a fourth–order approximation to $u(x_j)$, $j = 0, \ldots, N+1$.

In deferred corrections, the difference equations (1.4)–(1.5) are solved in the usual way to obtain $\{u_j\}$. Then a fourth–order difference approximation $\{\hat{u}_j\}_{j=0}^{N+1}$ to $u(x)$ is computed by solving difference equations which are a perturbation of (1.4)–(1.5) expressed in terms of $\{u_j\}$. Suitable definitions of $\{\hat{u}_j\}_{j=0}^{N+1}$ are discussed once we derive (1.15).

Theorem 1.4. *Suppose $u \in C^6(I)$, and $h < 2/p^*$. Then (1.14) and (1.15) hold with $e(x)$ defined as the solution of the boundary value problem*

$$\begin{aligned} Le(x) &= \tau[u(x)], \qquad x \in I, \\ e(0) &= e(1) = 0, \end{aligned} \tag{1.16}$$

where

$$\tau[u(x)] = -\frac{1}{12}[u^{(4)}(x) - 2p(x)u^{(3)}(x)]. \tag{1.17}$$

Proof — Since $u \in C^6(I)$, it is easy to show by extending the argument used in Lemma 1.1 that (1.14) holds with $\tau[u(x)]$ defined by (1.17).

As in the proof of Theorem 1.3, we have

$$L_h[u(x_j) - u_j] = \tau_{j,\pi}[u], \quad j = 1, \ldots, N. \tag{1.18}$$

With (1.14) in (1.18) and using (1.16), we obtain

$$\begin{aligned} L_h[u(x_j) - u_j] &= h^2\tau[u(x_j)] + O(h^4) \\ &= h^2 Le(x_j) + O(h^4) \\ &= h^2\left[Le(x_j) - L_h e(x_j)\right] + h^2 L_h e(x_j) + O(h^4) \\ &= h^2\tau_{j,\pi}[e] + h^2 L_h e(x_j) + O(h^4). \end{aligned} \tag{1.19}$$

From the smoothness properties of the functions p, q, and τ, it follows that the solution $e(x)$ of (1.16) is unique. Moreover, since $\tau \in C^2(I)$, $e(x) \in C^4(I)$ and $\tau_{j,\pi}[e] = O(h^2)$. Using this result in (1.19) and rearranging, we have

$$L_h[u(x_j) - \{u_j + h^2 e(x_j)\}] = O(h^4).$$

The desired result, (1.15), follows from the stability of L_h. □

Deferred corrections is defined in the following way. Suppose $\hat{\tau}_{j,\pi}[\,\cdot\,]$ is a difference operator such that

$$|\tau[u(x_j)] - \hat{\tau}_{j,\pi}[u^h]| = O(h^2), \tag{1.20}$$

where u^h denotes the solution $\{u_j\}_{j=0}^{N+1}$ of (1.4)–(1.5) We define the mesh function $\{\hat{u}_j\}_{j=0}^{N+1}$ by

$$\begin{aligned} &L_h\hat{u}_j = f_j + h^2\hat{\tau}_{j,\pi}[u^h], \quad j = 1,\ldots,N, \\ &\hat{u}_0 = g_0, \quad \hat{u}_{N+1} = g_1. \end{aligned}$$

Then it is easy to show that $\{\hat{u}_j\}_{j=0}^{N+1}$ is a fourth-order approximation to $u(x)$, since

$$\begin{aligned} L_h[\hat{u}_j - u(x_j)] &= f_j + h^2\hat{\tau}_{j,\pi}[u^h] - L_h u(x_j) \\ &= [Lu(x_j) - L_h u(x_j)] + h^2\hat{\tau}_{j,\pi}[u^h] \\ &= -\tau_{j,\pi}[u] + h^2\hat{\tau}_{j,\pi}[u^h] \\ &= -h^2\{\tau[u(x_j)] - \hat{\tau}_{j,\pi}[u^h]\} + O(h^4). \end{aligned}$$

Using (1.20) and the stability of L_h, it follows that

$$|\hat{u}_j - u(x_j)| = O(h^4).$$

The main problem in deferred corrections is the construction of second-order difference approximations $\hat{\tau}_{j,\pi}[u^h]$ to the truncation error term $\tau[u(x_j)]$. One way is to replace the derivatives appearing in $\tau[u(x)]$ by standard difference approximations using the finite difference solution $\{u_j\}_{j=0}^{N+1}$ in place of the exact solution $u(x)$. One problem with this approach is that it requires some modification near the end–points of the interval, or it is necessary to compute the numerical solution outside the interval I. A second approach is to use the differential equation to express $\tau[u(x)]$ in the form

$$\tau[u(x)] = C_2(x)u'(x) + C_1(x)u(x) + C_0(x), \tag{1.21}$$

where the functions C_0, C_1, C_2 are expressible in terms of the functions p, q, f and their derivatives. Then choose

$$\hat{\tau}_{j,\pi}[u^h] = C_2(x_j)\frac{u_{j+1} - u_{j-1}}{2h} + C_1(x_j)u_j + C_0(x_j).$$

Since

$$u'' = pu' + qu - f,$$

we have

$$
\begin{aligned}
u^{(3)} &= pu'' + p'u' + qu' + q'u - f' \\
&= p(pu' + qu - f) + (p' + q)u' + q'u - f' \\
&= (p^2 + p' + q)u' + (pq + q')u - (pf + f') \\
&\equiv Pu' + Qu - F.
\end{aligned} \tag{1.22}
$$

Similarly,

$$
u^{(4)} = (Pp + P' + Q)u' + (Pq + Q')u - (Pf + F'). \tag{1.23}
$$

When (1.22) and (1.23) are substituted into (1.17), we obtain the desired form (1.21).

In deferred corrections, the linear algebraic systems defining the basic difference approximation and the fourth-order approximation have the same coefficient matrix, which simplifies the algebraic problem.

If the solution u of the boundary value problem (1.1)–(1.2) is sufficiently smooth, then it can be shown that the local truncation error $\tau_{j,\pi}[u]$ has an asymptotic expansion of the form

$$
\tau_{j,\pi}[u] = \sum_{\nu=1}^{m} h^{2\nu} \tau_\nu[u(x_j)] + O(h^{2m+2}),
$$

and, if $m > 1$, deferred corrections can be extended to compute approximations of higher order than 4.

1.5.2 Numerov's method

A second higher-order finite difference method is easily constructed in the case $p = 0$, when (1.1)–(1.2) becomes

$$
\begin{aligned}
&Lu(x) = -u''(x) + q(x)u(x) = f(x), \quad x \in I, \\
&u(0) = g_0, \quad u(1) = g_1.
\end{aligned}
$$

Now

$$
\begin{aligned}
\Delta_h u(x_j) &\equiv [u(x_{j+1}) - 2u(x_j) + u(x_{j-1})]/h^2 \\
&= u''(x_j) + \frac{1}{12} h^2 u^{(4)}(x_j) + O(h^4).
\end{aligned} \tag{1.24}
$$

From the differential equation, we have

$$
u^{(4)}(x) = [q(x)u(x) - f(x)]''
$$

and therefore

$$u^{(4)}(x_j) = \Delta_h[q(x_j)u(x_j) - f(x_j)] + O(h^2). \tag{1.25}$$

Thus, from (1.24) and (1.25),

$$u''(x_j) = \Delta_h u(x_j) - \frac{1}{12}h^2\Delta_h[q(x_j)u(x_j) - f(x_j)] + O(h^4).$$

We define $\{\tilde{u}_j\}_{j=0}^{N+1}$ by

$$\begin{aligned} \mathcal{L}_h\tilde{u}_j \equiv\ & -\Delta_h\tilde{u}_j + \left(1 + \frac{1}{12}h^2\Delta_h\right) q_j\tilde{u}_j = \left(1 + \frac{1}{12}h^2\Delta_h\right) f(x_j), \quad j = 1, \ldots, N, \\ & \tilde{u}_0 = g_0, \qquad \tilde{u}_{N+1} = g_{N+1}, \end{aligned} \tag{1.26}$$

which is commonly known as *Numerov's method*. Equations (1.26) are symmetric tridiagonal and may be written in the form

$$\tilde{c}_j u_{j-1} + \tilde{d}_j u_j + \tilde{e}_j u_{j+1} = \frac{1}{12}h^2[f_{j+1} + 10f_j + f_{j-1}], \quad j = 1, \ldots, N,$$

where

$$\tilde{c}_j = -\left(1 - \frac{1}{12}h^2 q_{j-1}\right), \quad \tilde{d}_j = 2 + \frac{5}{6}h^2 q_j, \quad \tilde{e}_j = -\left(1 - \frac{1}{12}h^2 q_{j+1}\right).$$

It is easy to show that, for h sufficiently small, the coefficient matrix of this system is strictly diagonally dominant and hence $\{\tilde{u}_j\}_{j=0}^{N+1}$ is unique. From a similar analysis, it follows that the difference operator $\mathcal{L}_h$ is stable. Also, since

$$\mathcal{L}_h[\tilde{u}_j - u(x_j)] = \left(1 + \frac{1}{12}h^2\Delta_h\right) f(x_j) - \mathcal{L}_h u(x_j) = O(h^4),$$

the stability of $\mathcal{L}_h$ implies that

$$|\tilde{u}_j - u(x_j)| = O(h^4).$$

1.6 Second–Order Nonlinear Equations

In this section, we discuss finite difference methods for the solution of the second–order nonlinear two-point boundary value problem

$$\begin{aligned} & \mathcal{L}u(x) \equiv -u'' + f(x, u) = 0, \qquad x \in I, \\ & u(0) = g_0, \quad u(1) = g_1. \end{aligned} \tag{1.27}$$

The basic second–order finite difference approximation to (1.27) takes the form

$$\mathcal{L}_h u_j \equiv -\Delta_h u_j + f(x_j, u_j) = 0, \quad j = 1, \ldots, N.$$

(1.28)

$$u_0 = g_0, \quad u_{N+1} = g_1.$$

If

$$\tau_{j,\pi}[u] \equiv \mathcal{L}_h u(x_j) - \mathcal{L}u(x_j)$$

then clearly

$$\tau_{j,\pi}[u] = -\frac{h^2}{12} u^{(4)}(\xi_j), \qquad \xi_j \in (x_{j-1}, x_{j+1}),$$

if $u \in C^4(I)$. Stability of the nonlinear difference problem is defined in the following way.

Definition 1.6. *A difference problem defined by the nonlinear difference operator $\mathcal{L}_h$ is* **stable** *if, for sufficiently small h, there exists a positive constant K, independent of h, such that, for all mesh functions $\{v_j\}_{j=0}^{N+1}$ and $\{w_j\}_{j=0}^{N+1}$,*

$$|v_j - w_j| \le K\Big\{\max(|v_0 - w_0|, |v_{N+1} - w_{N+1}|) + \max_{1\le i\le N} |\mathcal{L}_h v_j - \mathcal{L}_h w_j|\Big\}.$$

Notice that when $\mathcal{L}_h$ is linear, this definition reduces to the definition of Section 1.3 applied to the mesh function $\{v_j - w_j\}_{j=0}^{N+1}$.

Theorem 1.5. *If $f_u \equiv \dfrac{\partial f}{\partial u}$ is continuous on $I \times (-\infty, \infty)$ such that*

$$0 < q_* \le f_u,$$

then the difference problem (1.28) is stable, with $K = \max(1, 1/q_*)$

Proof — For mesh functions $\{v_j\}$ and $\{w_j\}$, we have

$$\begin{aligned} \mathcal{L}_h v_j - \mathcal{L}_h w_j &= -\Delta_h(v_j - w_j) + f(x_j, v_j) - f(x_j, w_j) \\ &= -\Delta_h(v_j - w_j) + f_u(x_j, \hat{v_j})(v_j - w_j), \end{aligned}$$

on using the mean value theorem, where $\hat{v_j}$ lies between v_j and w_j. Then

$$\begin{aligned} h^2[\mathcal{L}_h v_j - \mathcal{L}_h w_j] &= c_j[v_{j-1} - w_{j-1}] + d_j[v_j - w_j] \\ &+ e_j[v_{j+1} - w_{j+1}] \end{aligned}$$

where $c_j = e_j = -1$ and

$$d_j = 2 + h^2 f_u(x_j, \hat{v_j}).$$

Clearly

$$|c_j| + |e_j| = 2 < |d_j|.$$

The remainder of the proof is similar to that of Theorem 1.2. □

An immediate consequence of the stability of $\mathcal{L}_h$ is the following:

Corollary 1.1. *If the difference approximation $\{u_j\}$ defined by (1.28) exists then it is unique.*

For a proof of the existence of the solution, see Keller (1968).

Corollary 1.2. *If u is the solution of (1.27) and $\{u_j\}_{j=0}^{N+1}$ the solution of (1.28) then, under the assumptions of Theorem 1.5,*

$$|u_j - u(x_j)| \leq \frac{h^2}{12} K \max_{x \in I} |u^{(4)}(x)|, \quad j = 0, \ldots, N+1,$$

provided $u \in C^4(I)$.

In order to determine the difference approximation $\{u_j\}_{j=0}^{N+1}$, we must solve the system of nonlinear equations (1.28), which may be written as

$$J\mathbf{u} + h^2 \mathbf{f}(\mathbf{u}) = \mathbf{g}, \tag{1.29}$$

where

$$J = \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2 \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{N-1} \\ u_N \end{bmatrix}, \quad \mathbf{f}(\mathbf{u}) = \begin{bmatrix} f(x_1, u_1) \\ f(x_2, u_2) \\ \vdots \\ f(x_{N-1}, u_{N-1}) \\ f(x_N, u_N) \end{bmatrix}, \quad \mathbf{g} = \begin{bmatrix} g_0 \\ 0 \\ \vdots \\ 0 \\ g_1 \end{bmatrix}. \tag{1.30}$$

This system is usually solved using Newton's method (or a variant of it), which is described in Section 1.8.

1.7 Numerov's Method for Second Order Nonlinear Equations

Numerov's method for the solution of (1.27) may be derived in the following way. If u is the solution of (1.27) and f is sufficiently smooth then

$$\begin{aligned} u^{(4)}(x_j) &= \frac{d^2}{dx^2} f(x,u)|_{x=x_j} \\ &= \Delta_h f(x_j, u(x_j)) + O(h^2). \end{aligned}$$

Since

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

we have

$$\Delta_h u(x_j) = f(x_j, u(x_j)) + \frac{1}{12} h^2 \Delta_h f(x_j, u(x_j)) + O(h^4).$$

Based on this equation, Numerov's method is defined by the nonlinear equations

$$\begin{aligned} &-\Delta_h u_j + [1 + \frac{1}{12} h^2 \Delta_h] f(x_j, u_j) = 0, \quad j = 1, \ldots, N \\ &u_0 = g_0, \quad u_{N+1} = g_1, \end{aligned} \tag{1.31}$$

which may be written in the form

$$J\mathbf{u} + h^2 \mathcal{B}\mathbf{f}(\mathbf{u}) = \mathbf{g}, \tag{1.32}$$

where J and $\mathbf{u}$ are as in (1.30), and

$$\mathcal{B} = \frac{1}{12} \begin{bmatrix} 10 & 1 & & & \\ 1 & 10 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & 1 \\ & & & 1 & 10 \end{bmatrix}, \mathbf{g} = \begin{bmatrix} g_0 - \frac{1}{12} h^2 f(0, g_0) \\ 0 \\ \vdots \\ 0 \\ g_1 - \frac{1}{12} h^2 f(1, g_1) \end{bmatrix}. \tag{1.33}$$

Again Newton's method is used to solve the nonlinear system (1.32).

Stepleman (1976) formulated and analyzed a fourth-order difference method for the solution of the equation

$$-u'' + f(x, u, u') = 0, \quad x \in I,$$

subject to general linear boundary conditions. When applied to the boundary value problem (1.27), this method reduces to Numerov's method. Higher-order finite difference approximations to nonlinear equations have also been derived by Doedel (1979) using techniques different from those employed by Stepleman.

1.8 Newton's Method for Systems of Nonlinear Equations

In the case of a single equation, Newton's method consists in linearizing the given equation

$$\phi(u) = 0$$

by approximating $\varphi(u)$ by

$$\phi(u^{(0)}) + \phi'(u^{(0)})(u - u^{(0)}),$$

where $u^{(0)}$ is an approximation to the actual solution, and solving the linearized equation

$$\phi(u^{(0)}) + \phi'(u^{(0)})\Delta u = 0.$$

The value $u^{(1)} = u^{(0)} + \Delta u$ is then accepted as a better approximation and the process is continued if necessary.

Consider now the N equations

$$\phi_i(u_1, u_2, \ldots, u_N) = 0, \quad i = 1, \ldots, N, \tag{1.34}$$

for the unknowns $u_1, u_2, \ldots, u_N$, which we may write in vector form as

$$\mathbf{\Phi}(\mathbf{u}) = \mathbf{0}.$$

If we linearize the i^{th} equation, we obtain

$$\phi_i(u_1^{(0)}, u_2^{(0)}, \ldots, u_N^{(0)}) + \sum_{j=1}^{N} \frac{\partial \phi_i}{\partial u_j}(u_1^{(0)}, \ldots, u_N^{(0)})\Delta u_j = 0, \quad i = 1, \ldots, N. \tag{1.35}$$

If $\mathcal{J}(\mathbf{u})$ denotes the matrix with (i, j) element $\dfrac{\partial \phi_i}{\partial u_j}(\mathbf{u})$, then (1.35) can be written in the form

$$\mathcal{J}\left(\mathbf{u}^{(0)}\right)\Delta\mathbf{u} = -\mathbf{\Phi}\left(\mathbf{u}^{(0)}\right).$$

If $\mathcal{J}(\mathbf{u}^{(0)})$ is nonsingular, then

$$\Delta\mathbf{u} = -\left[\mathcal{J}(\mathbf{u}^{(0)})\right]^{-1}\mathbf{\Phi}(\mathbf{u}^{(0)}),$$

and

$$\mathbf{u}^{(1)} = \mathbf{u}^{(0)} + \Delta\mathbf{u}$$

is taken as the new approximation. If the matrices $\mathcal{J}(\mathbf{u}^{(\nu)})$, $\nu = 1, 2, \ldots$, are nonsingular, one hopes to determine a sequence of successively better approximations $\mathbf{u}^{(\nu)}$, $\nu = 1, 2, \ldots$ from the algorithm

$$\mathbf{u}^{(\nu+1)} = \mathbf{u}^{(\nu)} + \Delta\mathbf{u}^{(\nu)}, \quad \nu = 0, 1, 2, \ldots,$$

where $\Delta\mathbf{u}^{(\nu)}$ is obtained by solving the system of linear equations

$$\mathcal{J}(\mathbf{u}^{(\nu)})\Delta\mathbf{u}^{(\nu)} = -\mathbf{\Phi}(\mathbf{u}^{(\nu)}).$$

This procedure is known as Newton's method for the solution of the system of nonlinear equations (1.34). It can be shown that as in the scalar case this procedure converges quadratically if $\mathbf{u}^{(0)}$ is chosen sufficiently close to $\mathbf{u}$, the solution of (1.34).

Now consider the system of equations

$$\begin{aligned} \phi_i(u_1, \ldots, u_N) &= -u_{i-1} + 2u_i - u_{i+1} \\ &+ \frac{h^2}{12}[f(x_{i-1}, u_{i-1}) + 10f(x_i, u_i) + f(x_{i+1}, u_{i+1})], \\ &\quad i = 1, \ldots, N, \end{aligned}$$

arising in Numerov's method (1.31). In this case, the (i, j) element of the Jacobian $\mathcal{J}$ is given by

$$\frac{\partial\phi_i}{\partial u_j} = \begin{cases} 2 + \frac{5}{6}h^2 f_u(x_i, u_i), & \text{if } j = i, \\ -1 + \frac{1}{12}h^2 f_u(x_j, u_j), & \text{if } |i - j| = 1, \\ 0, & \text{if } |i - j| > 1. \end{cases}$$

Thus

$$\mathcal{J}(\mathbf{u}) = J + h^2 BF(\mathbf{u})$$

where

$$F(\mathbf{u}) = \text{diag}(f_u(x_i, u_i)).$$

In this case, Newton's method becomes

$$\begin{aligned} \left[J + h^2 BF(\mathbf{u}^{(\nu)})\right]\Delta\mathbf{u}^{(\nu)} &= -\left[J\mathbf{u}^{(\nu)} + h^2 B\mathbf{f}(\mathbf{u}^{(\nu)}) - \mathbf{g}\right], \\ \mathbf{u}^{(\nu+1)} &= \mathbf{u}^{(\nu)} + \Delta\mathbf{u}^{(\nu)}, \qquad \nu = 0, 1, 2, \ldots. \end{aligned} \tag{1.36}$$

For the system (1.28),

$$\phi_i(u_1, \ldots, u_N) \equiv -u_{i-1} + 2u_i - u_{i+1} + h^2 f(x_i, u_i) = 0, \ i = 1, \ldots, N,$$

and Newton's method is given by (1.36) with $B = I$ and $\mathbf{g} = (g_0, 0, \ldots, 0, g_1)^T$. Note that in each case the Jacobian has the same structure and properties as the matrix arising in the linear case.

It is customary to stop the iteration (1.36) when

$$\max_{1 \le i \le N} |\Delta u_i^{(\nu)}| \le \text{XTOL}(1 + \max_{1 \le i \le N} |u_i^{(\nu)}|)$$

and

$$\max_{1 \le i \le N} |\phi_i(\mathbf{u}^{(\nu+1)})| \le \text{FTOL},$$

where XTOL and FTOL are prescribed tolerances.

2 Algorithms for Solving Tridiagonal Linear Systems

2.1 General Tridiagonal Systems

The algorithm used to solve a general tridiagonal system of the form

$$A\mathbf{u} = \mathbf{b}, \tag{2.1}$$

where

$$A = \begin{bmatrix} d_1 & e_1 & & & & \\ c_2 & d_2 & e_2 & & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & c_{N-1} & d_{N-1} & e_{N-1} \\ & & & & c_N & d_N \end{bmatrix}, \tag{2.2}$$

is a version of Gaussian elimination with partial pivoting. In the decomposition step of the algorithm, the matrix A is factored in the form

$$A = PLR, \tag{2.3}$$

where P is a permutation matrix recording the pivotal strategy, and L is unit lower bidiagonal and contains on its subdiagonal the multipliers used in the elimination process. If no row interchanges are performed, then R is upper bidiagonal, but, in general, this matrix may also have nonzero elements on its second superdiagonal. Thus fill–in may occur in this process; that is, the factored form of A may require more storage than A, in addition to a vector to record the pivotal strategy.

Using (2.3), the system (2.1) is equivalent to the systems

$$\begin{aligned} PL\mathbf{v} &= \mathbf{b}. \\ R\mathbf{u} &= \mathbf{v}. \end{aligned}$$

which can be easily solved for $\mathbf{v}$ and the desired solution $\mathbf{u}$. Software, *sgtsv*, based on this algorithm is available in LAPACK [4].

2.2 Diagonally Dominant Systems

Finite difference methods discussed in section 1 give rise to tridiagonal linear systems with certain properties which can be exploited to develop stable

algorithms in which no pivoting is required. In this section, we consider the case in which the elements of the coefficient matrix A satisfy

$$\begin{aligned} &(i) && |d_1| > |e_1|, \\ &(ii) && |d_i| \geq |c_i| + |e_i|,\ i = 2, \ldots, N-1, \\ &(iii) && |d_N| > |c_N|, \end{aligned} \tag{2.4}$$

and

$$c_i e_{i-1} \neq 0,\ i = 2, \ldots, N; \tag{2.5}$$

that is, A is irreducibly diagonally dominant and hence nonsingular. We show that, without pivoting, such a matrix can be factored in a stable fashion in the form

$$A = \hat{L}\hat{R}, \tag{2.6}$$

where $\hat{L}$ is lower bidiagonal and $\hat{R}$ is unit upper bidiagonal. The matrices $\hat{L}$ and $\hat{R}$ are given by

$$\hat{L} = \begin{bmatrix} \alpha_1 & & & & \\ c_2 & \alpha_2 & & & \\ & \cdot & \cdot & & \\ & & \cdot & \cdot & \\ & & & c_N & \alpha_N \end{bmatrix}, \hat{R} = \begin{bmatrix} 1 & \gamma_1 & & & \\ & 1 & \gamma_2 & & \\ & & \cdot & \cdot & \\ & & & 1 & \gamma_{N-1} \\ & & & & 1 \end{bmatrix}, \tag{2.7}$$

where

$$\begin{aligned} &(i) && \alpha_1 = d_1, \\ &(ii) && \gamma_i = e_i/\alpha_i, \quad \alpha_{i+1} = d_{i+1} - c_{i+1}\gamma_i, \quad i = 1, \ldots, N-1. \end{aligned}$$

Since

$$\det A = \det \hat{L} \cdot \det \hat{R} = \prod_{i=1}^{N} \alpha_i \cdot 1$$

and A is nonsingular, it follows that none of the α_i vanishes; hence the recursion in (2.8) is well–defined.

We now show that conditions (2.4) ensure that the quantities α_i and γ_i are bounded and that the bounds are independent of the order of the matrix A.

Theorem 2.1. *If the elements of A satisfy conditions (2.4), then*

$$|\gamma_i| < 1, \tag{2.8}$$

and

$$0 < |d_i| - |c_i| < |\alpha_i| < |d_i| + |c_i|, \quad i = 2, \ldots, N. \tag{2.9}$$

Proof — Since $\gamma_i = e_i/\alpha_i = e_i/d_i$,

$$|\gamma_1| = |e_1|/|d_1| < 1,$$

from (2.4(i)). Now suppose that

$$|\gamma_j| < 1, \quad j = 1, \ldots, i-1. \tag{2.10}$$

Then, with (2.8(iii)) in (2.8(ii)), we have

$$\gamma_i = e_i/(d_i - c_i\gamma_{i-1})$$

and

$$|\gamma_i| = |e_i|/(|d_i| - |c_i||\gamma_{i-1}|) < |e_i|/(|d_i| - |c_i|)$$

from (2.10). Thus, using (2.4(ii)), it follows that $|\gamma_i| < 1$, and (2.8) follows by induction.

From (2.8(ii)), (2.8) and (2.4(ii)), it is easy to show that (2.9) holds. □

The system (2.1) with A given by (2.6) is equivalent to

$$\hat{L}\mathbf{v} = \mathbf{b}, \quad \hat{R}\mathbf{u} = \mathbf{v}, \tag{2.11}$$

from which the desired solution $\mathbf{u}$ is obtained in the following fashion:

$$\begin{aligned}
&v_1 = b_1/\alpha_1 \\
&\textbf{For } i = 2 \textbf{ to } N \textbf{ do:} \\
&\quad v_i = (b_i - c_i v_{i-1})/\alpha_i \\
&\textbf{end} \\
&u_N = v_N \\
&\textbf{For } i = 1 \textbf{ to } N-1 \textbf{ do:} \\
&\quad u_{N-i} = v_{N-i} - \gamma_{N-i} u_{N-i+1} \\
&\textbf{end}
\end{aligned} \tag{2.12}$$

The nonzero elements of $\hat{L}$ and $\hat{R}$ overwrite the corresponding elements of A, and no additional storage is required.

2.3 Positive Definite Systems

In section 1, we saw that if in equation (1.1) $p = 0$, then in addition to possessing properties (2.4)–(2.5), the matrix A is also symmetric (so that $c_i = e_{i-1}$, $i = 1, \ldots, N-1$) and its diagonal elements, d_i, $i = 1, \ldots, N$, are positive. In this case, the matrix A is positive definite.

It is well known that when A is positive definite a stable decomposition is obtained using Gaussian elimination without pivoting. The LDL^T decomposition of A can easily be obtained from the recursion (2.8). If

$$\begin{bmatrix} 1 & & & & \\ l_1 & 1 & & & \\ & \cdot & \cdot & & \\ & & \cdot & 1 & \\ & & & l_{N-1} & 1 \end{bmatrix},$$

and $D = diag(\delta_1, \ldots, \delta_N)$, then by identifying l_i with γ_i, and δ_i with α_i, and setting $c_i = e_{i-1}$, we obtain from (2.8)

$$\begin{aligned} &\delta_1 = d_1 \\ &\textbf{For } i = 1 \textbf{ to } N-1 \textbf{ do:} \\ &\quad l_i = e_i/\delta_i \\ &\quad \delta_{i+1} = d_{i+1} - l_i e_i \\ &\textbf{end} \end{aligned} \tag{2.13}$$

Since A is positive definite, D is also positive definite[1], and hence the diagonal elements of D are positive. Thus the recursions (2.13) are well defined.

Using the LDL^T decomposition of A, the solution of the system (2.1) is determined by first solving

$$L\mathbf{v} = \mathbf{b},$$

which gives

$$\begin{aligned} &v_1 = b_1 \\ &v_{i+1} = b_{i+1} - l_i v_i,\ i = 1, \ldots, N-1; \end{aligned}$$

then

$$DL^T\mathbf{u} = \mathbf{v}$$

[1]For $\mathbf{u} = L^T\mathbf{v} \neq 0$, $\mathbf{u}^T D\mathbf{u} = \mathbf{v}^T LDL^T\mathbf{v} = \mathbf{v}^T A\mathbf{v} > 0$, $\mathbf{v} \neq 0$ since L nonsingular.

yields

$$u_N = v_N/\delta_N$$

and, for $i = 1, \ldots, N-1$,

$$\begin{aligned} u_{N-i} &= \frac{v_{N-i}}{\delta_{N-i}} - l_{N-i}u_{N-i+1} \\ &= (v_{N-i} - l_{N-i}\delta_{N-i}u_{N-i+1})/\delta_{N-i} \\ &= (v_{N-i} - e_{N-i}u_{N-i+1})/\delta_{N-i}, \end{aligned}$$

where in the last step we have used from (2.13) the fact that $l_{N-i}\delta_{N-i} = e_{N-i}$. Thus,

(2.14)

$v_1 = b_1$
For $i = 1$ **to** $N-1$ **do:**
  $v_{i+1} = b_{i+1} - l_i v_i$
end
$u_N = v_N/\delta_N$
For $i = 1$ **to** $N-1$ **do:**
  $u_{N-i} = (v_{N-i} - e_{N-i}u_{N-i+1})/\delta_{N-i}$
end

The code *sptsv* in LAPACK is based on (2.13)–(2.14)

3 The Finite Element Galerkin Method

3.1 Introduction

Consider the two–point boundary value problem

$$\begin{aligned} Lu \equiv -u'' + p(x)u' + q(x)u &= f(x), \quad x \in I. \\ u(0) = u(1) &= 0, \end{aligned} \tag{3.1}$$

where the functions p, q and f are smooth on I, and (1.3) holds. Let $H_0^1(I)$ denote the space of all piecewise continuously differentiable functions on I which vanish at 0 and 1. If $v \in H_0^1(I)$, then

$$-u''v + pu'v + quv = fv,$$

and

$$\int_0^1 [-u''v + pu'v + quv]dx = \int_0^1 fvdx.$$

On integrating by parts, we obtain

$$\int_0^1 u'v'dx - [u'v]|_0^1 + \int_0^1 pu'vdx + \int_0^1 quvdx = \int_0^1 fvdx.$$

Since $v(0) = v(1) = 0$, we have

$$(u', v') + (pu', v) + (qu, v) = (f, v), \quad v \in H_0^1(I), \tag{3.2}$$

where

$$(\varphi, \psi) = \int_0^1 \varphi(x)\psi(x)dx.$$

Equation (3.2) is called the *weak form* of the boundary value problem (3.1), and is written in the form

$$a(u, v) = (f, v), \quad v \in H_0^1(I), \tag{3.3}$$

where

$$a(\phi, \psi) = (\phi', \psi') + (p\phi', \psi) + (q\phi, \psi), \quad \phi, \psi \in H_0^1(I).$$

Suppose that S_h is a finite dimensional subspace of $H_0^1(I)$. The finite element Galerkin method consists in finding the element $u_h \in S_h$, the *Galerkin approximation* to the solution u of (3.1), satisfying

$$a(u_h, v) = (f, v), \quad v \in S_h. \tag{3.4}$$

Suppose $\{w_1, \ldots, w_s\}$ is a basis for S_h, and let

$$u_h(x) = \sum_{j=1}^{s} \alpha_j w_j(x). \tag{3.5}$$

Then, on substituting (3.5) in (3.4) with $v = w_i$, we have

$$(\sum_{j=1}^{s} \alpha_j w_j', w_i') + (p \sum_{j=1}^{s} \alpha_j w_j', w_i) + (q \sum_{j=1}^{s} \alpha_j w_j, w_i) = (f, w_i), \quad i = 1, \ldots, s,$$

from which it follows that

$$\sum_{j=1}^{s} [(w_j', w_i') + (p w_j', w_i) + (q w_j, w_i)] \alpha_j = (f, w_i), \quad i = 1, \ldots, s;$$

that is,

$$\mathcal{A}\boldsymbol{\alpha} = \mathbf{f}, \tag{3.6}$$

where the (i, j) element of the matrix $\mathcal{A}$ is $a(w_j, w_i)$, and the vectors $\boldsymbol{\alpha}$ and $\mathbf{f}$ are given by

$$\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_s]^T, \quad \mathbf{f} = [(f, w_1), \ldots, (f, w_s)]^T.$$

The system of equations (3.6) is often referred to as the *Galerkin equations*.

3.2 Spaces of Piecewise Polynomial Functions

The choice of the subspace S_h is a key element in the success of the Galerkin method. It is essential that S_h be chosen so that the Galerkin approximation can be computed efficiently. Secondly, S_h should possess good approximation properties as the accuracy of the Galerkin approximation u_h depends on how well u can be approximated by elements of S_h. The subspace S_h is usually chosen to be a space of piecewise polynomial functions. To define such spaces, let P_r denote the set of polynomials of degree $\leq r$, let

$$\pi : 0 = x_0 < x_1 < \ldots < x_N < x_{N+1} = 1 \tag{3.7}$$

denote a partition of $\overline{I}$, and set

$$I_j = [x_{j-1}, x_j], \quad j = 1, \ldots, N+1,$$

$h_j = x_j - x_{j-1}$ and $h = \max_j h_j$. We define

$$\mathcal{M}_k^r(\pi) = \{v \in C^k(\overline{I}) : v|_{I_j} \in P_r,\ j = 1, \ldots, N+1\},$$

where $C^k(\overline{I})$ denotes the space of functions which are k times continuously differentiable on $\overline{I}$, $0 \le k < r$, and $v|_{I_j}$ denotes the restriction of the function v to the interval I_j. We denote by $\mathcal{M}_k^{r,0}(\pi)$ the space

$$\mathcal{M}_k^r(\pi) \cap \{v | v(0) = v(1) = 0\}.$$

It is easy to see that $\mathcal{M}_k^r(\pi)$ and $\mathcal{M}_k^{r,0}(\pi)$ are linear spaces of dimensions $N(r-k)+r+1$ and $N(r-k)+r-1$, respectively. These spaces have the following approximation properties.

Theorem 3.1. *For any $u \in W_p^j(I)$, there exists a $\bar{u} \in \mathcal{M}_k^r(\pi)$ and a constant C independent of h and u such that*

$$\|(u-\bar{u})^{(\ell)}\|_{L^p(I)} \le C h^{j-\ell} \|u^{(j)}\|_{L^p(I)}, \tag{3.8}$$

for all integers ℓ and j such that $0 \le \ell \le k+1$, $\ell \le j \le r+1$. If $u \in W_p^j(I) \cap \{v | v(0) = v(1) = 0\}$ then there exists a $\bar{u} \in \mathcal{M}_k^{r,0}(\pi)$ such that (3.8) holds.

Commonly used spaces are the spaces of piecewise Hermite polynomials, $\mathcal{M}_{m-1}^{2m-1}(\pi)$, $m \ge 1$. A convenient basis for $\mathcal{M}_{m-1}^{2m-1}(\pi)$ is

$$\{S_{i,k}(x; m; \pi)\}_{i=0,\,k=0}^{N+1,\,m-1}, \tag{3.9}$$

where

$$D^l S_{i,k}(x_j; m; \pi) = \delta_{ij}\delta_{lk}, \quad 0 \le l \le m-1, \quad 0 \le j \le N+1,$$

and δ_{mn} denotes the Kronecker delta. It is easy to see that $S_{i,k}(x; m; \pi)$ is zero outside $[x_{i-1}, x_{i+1}]$. A basis for $\mathcal{M}_{m-1}^{2m-1,0}(\pi)$ is obtained by omitting from (3.9) the functions $S_{i,0}(x; m; \pi)$, $i = 0, N+1$, which are nonzero at $x = 0$ and $x = 1$.

Example 1. $\mathcal{M}_0^1(\pi)$: the space of piecewise linear functions. The dimension of this space is $N+2$, and, if $\ell_i(x) = (x - x_i)/h_{i+1}$, then the basis functions $S_{i,0}(x; 1; \pi)$, $i = 0, \ldots, N+1$, are the piecewise linear functions defined by

$$S_{0,0}(x; 1; \pi) = \begin{cases} 1 - \ell_0(x), & x \in I_1, \\ 0, & \text{otherwise}, \end{cases}$$

$$S_{i,0}(x; 1; \pi) = \begin{cases} \ell_{i-1}(x), & x \in I_i, \\ 1 - \ell_i(x), & x \in I_{i+1}, \\ 0, & \text{otherwise}, \ 1 \le i \le N, \end{cases}$$

$$S_{N+1,0}(x;1;\pi) = \begin{cases} \ell_N(x), & x \in I_{N+1}, \\ 0, & \text{otherwise.} \end{cases}$$

Thus there is one basis function associated with each node of the partition π. For convenience, we set

$$w_i(x) = S_{i,0}(x;1;\pi),\ i = 0,\ldots,N+1. \tag{3.10}$$

Note that if $u_h(x) = \sum_{j=0}^{N+1} \alpha_j w_j(x)$ then $\alpha_j = u_h(x_j),\ j = 0,\ldots,N+1$, since $w_j(x_i) = \delta_{ij}$. A basis for $\mathcal{M}_0^{1,0}(\pi)$ is obtained by omitting from (3.10) the functions $w_0(x)$ and $w_{N+1}(x)$.

Example 2. $\mathcal{M}_1^3(\pi)$: the space of piecewise Hermite cubic functions. This space has dimension $2N+4$. If

$$g_1(x) = -2x^3 + 3x^2,$$

and

$$g_2(x) = x^3 - x^2,$$

the basis functions $S_{i,0}(x;2;\pi),\ i = 0,\ldots,N+1$, are the piecewise cubic functions defined by

$$S_{0,0}(x;2;\pi) = \begin{cases} g_1(1-\ell_0(x)), & x \in I_1, \\ 0, & \text{otherwise,} \end{cases}$$

$$S_{i,0}(x;2;\pi) = \begin{cases} g_1(\ell_{i-1}(x)), & x \in I_i, \\ g_1(1-\ell_i(x)), & x \in I_{i+1}, \\ 0, & \text{otherwise, } 1 \le i \le N, \end{cases}$$

$$S_{N+1,0}(x;2;\pi) = \begin{cases} g_1(\ell_N(x)), & x \in I_{N+1}, \\ 0, & \text{otherwise,} \end{cases}$$

and the functions $S_{i,1}(x;2;\pi),\ i = 0,\ldots,N+1$, are the piecewise cubic functions

$$S_{0,1}(x;2;\pi) = \begin{cases} -h_1 g_2(1-\ell_0(x)), & x \in I_1, \\ 0, & \text{otherwise,} \end{cases}$$

$$S_{i,1}(x;2;\pi) = \begin{cases} h_i g_2(\ell_{i-1}(x)), & x \in I_i, \\ -h_{i+1} g_2(1-\ell_i(x)), & x \in I_{i+1}, \\ 0, & \text{otherwise, } 1 \le i \le N, \end{cases}$$

$$S_{N+1,1}(x;2;\pi) = \begin{cases} h_{N+1} g_2(\ell_N(x)), & x \in I_{N+1}, \\ 0, & \text{otherwise.} \end{cases}$$

For notational convenience, we write

$$v_i(x) = S_{i,0}(x;2;\pi), \quad s_i(x) = S_{i,1}(x;2;\pi), \quad i = 0,\ldots,N+1. \tag{3.11}$$

The functions v_i and s_i are known as the *value function* and the *slope function*, respectively, associated with the point $x_i \in \pi$. Note that if

$$u_h(x) = \sum_{j=0}^{N+1} \{\alpha_j v_j(x) + \beta_j s_j(x)\},$$

then

$$\alpha_j = u_h(x_j), \quad \beta_j = u_h'(x_j), \quad j = 0,\ldots,N+1,$$

since

$$v_j(x_i) = \delta_{ij}, \quad v_j'(x_i) = 0, \quad s_j(x_i) = 0, \quad s_j'(x_i) = \delta_{ij}, \quad i,j = 0,\ldots N+1.$$

A basis for $\mathcal{M}_1^{3,0}(\pi)$ is obtained by omitting from (3.11) the functions $v_0(x)$ and $v_{N+1}(x)$.

3.3 The Algebraic Problem

First we examine the structure of the coefficient matrix $\mathcal{A}$ of the Galerkin equations (3.6) in the case in which $S_h = \mathcal{M}_0^{1,0}(\pi)$. With the basis functions given by (3.10), it is clear that

$$(w_i', w_j') + (pw_i, w_j') + (qw_i, w_j) = 0 \quad \text{if} \quad |i-j| > 1,$$

since $w_i(x) = 0$, $x \notin (x_{i-1}, x_{i+1})$. Thus $\mathcal{A}$ is tridiagonal. Since the tridiagonal matrices

$$A = ((w_i', w_j')), \quad B = ((w_i, w_j)), \tag{3.12}$$

corresponding to $p = 0$, $q = 1$, occur quite often in practice, their elements are given in Appendix A. If the partition π is uniform and

$$J = h^{-2} \begin{bmatrix} 2 & -1 & & & & & & \\ -1 & 2 & -1 & & & & & \\ & \ddots & \ddots & \ddots & & & & \\ & & -1 & 2 & -1 & & & \\ & & & \ddots & \ddots & \ddots & & \\ & & & & -1 & 2 & -1 & \\ & & & & & -1 & 2 & \end{bmatrix}, \tag{3.13}$$

then it is easy to see from Appendix A that

$$A = hJ, \quad B = h\left(I - \frac{h^2}{6}J\right) = \frac{h}{6}\text{tridiag}(1 \;\; 4 \;\; 1). \tag{3.14}$$

Now consider the case in which $S_h = \mathcal{M}_1^3(\pi)$. It is convenient to consider the basis $\{w_i\}_{i=0}^{2N+3}$, where

$$w_{2i} = v_i, \quad w_{2i+1} = s_i, \quad i = 0, \ldots, N+1. \tag{3.15}$$

For this ordering of the basis functions (3.11), it follows that, since v_i and s_i vanish outside $[x_{i-1}, x_{i+1}]$, $1 \le i \le N$, the quantities (w_k', w_l'), (pw_k, w_l') and (qw_k, w_l) are nonzero only if w_k and w_l are basis functions associated with the same node or adjacent nodes of the partition π. Consequently

$$a(w_k, w_l) = 0 \quad \text{if} \quad |k - l| > 3.$$

Thus the matrix $\mathcal{A}$ whose (i, j) element is $a(w_i, w_j)$ is a band matrix of bandwidth seven. More precisely, in the $2i^{th}$ and $(2i+1)^{th}$ rows of $\mathcal{A}$, only the elements in columns $2i + j - 2$, $j = 0, 1, \ldots, 5$, can be nonzero. Thus we can partition the $(2N + 4) \times (2N + 4)$ matrix $\mathcal{A}$ in the form

$$\mathcal{A} = \begin{bmatrix} \mathcal{A}_{00} & \mathcal{A}_{01} & & & & & \\ \mathcal{A}_{10} & \mathcal{A}_{11} & \mathcal{A}_{12} & & & & \\ & \cdots & \cdots & \cdots & & & \\ & & \cdots & \cdots & \cdots & & \\ & & & \cdots & \cdots & \mathcal{A}_{N,N+1} & \\ & & & & \mathcal{A}_{N+1,N} & \mathcal{A}_{N+1,N+1} \end{bmatrix}; \tag{3.16}$$

that is, $\mathcal{A}$ is block tridiagonal, the submatrices $\mathcal{A}_{ii}$ and $\mathcal{A}_{i,i+1}$ being 2×2. The matrices $A = ((w_i', w_j'))$, $B = ((w_i, w_j))$ are given in Appendix B for the case in which the partition π is uniform. If $S_h = \mathcal{M}_1^{3,0}(\pi)$, the corresponding $(2N+2)\times(2N+2)$ matrix $\mathcal{A}$ is obtained from (3.16) by omitting the first and the $(2N + 3)^{th}$ rows and columns. It is easy to see that if $S_h = \mathcal{M}_{m-1}^{2m-1}(\pi)$ and the basis given by (3.9) is chosen, then $\mathcal{A}$ is block tridiagonal with $m \times m$ diagonal blocks.

3.4 Treatment of Inhomogeneous Dirichlet Boundary Conditions and Flux Boundary Conditions

The Galerkin method can be easily modified to treat boundary conditions other than the homogeneous Dirichlet conditions. Consider, for example,

the inhomogeneous Dirichlet boundary conditions

$$u(0) = g_0, \quad u(1) = g_1, \quad g_0 g_1 \neq 0, \tag{3.17}$$

and choose $S_h = \mathcal{M}_1^{3,0}(\pi)$. If we set

$$w = u - g$$

where

$$g(x) = g_0 v_0(x) + g_1 v_{N+1}(x)$$

then

$$w(0) = w(1) = 0,$$

and, from (3.2),

$$(p(w+g)', v') + (q(w+g), v) = (f, v), \quad v \in H_0^1(I).$$

The Galerkin approximation $U \in \mathcal{M}_1^3(\pi)$ is defined uniquely by

$$(pU', v') + (qU, v) = (f, v), \quad v \in \mathcal{M}_1^{3,0}(\pi),$$

and the condition that if

$$U(x) = \sum_{i=0}^{N+1} \{\alpha_i v_i(x) + \beta_i s_i(x)\}$$

then $\alpha_0 = g_0$ and $\alpha_{N+1} = g_1$.

With the basis functions ordered as in (3.15), the coefficient matrix of the Galerkin equations in this case is the matrix A of (3.16) with the elements of rows 1 and $2N+3$ replaced by

$$a_{1j} = \delta_{1j}, \quad j = 1, \ldots, 2N+4$$

and

$$a_{2N+3,j} = \delta_{2N+3,j}, \quad j = 1, \ldots, 2N+4,$$

respectively, and the right hand side vector has as its first and $(2N+3)^{th}$ components, g_0 and g_1, respectively.

In the case in which the general linear boundary conditions

$$\lambda_0 u(0) - p(0)u'(0) = \nu_0, \quad \lambda_1 u(1) + p(1)u'(1) = \nu_1, \tag{3.18}$$

are prescribed then as in derivation of (3.2),we have,

$$(pu', v') - [pu'v]|_0^1 + (qu, v) = (f, v), \tag{3.19}$$

for $v \in H^1(I)$. Using (3.18) in (3.19), we obtain

$$\begin{aligned}(pu', v') &+ \lambda_1 u(1)v(1) + \lambda_0 u(0)v(0) + (qu, v)\\ &= (f, v) + \nu_0 v(0) + \nu_1 v(1), \quad v \in H^1(I).\end{aligned}$$

If S_h is now a subspace of $H^1(I)$, then the Galerkin approximation $U \in S_h$ is defined by

$$(pU', v') + \lambda_1 U(1)v(1) + \lambda_0 U(0)v(0) + (qU, v) = (f, v) + \nu_0 v(0) + \nu_1 v(1), \quad v \in S_h. \tag{3.20}$$

Thus, if $S_h = \mathcal{M}_1^3(\pi)$ and the basis functions are again ordered as in (3.15), the matrix of coefficents of the Galerkin equations is obtained from (3.16) by adding λ_0 and λ_1 to the $(1,1)$ element and the $(2N+3, 2N+3)$ element, respectively. The term $\nu_0 v(0) + \nu_1 v(1)$ on the right hand side of (3.20) merely adds ν_0 to the first component and ν_1 to the $(2N+3)$–th component of the right hand side vector.

3.5 The Uniqueness of the Galerkin Approximation

We now show that there exists a unique solution $u_h \in S_h$ of (3.4) for h sufficiently small. Since the problem under consideration is linear, it is sufficient to show uniqueness.

Let U_1 and U_2 be two solutions of (3.4) and set $Y = U_1 - U_2$. Then

$$a(Y, v) = 0, \quad v \in S_h. \tag{3.21}$$

Let $\psi \in H_0^1(I)$ be the solution of

$$\begin{aligned}L^*\psi &= Y(x), \quad x \in I,\\ \psi(0) &= \psi(1) = 0,\end{aligned}$$

where L^* denotes the formal adjoint of L,

$$L^*u = -u'' - (pu)' + qu.$$

From the hypotheses on p and q, it follows that $\psi \in H^2(I) \cap H_0^1(I)$ and

$$\|\psi\|_{H^2} \le C\|Y\|_{L^2}. \tag{3.22}$$

Thus, using integration by parts and (3.21), we have

$$\|Y\|_{L^2}^2 = (L^*\psi, Y) = a(Y, \psi) = a(Y, \psi - \chi),$$

where $\chi \in S_h$, and since

$$a(\phi, \psi) \leq C\|\phi\|_{H^1}\|\psi\|_{H^1}, \tag{3.23}$$

for $\phi, \psi \in H^1(I)$, it follows that

$$\|Y\|_{L^2}^2 \leq C\|Y\|_{H^1}\|\psi - \chi\|_{H^1}.$$

From Theorem 3.1, we can choose $\chi \in S_h$ such that

$$\|\psi - \chi\|_{H^1} \leq Ch\|\psi\|_{H^2}.$$

Thus

$$\|Y\|_{L^2}^2 \leq Ch\|Y\|_{H^1}\|\psi\|_{H^2} \leq Ch\|Y\|_{H^1}\|Y\|_{L^2},$$

where in the last inequality we have used (3.22), and we obtain

$$\|Y\|_{L^2} \leq Ch\|Y\|_{H^1}. \tag{3.24}$$

Since $Y \in S_h$,

$$a(Y, Y) = 0,$$

from which it follows that

$$\|Y'\|_{L^2}^2 = -(pY', Y) - (qY, Y) \leq C\{\|Y'\|_{L^2} + \|Y\|_{L^2}\}\|Y\|_{L^2}.$$

Using this inequality and (3.24), we obtain

$$\|Y\|_{H^1}^2 \leq Ch\|Y\|_{H^1}^2.$$

Thus, for h sufficiently small,

$$\|Y\|_{H^1} = 0,$$

and hence from Sobolev's inequality,

$$Y = 0.$$

For the self-adjoint boundary value problem

$$\begin{aligned} Lu \equiv -(pu')' + q(x)u &= f(x), \quad x \in I. \\ u(0) = u(1) &= 0, \end{aligned}$$

the uniqueness of the Galerkin approximation $u_h \in S_h$ satisfying

$$(pu_h', v') + (qu_h, v) = (f, v), v \in S_h,$$

is much easier to prove. In this case, the coefficient matrix $\mathcal{A}$ of the Galerkin equations is positive definite, from which it follows immediately that the Galerkin approximation is unique. This result is proved in the following theorem.

Theorem 3.2. *The matrix $\mathcal{A}$ is positive definite.*

Proof — Suppose $\boldsymbol{\beta} \in \mathbf{R}^s$ and $\boldsymbol{\beta} \neq \mathbf{0}$. Then, if $\mathcal{A} = (a_{ij})$ and $w = \sum_{j=1}^s \beta_j w_j$, then

$$\begin{aligned}
\boldsymbol{\beta}^T \mathcal{A} \boldsymbol{\beta} &= \sum_{i,j=1}^s a_{ij}\beta_i\beta_j \\
&= \sum_{i,j=1}^s \{(p\beta_i w_i', \beta_j w_j') + (q\beta_i w_i, \beta_j w_j)\} \\
&= (pw', w') + (qw, w) \\
&= \|p^{1/2}w'\|_{L^2}^2 + \|q^{1/2}w\|_{L^2}^2 \\
&\geq p_*\|w'\|_{L^2}^2.
\end{aligned}$$

The proof is complete if $\|w'\| > 0$. Suppose $\|w'\| = 0$. Then $w' = 0$ and $w = C$, where C is a constant. Since $w \in S_h$, $w(0) = w(1) = 0$, from which it follows that $C = 0$. Therefore $w = 0$; that is $\sum_{j=1}^s \beta_j w_j = 0$. Since $\{w_1, \ldots, w_s\}$ is a basis for S_h and is therefore linearly independent, this implies that $\beta_j = 0$, $j = 1, \ldots, N$, which is a contradiction. Therefore $\|w'\| > 0$ and $\mathcal{A}$ is positive definite. □

3.6 The Accuracy of the Galerkin Approximation

3.6.1 Optimal H^1 and L^2 error estimates

In the following, an estimate of the error $u - u_h$ in an H^k–norm (or an L^p–norm) will be called *optimal in H^k (or L^p)* if the estimate has the same power of h as is possible by the approximation properties of the subspace S_h with the same smoothness assumptions on u.

Throughout this and subsequent analyses, C denotes a generic constant which is independent of u and h.

Theorem 3.3. *Suppose $u \in H^{r+1}(I) \cap H_0^1(I)$. Then, for h sufficiently small,*

$$\|u - u_h\|_{L^2} + h\|u - u_h\|_{H^1} \leq Ch^{r+1}\|u\|_{H^{r+1}}. \tag{3.25}$$

Proof — Let $\phi \in H_0^1(I)$ satisfy

$$\begin{aligned}
L^*\phi &= e(x), \quad x \in I, \\
\phi(0) &= \phi(1) = 0,
\end{aligned}$$

where $e = u - u_h$. Then, as in section 3.2 but with ϕ and e replacing ψ and Y, respectively, we have

$$\|e\|_{L^2} \leq Ch\|e\|_{H^1}, \tag{3.26}$$

since

$$a(e, v) = 0, \quad v \in S_h.$$

Since

$$a(e, e) = a(e, u - \chi), \quad \chi \in S_h,$$

it follows that

$$\begin{aligned} \|e'\|_{L^2}^2 &= -(pe', e) - (qe, e) + a(e, u - \chi) \\ &\leq C\Big\{[\|e'\|_{L^2} + \|e\|_{L^2}]\|e\|_{L^2} + \|e\|_{H^1}\|u - \chi\|_{H^1}\Big\} \end{aligned}$$

on using Schwarz's inequality and (3.23). On using (3.26), we have, for h sufficiently small,

$$\|e\|_{H^1} \leq C\|u - \chi\|_{H^1}. \tag{3.27}$$

Since $u \in H^{r+1}(I) \cap H_0^1(I)$, from Theorem 3.1, χ can be chosen so that

$$\|u - \chi\|_{H^1} \leq Ch^r\|u\|_{H^{r+1}},$$

and hence

$$\|e\|_{H^1} \leq Ch^r\|u\|_{H^{r+1}}. \tag{3.28}$$

The use of this estimate in (3.26) completes the proof. □

3.6.2 Optimal L^∞ error estimate

In this section, we derive an optimal L^∞ error estimate when $S_h = \mathcal{M}_k^{r,0}(\pi)$ and the partition π of I is from a *quasi–uniform collection of partitions*, which we now define.

Definition 3.1. *Let*

$$\pi : \ 0 = x_0 < x_1 < \ldots < x_N < x_{N+1} = 1$$

denote a partition of I and let $\Pi(I)$ denote the collection of all such partitions π of I. As before, set

$$h_i = x_i - x_{i-1}, \quad i = 1, \ldots, N + 1,$$

and $h = \max_i h_i$. *A collection of partitions* $\mathcal{C} \subset \Pi(I)$ *is called quasi–uniform if there exists a constant* $\sigma \geq 1$ *such that, for all* $\pi \in \mathcal{C}$,

$$\max_{1 \leq j \leq N+1} h h_j^{-1} \leq \sigma.$$

The following lemma, which is proved in [14], plays an important role in the derivation of the L^∞ estimate.

Lemma 3.1. *Let* Pu *be the* L^2 *projection of* u *into* $\mathcal{M}_k^r(\pi)$, *that is,*

$$(Pu - u, v) = 0 \quad v \in \mathcal{M}_k^r(\pi),$$

where $-1 \leq k \leq r-1$. *Then, if* $u \in W_\infty^{r+1}(I)$,

$$\|Pu - u\|_{L^\infty(I)} \leq C h^{r+1} \|u\|_{W_\infty^{r+1}(I)},$$

provided π *is chosen from a quasi–uniform collection of partitions of* I.

We now prove the following theorem.

Theorem 3.4. *Suppose* $S_h = \mathcal{M}_k^{r,0}(\pi)$, *with* π *chosen from a quasi–uniform collection of partitions. If* $u \in W_\infty^{r+1}(I) \cap H_0^1(I)$, *then*

$$\|u - u_h\|_{L^\infty} \leq C h^{r+1} \|u\|_{W_\infty^{r+1}}. \tag{3.29}$$

Proof — Let $W \in S_h$ satisfy

$$(W', v') = (u', v'), \quad v \in S_h.$$

Since

$$a(u - u_h, v) = 0, \quad v \in S_h,$$

it follows that

$$((W - u_h)', v') + (u - u_h, qv - (pv)') = 0,$$

for all $v \in S_h$. If we set $v = W - u_h$, then

$$\|(W - u_h)'\|_{L^2}^2 \leq C \|u - u_h\|_{L^2} \|W - u_h\|_{H^1}.$$

Therefore, since $\|\cdot\|_{H_0^1}$ and $\|\cdot\|_{H^1}$ are equivalent norms on H_0^1,

$$\|W - u_h\|_{H^1} \leq C \|u - u_h\|_{L^2}$$

and from Sobolev's inequality, we obtain

$$\|W - u_h\|_{L^\infty} \leq C\|u - u_h\|_{L^2}.$$

Then, from Theorem 3.3, it follows that

$$\|W - u_h\|_{L^\infty} \leq Ch^{r+1}\|u\|_{H^{r+1}}. \tag{3.30}$$

Since

$$\|u - u_h\|_{L^\infty} \leq \|u - W\|_{L^\infty} + \|W - u_h\|_{L^\infty}, \tag{3.31}$$

we need to estimate $\|u - W\|_{L^\infty}$ to complete the proof.

Note that since

$$((u - W)', 1) = 0,$$

it follows that W' is the L^2 projection of u' into $\mathcal{M}_{k-1}^{r-1}(\pi)$, and hence, from Lemma 3.1, we obtain

$$\|(u - W)'\|_{L^\infty} \leq Ch^r\|u'\|_{W_\infty^r} \leq Ch^r\|u\|_{W_\infty^{r+1}}. \tag{3.32}$$

Now suppose $g \in L^1$ and define G by

$$\begin{aligned} G'' &= -g(x), \quad x \in I, \\ G(0) &= G(1) = 0. \end{aligned}$$

Then

$$\|G\|_{W_1^2} \leq C\|g\|_{L^1}, \tag{3.33}$$

and, for $\chi \in S_h$,

$$(u - W, g) = -(u - W, G'') = ((u - W)', (G - \chi)').$$

On using Hölder's inequality, we obtain

$$(u - W, g) \leq \|(u - W)'\|_{L^\infty}\|(G - \chi)'\|_{L^1}.$$

From Theorem 3.1, we can choose χ so that

$$\|(G - \chi)'\|_{L^1} \leq Ch\|G\|_{W_1^2}.$$

Hence, on using (3.33), it follows that

$$|(u - W, g)| \leq Ch\|(u - W)'\|_{L^\infty}\|g\|_{L^1}.$$

On using (3.32) and duality, we have

$$\|u - W\|_{L^\infty} \leq Ch^{r+1}\|u\|_{W_\infty^{r+1}}. \tag{3.34}$$

The desired result now follows from (3.30), (3.31) and (3.34). □

3.6.3 Superconvergence results

The error estimates of Theorems 3.3 and 3.4 are optimal and consequently no better global rates of convergence are possible. However, there can be identifiable points at which the approximate solution converges at rates that exceed the optimal global rate. In the following theorem, we derive one such *superconvergence result.*

Theorem 3.5. *If* $S_h = \mathcal{M}_0^{r,0}(\pi)$ *and* $u \in H^{r+1}(I) \cap H_0^1(I)$*, then, for* h *sufficiently small,*

$$|(u - u_h)(x_i)| \leq Ch^{2r}\|u\|_{H^{r+1}}, \quad i = 0, \ldots, N+1. \tag{3.35}$$

Proof — Let $G(x, \xi)$ denote the Green's function for (3.1); that is,

$$u(x) = -(Lu, G(x, \cdot)) = a(u, G(x, \cdot))$$

for sufficiently smooth u. This representation is valid for $u \in H_0^1(I)$ and hence it can be applied to $e = u - u_h$. Thus, for $\chi \in S_h$,

$$e(x_i) = a(e, G(x_i, \cdot)) = a(e, G(x_i, \cdot) - \chi),$$

since

$$a(e, \chi) = 0, \quad \chi \in S_h.$$

Thus,

$$|e(x_i)| \leq C\|e\|_{H^1}\|G(x_i, \cdot) - \chi\|_{H^1}. \tag{3.36}$$

From the smoothness assumptions on p and q, it follows that

$$G(x_i, \cdot) \in H^{r+1}([0, x_i]) \cap H^{r+1}([x_i, 1]),$$

and

$$\|G(x_i, \cdot)\|_{H^{r+1}([0,x_i])} + \|G(x_i, \cdot)\|_{H^{r+1}([x_i,1])} \leq C.$$

Hence there exists $\chi \in S_h$ (obtained, for example, by Lagrange interpolation on each subinterval) such that

$$\|G(x_i, \cdot) - \chi\|_{H^1} \leq Ch^r. \tag{3.37}$$

From Theorem 3.3, we have

$$\|e\|_{H^1} \leq Ch^r\|u\|_{H^{r+1}}, \tag{3.38}$$

for h sufficiently small, and hence combining (3.36)–(3.38), we obtain

$$|e(x_i)| \leq Ch^{2r}\|u\|_{H^{r+1}},$$

as desired. □

A method which involves very simple auxiliary computations using the Galerkin solution can be used to produce superconvergent approximations to the derivative, [24]. First we consider approximations to $u'(0)$ and $u'(1)$. Motivated by the fact that

$$(f,(1-x)) = (-u'' + pu' + qu, 1-x) = u'(0) + a(u, 1-x),$$

we define an approximation Γ_0 to $u'(0)$ by

$$\Gamma_0 = (f, 1-x) - a(u_h, 1-x),$$

where u_h is the solution to (3.4). Also, with $1-x$ replaced by x in the above, we find that

$$u'(1) = a(u, x) - (f, x),$$

and hence we define an approximation Γ_{N+1} to $u'(1)$ by

$$\Gamma_{N+1} = a(u_h, x) - (f, x).$$

It can be shown that that if $u \in H^{r+1}(I)$, then

$$|\Gamma_j - u'(x_j)| \leq Ch^{2r}\|u\|_{H^{r+1}}, \tag{3.39}$$

$j = 0, N+1$, when for example, $S_h = \mathcal{M}_k^{r,0}(\pi)$.

If $S_h = \mathcal{M}_0^{r,0}(\pi)$, a procedure can be defined which at the nodes x_i, $i = 1, \ldots, N$, gives superconvergence results similar to (3.39). Specifically, if Γ_j, an approximation to $u'(x_j)$, $j = 1, \ldots, N$, is defined by

$$\Gamma_j = \frac{a(u_h, x)_{I_j'} - (f, x)_{I_j'}}{x_j},$$

where the subscript I_j' denotes that the inner products are taken over $I_j' = (0, x_j)$, then (3.39) holds for $j = 1, \ldots, N$. The approximation Γ_j is motivated by the fact that

$$(Lu, x)_{I_j'} = (f, x)_{I_j'},$$

and, after integration by parts,

$$-u'(x_j)x_j + a(u, x)_{I_j'} = (f, x)_{I_j'}.$$

3.6.4 Quadrature Galerkin methods

In most cases, the integrals occurring in (3.6) cannot be evaluated exactly and one must resort to an approximation technique for their evaluation. In this section, the effect of the use of certain quadrature rules the Galerkin method is discussed. To illustrate the concepts, we consider the problem

$$\begin{aligned} -(pu')' &= f, \quad x \in I, \\ u(0) &= u(1) = 0, \end{aligned}$$

where we shall assume that there exists a constant p_0 such that

$$0 < p_0 \leq p(x), \quad x \in I.$$

The Galerkin method in this case takes the form

$$(pu_h', v') = (f, v), \quad v \in S_h, \tag{3.40}$$

where we shall take S_h to be $\mathcal{M}_0^{r,0}(\pi)$. Let

$$0 \leq \rho_1 < \rho_2 < \ldots < \rho_\nu \leq 1,$$

and set

$$\rho_{ij} = x_{i-1} + h_i \rho_j, \quad i = 1, \ldots, N+1,\ j = 1, \ldots, \nu.$$

Suppose $\omega_j > 0$, $j = 1, \ldots, \nu$, and denote by

$$\langle \alpha, \beta \rangle_i = h_i \sum_{j=1}^{\nu} \omega_j (\alpha\beta)(\rho_{ij}),$$

a quadrature rule on I_i which is exact for $\alpha\beta \in P_t(I_i)$.

Set

$$\langle \alpha, \beta \rangle = \sum_{i=1}^{N+1} \langle \alpha, \beta \rangle_i.$$

If this quadrature formula is used in (3.40), the problem becomes that of finding $z_h \in S_h$ such that

$$\langle pz_h', v' \rangle = \langle f, v \rangle, \quad v \in S_h. \tag{3.41}$$

If t is sufficiently large, the existence and uniqueness of z_h follow from the next lemma.

Lemma 3.2. *If $t \geq 2r-2$,*

$$\langle pV', V' \rangle \geq p_0 \|V'\|_{L^2}^2, \quad V \in S_h.$$

Proof — Since the weights ω_j, $j = 1, \ldots, q$ are positive, and $(V')^2 \in P_{2r-2}(I_i)$,

$$\begin{aligned} \langle pV', V' \rangle &\geq p_0 \langle V', V' \rangle \\ &= p_0 \|V'\|_{L^2}^2. \end{aligned}$$

□

From this lemma, it follows that, if $t \geq 2r-2$, there exists a unique solution $z_h \in S_h$ of (3.41). In order to have $t \geq 2r-2$, it is necessary to have at least r quadrature points; the use of fewer leads to non–existence, as is shown by Douglas and Dupont (1974).

It can be shown (cf. Douglas and Dupont, 1974) that if $t \geq 2r-1$ then

$$\|u - z_h\|_{L^2} \leq Ch^{r+1} \|f\|_{H^{r+1}}.$$

In addition, the superconvergence indicated in Theorem 3.5 is preserved. More specifically

$$|(u - z_h)(x_i)| \leq Ch^{2r} \|f\|_{H^{t+1}}.$$

Notice that the use of an r–point Gaussian quadrature rule produces the desired accuracy and satisfies the condition of Lemma 3.2.

3.7 The Galerkin Method for Nonlinear Problems

Consider the boundary value problem

$$\begin{aligned} &-u'' + f(x,u) = 0, \quad x \in I, \\ &u(0) = u(1) = 0. \end{aligned}$$

It is easy to show that the weak form of this boundary value problem is

$$(u', v') + (f(u), v) = 0, \quad v \in H_0^1(I). \tag{3.42}$$

As before, let S_h be a finite dimensional subspace of $H_0^1(I)$ with basis $\{w_1, \ldots, w_s\}$. The Galerkin approximation to u is the element $u_h \in S_h$ such that

$$(u_h', v') + (f(u_h), v) = 0, \quad v \in S_h, \tag{3.43}$$

and if

$$u_h(x) = \sum_{j=1}^{s} \alpha_j w_j(x),$$

we obtain from (3.43) with $v = w_i$, the nonlinear system

$$\sum_{j=1}^{s}(w_i', w_j')\alpha_j + \Big(f(\sum_{\nu=1}^{s}\alpha_\nu w_\nu), w_i\Big) = 0, \quad i = 1, \ldots, s, \tag{3.44}$$

for $\alpha_1, \ldots, \alpha_s$. Newton's method for the solution of (3.44) can be easily derived by linearizing (3.43) to obtain

$$(u_h^{(n)\prime}, v') + (f(u_h^{(n-1)}, v) + (f_u(u_h^{(n-1)})(u_h^{(n)} - u_h^{(n-1)}), v) = 0, \quad v \in S_h,$$

where $u_h^{(0)}$ is arbitrary. If

$$u_h^{(k)} = \sum_{j=1}^{s}\alpha_j^{(k)} w_j,$$

then

$$(A + B_n)\boldsymbol{\alpha}^{(n)} = -\mathbf{F}_n + B_n\boldsymbol{\alpha}^{(n-1)}, \tag{3.45}$$

where

$$\begin{aligned} A &= ((w_i', w_j')), \ \boldsymbol{\alpha}^{(n)} = (\alpha_1^{(n)}, \ldots, \alpha_N^{(n)})^T, \\ B_n &= ((f_u(\sum_{\nu=1}^{s}\alpha_\nu^{(n-1)} w_\nu)w_i, w_j)), \\ \mathbf{F}_n &= ((f(u_h^{(n-1)}), w_i)). \end{aligned}$$

Note that (3.45) may be written in the form

$$(A + B_n)(\boldsymbol{\alpha}^{(n)} - \boldsymbol{\alpha}^{(n-1)}) = -(A\boldsymbol{\alpha}^{(n-1)} + \mathbf{F}_n).$$

A comprehensive account of the Galerkin method for second order nonlinear problems is given by Fairweather (1978).

4 The Orthogonal Spline Collocation Method

4.1 Introduction

Consider the linear second order two-point boundary value problem

$$Lu \equiv -u'' + p(x)u' + q(x)u = f(x), \quad x \in I, \tag{4.1}$$

$$\mu_0 u(0) + \nu_0 u'(0) = g_0, \quad \mu_1 u(1) + \nu_1 u'(1) = g_1, \tag{4.2}$$

where the functions p, q and f are smooth on I. Let

$$\pi : 0 = x_0 < x_1 < \ldots < x_N < x_{N+1} = 1$$

denote a partition of $\overline{I}$, and set $h_i = x_i - x_{i-1}$. In the orthogonal spline collocation method for (4.1)–(4.2), the approximate solution $u_h \in \mathcal{M}_1^r(\pi)$, $r \geq 3$. If $\{w_j\}_{j=1}^s$ is a basis for $\mathcal{M}_1^r(\pi)$, where $s = (N+1)(r-1)+2$, we may write

$$u_h(x) = \sum_{j=1}^{s} u_j w_j(x). \tag{4.3}$$

Then the coefficients $\{u_j\}_{j=1}^s$ in (4.3) are determined by requiring that u_h satisfy (4.1) at the points $\{\xi_j\}_{j=1}^{s-2}$, and the boundary conditions (4.2):

$$\begin{aligned} \mu_0 u_h(0) + \nu_0 u_h'(0) &= g_0, \\ Lu_h(\xi_j) &= f(\xi_j), \quad j = 1, 2, \ldots, s-2, \\ \mu_1 u_h(1) + \nu_1 u_h'(1) &= g_1, \end{aligned}$$

where

$$\xi_{(i-1)(r-1)+k} = x_{i-1} + h_i \sigma_k, \quad i = 1, 2, \ldots, N+1, \quad k = 1, 2, \ldots, r-1, \tag{4.4}$$

and $\{\sigma_k\}_{k=1}^{r-1}$ are the nodes for the $(r-1)$-point Gauss-Legendre quadrature rule on the interval $[0, 1]$.

As an example, consider the case when $r = 3$, for which the collocation points are

$$\xi_{2i-1} = x_{i-1} + \frac{1}{2}\left(1 - \frac{1}{\sqrt{3}}\right) h_i, \quad \xi_{2i} = x_{i-1} + \frac{1}{2}\left(1 + \frac{1}{\sqrt{3}}\right) h_i, \quad i = 1, 2, \ldots, N+1. \tag{4.5}$$

With the usual basis (3.15) for $\mathcal{M}_1^3(\pi)$, we set

$$u_h(x) = \sum_{j=0}^{N+1} \{\alpha_j v_j(x) + \beta_j s_j(x)\}.$$

Since only four basis functions,

$$v_{i-1},\ s_{i-1},\ v_i,\ s_i, \tag{4.6}$$

are nonzero on $[x_{i-1}, x_i]$, the coefficient matrix of the collocation equations is of the form

$$\begin{pmatrix} D_0 & & & & & \\ S_1 & T_1 & & & & \\ & S_2 & T_2 & & & \\ & & \ddots & \ddots & & \\ & & & S_{N+1} & T_{N+1} \\ & & & & D_1 \end{pmatrix}, \tag{4.7}$$

with $D_a = [\mu_0 \quad \nu_0]$, $D_b = [\mu_1 \quad \nu_1]$, and, for $i = 1, 2, \ldots, N+1$,

$$S_i = \begin{pmatrix} Lv_{i-1}(\xi_{2i-1}) & Ls_{i-1}(\xi_{2i-1}) \\ Lv_{i-1}(\xi_{2i}) & Ls_{i-1}(\xi_{2i}) \end{pmatrix}, \quad T_i = \begin{pmatrix} Lv_i(\xi_{2i-1}) & Ls_i(\xi_{2i-1}) \\ Lv_i(\xi_{2i}) & Ls_i(\xi_{2i}) \end{pmatrix}.$$

For $r > 3$, with the commonly used B-spline basis [8], the coefficient matrix has the form

$$\begin{pmatrix} D_0 & & & & & & & \\ W_{11} & W_{12} & W_{13} & & & & & \\ & & W_{21} & W_{22} & W_{23} & & & \\ & & & & \ddots & & & \\ & & & & W_{N+1,1} & W_{N+1,2} & W_{N+1,3} \\ & & & & & & D_1 \end{pmatrix}, \tag{4.8}$$

with $W_{i1} \in \mathcal{R}^{(r-1)\times 2}, W_{i2} \in \mathcal{R}^{(r-1)\times(r-3)}, W_{i3} \in \mathcal{R}^{(r-1)\times 2}$. The matrices (4.7) and (4.8) are called *almost block diagonal* and the collocation equations form an *almost block diagonal linear system*. There exist efficient algorithms for solving such systems based on the idea of *alternate row and column elimination*; see section 5.

For the case in which the partition π is uniform, the values at the collocation points of the basis functions (4.6) and their first and second derivatives are given in Appendix C.

4.2 The Existence and Uniqueness of the Collocation Approximation

To demonstrate analytical tools used in the proof of the existence and uniqueness of the collocation approximation and in the derivation of error

estimates, we consider the linear two-point boundary value problem

$$Lu = -u'' + p(x)u' + q(x)u = f(x), \quad x \in I,$$

(4.9)

$$u(0) = u(1) = 0.$$

Then, for $v \in H^2(I) \cap H_0^1(I)$, there exists a constant C_0 such that

$$\|v\|_{H^2(I)} \leq C_0 \|Lv\|_{L^2(I)}. \tag{4.10}$$

We define the discrete inner product $\langle \cdot, \cdot \rangle$ by

$$\langle \phi, \psi \rangle = \sum_{i=1}^{N+1} \langle \phi, \psi \rangle_i,$$

where

$$\langle \phi, \psi \rangle_i = \sum_{k=1}^{r-1} h_i \omega_j \phi(\xi_{(i-1)(r-1)+k}) \psi(\xi_{(i-1)(r-1)+k}), \quad i = 1, \ldots . N+1,$$

and $\{\xi_j\}_{j=1}^{s-2}$ are the collocation points given in (4.4) and $\{\omega_k\}_{k=1}^{r-1}$ are the weights for the $(r-1)$-point Gauss-Legendre quadrature rule on the interval $[0, 1]$.

In order to demonstrate the existence and uniqueness of the collocation approximation $u_h \in S_h \equiv \mathcal{M}_1^{r,0}(\pi)$ satisfying

$$Lu_h(\xi_j) = f(\xi_j) \quad j = 1, \ldots, s-2,$$

we use the following lemma, [13].

Lemma 4.1. *There exists a constant C such that*

$$|\langle LY, LY \rangle - \|LY\|_{L^2(I)}^2| \leq Ch \|Y\|_{H^2(I)}^2, \tag{4.11}$$

for all $Y \in S_h$.

From (4.10) and this lemma, we find that, for $Y \in S_h$,

$$\|Y\|_{H^2(I)}^2 \leq C_0 \|LY\|_{L^2(I)}^2 \leq C_0(\langle LY, LY \rangle + Ch \|Y\|_{H^2(I)}^2).$$

Thus, for h sufficiently small,

$$\|Y\|_{H^2(I)}^2 \leq C \langle LY, LY \rangle. \tag{4.12}$$

If

$$LY(\xi_j) = 0, \quad j = 1, \ldots, s-2,$$

then (4.12) implies that $Y = 0$, which proves the uniqueness and hence existence of the collocation approximation u_h.

4.3 Optimal H^2 Error Estimates

We now derive an optimal H^2 error estimate.

Theorem 4.1. *Suppose $u \in H^{r+1}(I) \cap H_0^1(I)$. Then there exists a constant C such that, for h sufficiently small,*

$$\|u - u_h\|_{H^2(I)} \le C\, h^{r-1} \|u\|_{H^{r+1}(I)}. \tag{4.13}$$

Proof — Let $T_{r,h}$ be the interpolation operator from $C^1(I)$ to S_h determined by the conditions

$$\text{(i)} \quad (T_{r,h}v)(x_i) = v(x_i),\ (T_{r,h}v)'(x_i) = v'(x_i), \quad i = 0, \ldots, N+1,$$
$$\text{(ii)} \quad (T_{r,h}v)(\eta_{ij}) = v(\eta_{ij}), \quad i = 1, \ldots, N+1,\ j = 1, \ldots, r-3,$$

for $v \in C^1(I)$, where the points η_{ij}, $j = 1, \ldots, r-3$, are certain specific points in I_i; see [13] for details. Clearly $T_{r,h}$ is local in the sense that $T_{r,h}v$ is determined on I_i by v on I_i. (Note that when $r = 3$, $T_{r,h}v$ is the Hermite cubic interpolant of v.) It can be shown that, if $W = T_{r,h}u$ and $u \in H^{r+1}(I)$,

$$\sum_{k=0}^{2} h^k \|(u - W)^{(k)}\|_{L^\infty(I_i)} \le C h_i^{r+\frac{1}{2}} \|u^{(r+1)}\|_{L^2(I_i)}, \quad i = 1, \ldots, N+1, \tag{4.14}$$

and

$$\sum_{k=0}^{2} h^k \|(u - W)^{(k)}\|_{L^2(I)} \le C\, h^{r+1} \|u^{(r+1)}\|_{L^2(I)}. \tag{4.15}$$

Then, using (4.14), it follows that

$$|L(u_h - W)(\xi_j)| = |L(u - W)(\xi_j)| \le C \sum_{k=0}^{2} \|(u - W)^{(k)}\|_{L^\infty(I_i)} \le C\, h^{r-\frac{3}{2}} \|u^{(r+1)}\|_{L^2(I_i)}.$$

Hence

$$\begin{aligned} \langle L(u_h - W), L(u_h - W) \rangle &= \sum_{i=1}^{N+1} \sum_{j=1}^{r-1} h_i \omega_j [L(u_h - W)(x_{ij})]^2 \\ &\le \sum_{i=1}^{N+1} \sum_{j=1}^{r-1} h_i^{2r-2} \omega_j \|u^{(r+1)}\|^2_{L^2(I_i)} \\ &\le C\, h^{2r-2} \|u^{(r+1)}\|^2_{L^2(I)}, \end{aligned}$$

and, from (4.12), it follows that

$$\|u_h - W\|_{H^2(I)} \leq C\, h^{r-1} \|u^{(r+1)}\|_{L^2(I)}. \tag{4.16}$$

The required result is now obtained from the triangle inequality, (4.15) and (4.16). □

4.4 Superconvergence Results

Douglas and Dupont [13] construct a "quasi–interpolant", $\hat{u} \in S_h$, which has the properties that, for $u \in W_\infty^{2r-1}(I)$,

$$|(u - \hat{u})^{(j)}(x_i)| \leq C\, h^{2r-2} \|u\|_{W_\infty^{2r-1}(I)}, \quad j = 0, 1, \tag{4.17}$$

and, if $u \in W_\infty^{2r}(I)$,

$$\begin{aligned} &L(u_h - \hat{u})(\xi_j) = \epsilon(\xi_j), \quad j = 1, \ldots, s-2, \\ &(u_h - \hat{u})(0) = (u_h - \hat{u})(1) = 0, \end{aligned} \tag{4.18}$$

where

$$|\epsilon(\xi_j)| \leq C\, h^{2r-2} \|u\|_{W_\infty^{2r}(I)}.$$

From (4.12) and (4.18), we obtain

$$\begin{aligned} \|u_h - \hat{u}\|_{H^2(I)} &\leq C \langle L(u_h - \hat{u}), L(u_h - \hat{u}) \rangle^{\frac{1}{2}} \\ &= C \langle \epsilon, \epsilon \rangle^{\frac{1}{2}} \\ &\leq C\, h^{2r-2} \|u\|_{W_\infty^{2r}(I)}, \end{aligned}$$

and hence, using Sobolev's inequality,

$$\|u_h - \hat{u}\|_{W_\infty^1(I)} \leq C\, h^{2r-2} \|u\|_{W_\infty^{2r}(I)}; \tag{4.19}$$

that is, the quasi–interpolant $\hat{u}$ differs from the collocation solution u_h along with first derivatives, uniformly by $O(h^{2r-2})$. For $r \geq 3$, this represents a higher order in h than is possible for $u - u_h$. However, using the triangle inequality, (4.17) and (4.18), we see that a superconvergence phenomenon occurs at the nodes, namely

$$|(u - u_h)^{(j)}(x_i)| \leq C\, h^{2r-2} \|u\|_{W_\infty^{2r}(I)}, \quad j = 0, 1. \tag{4.20}$$

4.5 L^2 and H^1 Error Estimates

From the inequality,

$$\left(\int_{I_i} g(x)^2\,dx\right)^{\frac{1}{2}} \le C\{h_i^{\frac{1}{2}}\,(|g(x_{i-1}| + |g(x_i)|) + h_i^2\left(\int_{I_i} g''(x)^2\,dx\right)^{\frac{1}{2}}\},$$

valid for any $g \in C^2(I_i)$, the H^2 estimate (4.13) and the superconvergence result (4.20), it follows that

$$\|u - u_h\|_{L^2} \le C\Big(h^{2r-1}\|u\|_{W^{2r}_\infty(I)} + h^{r+1}\|u\|_{H^{r+1}(I)}\Big).$$

From this inequality, (4.13) and the inequality [1],

$$\int_{I_i} |g'(x)|^2\,dx \le 54\left(h_i^{-2}\int_{I_i}|g(x)|^2\,dx + h_i^2\int_{I_i}|g''(x)|^2\,dx\right),$$

we obtain the estimate

$$\|u - u_h\|_{H^1(I)} \le C\left(h^{2r-2}\|u\|_{W^{2r}_\infty(I)} + h^r\|u\|_{H^{r+1}(I)}\right).$$

In these estimates, the exponent of h is optimal but the smoothness requirements on u are not minimal. It is shown in [13] that, by modifying the collocation procedure, the smoothness requirements on the solution can be reduced to the minimal ones required for approximation when global estimates are sought. To describe this modification, let $\hat{f}$ denote the standard L^2–projection of f into M^{r-2}_{-1}. The smoothed collocation method consists of finding $u_h \in S_h$ such that

$$(L\,u_h)(\xi_j) = \hat{f}(\xi_j), \quad j = 1, \ldots, s-2.$$

Douglas and Dupont [13] prove that, for h sufficiently small,

$$\|u - u_h\|_{H^j(I)} \le C_j h^{r+1-j}\|f\|_{H^{r-1}(I)}, \quad j = 0, 1, 2,$$

and

$$|(u - u_h)(x_i)| \le Ch^{2r}\|f\|_{H^{r-1}(I)},\ i = 1, \ldots, N.$$

Figure 1: Structure of a general ABD matrix

5 Algorithms for Solving Almost Block Diagonal Linear Systems

5.1 Introduction

In several numerical methods for solving two-point boundary value problems, there arise systems of linear algebraic equations with coefficient matrices which have a certain block structure, *almost block diagonal* (ABD), [8]; see, for example, section 4.1.

The most general ABD matrix, shown in Figure 1, has the following characteristics: the nonzero elements lie in blocks which may be of different sizes; each diagonal entry lies in a block; any column of the matrix intersects no more than two blocks (which are successive), and the overlap between successive blocks (that is, the number of columns of the matrix common to two successive blocks) need not be constant. In commonly used methods for solving two-point boundary value problems, the most frequently occurring ABD structure is shown in Figure 2, where the blocks $W^{(i)}$, $i = 1, 2, \ldots, N$, are all of equal size, and the overlap between successive blocks is constant and equal to the sum of the number of rows in TOP and BOT.

In this section, we outline efficient methods for the solution of systems with coefficient matrices having this structure. These algorithms are all variants of Gaussian elimination with partial pivoting, and are more efficient than Gaussian elimination primarily because they introduce no fill–in.

We describe the essential features of the algorithms using a simple example with a coefficient matrix of the form in Figure **??** namely the matrix of Figure **??** in which there are two 4×7 blocks $W^{(1)}, W^{(2)}$, and TOP and

Figure 2: Special ABD structure arising in BVODE solvers

Figure 3: Fill-in introduced by *SOLVEBLOK*

BOT are 2×3 and 1×3, respectively. The overlap between successive blocks is thus 3.

5.2 Gaussian Elimination with Partial Pivoting

The procedure implemented in *solveblok* [9, 10] uses conventional Gaussian elimination with partial pivoting. Fill-in may be introduced in the positions indicated $*$ in Figure 3. The possible fill-in, and consequently the possible additional storage and work, depends on the number of rows, N_T, in the block *TOP*.

Figure 4: Structure of the reduced matrix

5.3 Alternate Row and Column Elimination

This stable elimination procedure, based on the approach of Varah [23], generates no fill-in for the matrix $\mathcal{A}$ of Figure **??**. Suppose we choose a pivot from the first row. If we interchange the first column and the column containing the pivot, there is no fill-in. Moreover, if instead of performing row elimination as in conventional Gaussian elimination, we reduce the $(1,2)$ and $(1,3)$ elements to zero by column elimination, the corresponding multipliers are bounded in magnitude by unity. We repeat this process in the second step, choosing a pivot from the elements in the $(2,2)$ and $(2,3)$ positions, interchanging columns 2 and 3 if necessary and eliminating the $(2,3)$ element. If this procedure were adopted in the third step, fill-in could be introduced in the $(i,3)$ positions, $i = 7,8,9,10$. To avoid this, we switch to row elimination with partial pivoting to eliminate the $(4,3)$, $(5,3)$, $(6,3)$ elements, which does not introduce fill-in. We continue using row elimination with partial pivoting until a step is reached when fill-in could occur, at which point we switch back to the "column pivoting, column elimination" scheme. This strategy leads to a decomposition

$$\mathcal{A} = PL\tilde{\mathcal{B}}UQ,$$

where P, Q are permutation matrices recording the row and column interchanges, respectively, the unit lower and unit upper triangular matrices L, U contain the multipliers used in the row and column eliminations, respectively, and the matrix $\tilde{\mathcal{B}}$ has the structure shown in Figure 4, where $\cdot$ denotes a zeroed element. Since there is only one row or column interchange at each step, the pivotal information can be stored in a single vector of the order of

Figure 5: The coefficient matrix of the reordered system

the matrix, as in conventional Gaussian elimination. The nonzero elements of L, U can be stored in the zeroed positions in $\mathcal{A}$. The pattern of row and column eliminations is determined by N_T and the number of rows N_W in a general block $W^{(i)}$ (cf. Figure 2). In general, a sequence of N_T column eliminations is alternated with a sequence of $N_W - N_T$ row eliminations.

To solve $\mathcal{A}\mathbf{x} = \mathbf{b}$, we solve

$$PL\mathbf{z} = \mathbf{b}, \quad \tilde{\mathcal{B}}\mathbf{w} = \mathbf{z}, \quad UQ\mathbf{x} = \mathbf{w}.$$

The second step requires particular attention. If the components of $\mathbf{w}$ are ordered so that those associated with the column eliminations precede those associated with the row eliminations, and the equations are ordered accordingly, the system is reducible. In our example, if we use the ordering

$$\hat{\mathbf{w}} = [w_1, w_2, w_5, w_6, w_9, w_{10}, w_3, w_4, w_7, w_8, w_{11}]^T,$$

the coefficient matrix of the reordered equations has the structure in Figure 5. Thus, the components $w_1, w_2, w_5, w_6, w_9, w_{10}$, are determined by solving a lower triangular system and the remaining components by solving an upper triangular system.

5.4 Modified Alternate Row and Column Elimination

The alternate row and column elimination procedure can be made more efficient by using the fact that, after each sequence of row or column eliminations, a reducible matrix results. This leads to a reduction in the number of arithmetic operations because some operations involving matrix-matrix

multiplications in the decomposition phase can be deferred to the solution phase, where only matrix-vector multiplications are required. After the first sequence of column eliminations, involving a permutation matrix Q_1 and multiplier matrix U_1, say, the resulting matrix is

$$\mathcal{B}_1 = \mathcal{A}Q_1U_1 = \left(\begin{array}{c|c} C_1 & O \\ \hline M_1 & \\ & \mathcal{A}_1 \\ O & \end{array}\right),$$

where $C_1 \in \mathcal{R}^{2\times 2}$ is lower triangular, $M_1 \in \mathcal{R}^{4\times 2}$ and $\mathcal{A}_1 \in \mathcal{R}^{9\times 9}$. The equation $\mathcal{A}\mathbf{x} = \mathbf{b}$ becomes

$$\mathcal{B}_1\hat{\mathbf{x}} = \mathbf{b}, \quad \hat{\mathbf{x}} = U_1^{-1}Q_1^T\mathbf{x} = \begin{pmatrix} \hat{\mathbf{x}}_1 \\ \hat{\mathbf{x}}_2 \end{pmatrix},$$

and $\hat{\mathbf{x}}_1 \in \mathcal{R}^2$. Setting $\mathbf{b} = \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{pmatrix}$, where $\mathbf{b}_1 \in \mathcal{R}^2$, we obtain

$$C_1\hat{\mathbf{x}}_1 = \mathbf{b}_1, \quad \mathcal{A}_1\hat{\mathbf{x}}_2 = \mathbf{b}_2 - \begin{pmatrix} M_1\hat{\mathbf{x}}_1 \\ \mathbf{0} \end{pmatrix} \equiv \hat{\mathbf{b}}_2. \tag{5.1}$$

The next sequence of eliminations, that is, the first sequence of row eliminations, is applied only to $\mathcal{A}_1$ to produce another reducible matrix

$$L_1P_1\mathcal{A}_1 = \left(\begin{array}{c|cc} R_1 & N_1 & O \\ \hline O & \multicolumn{2}{c}{\mathcal{A}_2} \end{array}\right),$$

where $R_1 \in \mathcal{R}^{2\times 2}$ is upper triangular, $N_1 \in \mathcal{R}^{2\times 3}$ and $\mathcal{A}_2 \in \mathcal{R}^{7\times 7}$. If

$$\hat{\mathbf{x}}_2 = \begin{pmatrix} \tilde{\mathbf{x}}_1 \\ \tilde{\mathbf{x}}_2 \end{pmatrix}, \quad L_1P_1\hat{\mathbf{b}}_2 = \begin{pmatrix} \tilde{\mathbf{b}}_1 \\ \tilde{\mathbf{b}}_2 \end{pmatrix},$$

where $\tilde{\mathbf{x}}_1,\ \tilde{\mathbf{b}}_1 \in \mathcal{R}^2$, system (5.1) becomes

$$\mathcal{A}_2\tilde{\mathbf{x}}_2 = \tilde{\mathbf{b}}_2, \quad R_1\tilde{\mathbf{x}}_1 = \tilde{\mathbf{b}}_1 - [N_1 \quad O]\,\tilde{\mathbf{x}}_2, \tag{5.2}$$

and the next sequence of eliminations is applied to $\mathcal{A}_2$, which has the structure of the original matrix with one W block removed. Since row operations are not performed on M_1 in the second elimination step, and column operations are not performed on N_1 in the third, etc., there are savings in

arithmetic operations [11]. The decomposition phase differs from that in alternating row and column elimination in that if the r^{th} elimination step is a column (row) elimination, it leaves unaltered the first $(r-1)$ rows (columns) of the matrix. The matrix in the modified procedure has the same structure and diagonal blocks as $\tilde{B}$, and the permutation and multiplier matrices are identical.

In [11, 12], this modified alternate row and column elimination procedure was developed and implemented in the package *colrow* for systems with matrices of the form in Figure 2, and in the package *arceco* for ABD systems in which the blocks are of varying dimensions, and the first and last blocks protrude, as shown in Figure 1.

A comprehensive survey of the occurrence of, and solution techniques for, ABD linear systems is given in [3].

6 First Order Systems

6.1 Introduction

Consider the two–point boundary value problem for the first order system given by

$$\mathbf{u}' = \mathbf{f}(x, \mathbf{u}), \quad x \in I, \tag{6.1}$$

subject to the separated boundary conditions

$$\mathbf{g}_0(\mathbf{u}(0)) = \mathbf{0}, \quad \mathbf{g}_1(\mathbf{u}(1)) = 0, \tag{6.2}$$

where $\mathbf{u}$ and $\mathbf{f}$ are vector functions of order m, $\mathbf{g}_0$ and $\mathbf{g}_1$ are vector functions of order m_1 and m_2 respectively, with $m_1 + m_2 = m$, and the prime denotes differentiation with respect to x. In most practical applications, $\mathbf{f}$ is nonlinear while $\mathbf{g}_0$ and $\mathbf{g}_1$ are both linear.

Equations of the form (6.1) may arise in practice or they may be derived for one or more higher order equations of the form

$$u^{(k)} = f(x, u, u', \ldots, u^{(k-1)}),$$

subject to appropriate boundary conditions. By introducing

$$u_1 = u, \quad u_2 = u', \ldots, u_k = u^{(k-1)},$$

the first order system corresponding to such an equation is

$$\begin{aligned} u_i' &= u_{i+1} \quad i = 1, 2, \ldots, k-1, \\ u_k' &= f(x, u_1, \ldots, u_k), \end{aligned}$$

subject to rewritten boundary conditions. In many cases, it is desirable to reduce the higher–order equations in this way and to use a solution approach that is suitable for first order systems. However, it should be noted that there are situations where such a reduction is not necessary and approaches applicable to the original higher–order equation are more appropriate.

6.2 The Trapezoidal Rule

If

$$\pi : 0 = x_0 < x_1 < \ldots < x_{N+1} = 1$$

is a partition of I and $h_i = x_i - x_{i-1}$, then the trapezoidal rule for the solution of (6.1)–(6.2) takes the form

$$\boldsymbol{\phi}_i(\mathbf{U}) \equiv \mathbf{U}_i - \mathbf{U}_{i-1} - \tfrac{1}{2}h_i\{\mathbf{f}(x_i, \mathbf{U}_i) + \mathbf{f}(x_{i-1}, \mathbf{U}_{i-1})\} = \mathbf{0}, \quad i = 1, \ldots, N+1,$$
$$\boldsymbol{\phi}_L(\mathbf{U}) \equiv \mathbf{g}_0(\mathbf{U}_0) = \mathbf{0}, \quad \boldsymbol{\phi}_R(\mathbf{U}) \equiv \mathbf{g}_1(\mathbf{U}_{N+1}) = \mathbf{0},$$

where $\mathbf{U}_i \approx \mathbf{u}(x_i),\ i = 0, 1, \ldots, N+1$. The determination of the approximation

$$\mathbf{U} \equiv [\mathbf{U}_0^T, \mathbf{U}_1^T, \cdots, \mathbf{U}_{N+1}^T]^T$$

requires the solution of a system of $m(N+2)$ equations of the form

$$\boldsymbol{\Phi}(\mathbf{U}) \equiv \begin{bmatrix} \boldsymbol{\phi}_L(\mathbf{U}) \\ \boldsymbol{\phi}_0(\mathbf{U}) \\ \vdots \\ \boldsymbol{\phi}_{N+1}(\mathbf{U}) \\ \boldsymbol{\phi}_R(\mathbf{U}) \end{bmatrix} = \mathbf{0}. \tag{6.3}$$

This system is usually solved by Newton's method (or a variant of it), which takes the form

$$\begin{aligned} &(\mathcal{J}(\mathbf{U}^{\nu-1})\Delta\mathbf{U}^{\nu} = -\boldsymbol{\Phi}(\mathbf{U}^{\nu-1}), \\ &\mathbf{U}^{\nu} = \mathbf{U}^{\nu-1} + \Delta\mathbf{U}^{\nu},\ \nu = 1, 2, \ldots, \end{aligned} \tag{6.4}$$

where $\mathbf{U}^0$ is an initial approximation to $\mathbf{U}$ and the (block) matrix

$$\mathcal{J}(\mathbf{U}) \equiv \frac{\partial \boldsymbol{\Phi}(\mathbf{U})}{\partial \mathbf{U}} = \left(\frac{\partial \boldsymbol{\phi}_i}{\partial \mathbf{U}_j}\right)$$

is the Jacobian of the nonlinear system (6.3). In this case, this matrix has the ABD structure

$$\begin{bmatrix} A & & & & & \\ L_1 & R_1 & & & & \\ & L_2 & R_2 & & & \\ & & \cdot & \cdot & & \\ & & & \cdot & \cdot & \\ & & & & L_{N+1} & R_{N+1} \\ & & & & & B \end{bmatrix} \tag{6.5}$$

where

$$\begin{aligned} A &= \frac{\partial \mathbf{g}_0}{\partial \mathbf{U}_0} \text{ is } m_1 \times m, \\ L_i &= -\left[I + \frac{1}{2} h_i \frac{\partial \mathbf{f}}{\partial \mathbf{U}_{i-1}}(x_{i-1}, \mathbf{U}_{i-1}) \right] \text{ is } m \times m, \\ R_i &= I - \frac{1}{2} h_i \frac{\partial \mathbf{f}}{\partial \mathbf{U}_i}(x_i, \mathbf{U}_i) \text{ is } m \times m, \\ \text{and} & \\ B &= \frac{\partial \mathbf{g}_1}{\partial \mathbf{U}_{N+1}} \text{ is } m_2 \times m. \end{aligned}$$

It can be shown that the approximation $\mathbf{U}$ determined from the trapezoidal rule satisfies

$$\mathbf{u}(x_i) - \mathbf{U}_i = O(h^2),$$

where $h = \max_j h_j$. Under sufficient smoothness conditions on $\mathbf{u}$, higher–order approximations to $\mathbf{u}$ on the mesh π can be generated using the **method of deferred conditions** (cf. Section 1.5.1), which we shall describe briefly; full details are given by Lentini and Pereyra (1977).

The **local truncation error** of the trapezoidal rule is defined as

$$\tau_{\pi,i}[\mathbf{u}] = \{ \boldsymbol{\phi}_i(\mathbf{u}) - [\mathbf{u}' - \mathbf{f}(x, \mathbf{u})]|_{x=x_{i-1/2}} \}, \quad i = 1, \ldots, N+1,$$

where $x_{i-1/2} = x_{i-1} + \frac{1}{2} h_i$. By expanding in Taylor series about $x_{i-1/2}$, it is easy to show that

$$\tau_{\pi,i}[\mathbf{u}] = \sum_{\nu=1}^{L} \mathbf{T}_\nu(x_{i-1/2}) h_i^{2\nu} + O(h^{2L+2}), \quad i = 1, \ldots, N+1, \tag{6.6}$$

where

$$\mathbf{T}_\nu(x_{i-1/2}) = -\frac{\nu}{2^{2\nu-1}(2\nu+1)!}\frac{d^{2\nu}}{dx^{2\nu}}\mathbf{f}(x,\mathbf{u})|_{x=x_{i-1/2}}.$$

If $\mathbf{S}_k(\mathbf{U}^{(k-1)})$ is a finite difference approximation of order h^{2k+2} to the first k terms in the expansion (6.6) of the local truncation error, then the solution $\mathbf{U}^{(k)}$ of the system

$$\boldsymbol{\phi}(\mathbf{v}) = \mathbf{S}_k(U^{(k-1)}), \quad k = 1, 2, \ldots, L, \tag{6.7}$$

where $\mathbf{U}^{(0)} = \mathbf{U}$, the solution of (6.3), is an order h^{2k+2} approximation to $\mathbf{u}$ on the mesh π. Details of the construction of the operators $\mathbf{S}_k$ by numerical differentiation are given by Lentini and Pereyra (1974).

Note that the systems (6.3) and (6.7) are similar, the right hand side of (6.7) being simply a known vector. Once the first system (6.3) is solved, the remaining systems (6.7) are small perturbations of (6.3) and considerable computational effort can be saved by keeping the Jacobian, and therefore its LU decomposition, fixed during the iteration (6.4). Of course, if the mesh π is changed, the Jacobian must be recomputed and refactored. In principle, this modification degrades the quadratic convergence of Newton's method, but because of the accuracy of the approximate Jacobian and the initial approximation, convergence is still rapid.

6.3 Orthogonal Spline Collocation

This method produces an approximation to $\mathbf{u}$ which is a piecewise polynomial vector function $\mathbf{U}(x) = (U_1(x), \ldots, U_m(x))$, where, for each i, $i = 1, \ldots, m$, $U_i \in \mathcal{M}_0^r(\pi)$. Each piecewise polynomial U_i is represented in the form

$$U_i(x) = \sum_{l=1}^{s} \alpha_{li} B_l(x), \tag{6.8}$$

where $s = r(N+1)+1$, and $\{B_l(x)\}_{l=1}^M$ is a convenient basis, commonly known as the B–spline basis (de Boor, 1978), for the piecewise polynomial space $\mathcal{M}_0^r(\pi)$. The coefficients $\{\alpha_{li}\}$ are determined by requiring that the approximate solution $\mathbf{U}$ satisfy the boundary conditions, and the differential equation at r specific points, the collocation points, in each subinterval. These points are the Gauss points defined by

$$\xi_{(i-1)(r-1)+k} = x_{i-1} + h_i\rho_k, \quad i = 1, \ldots, N+1, \quad k = 1, \ldots, r,$$

where $\{\rho_k\}_{k=1}^r$ are the r zeros of the Legendre polynomial of degree r on the interval $[0,1]$ (cf., Section 4.1). The collocation equations then take the

form

$$\boldsymbol{\phi}_0(\mathbf{U}) \equiv \mathbf{g}_0(\mathbf{U}(0)) = \mathbf{0},$$

$$\boldsymbol{\phi}_l(\mathbf{U}) \equiv \mathbf{U}'(\xi_j) - \mathbf{f}(\xi_j, \mathbf{U}(\xi_j)) = \mathbf{0}, \quad j = 1, 2, \ldots, s-2, \tag{6.9}$$

$$\boldsymbol{\phi}_s(\mathbf{U}) \equiv \mathbf{g}_1(\mathbf{U}(1)) = \mathbf{0}.$$

Again a variant of Newton's method is usually used to solve the nonlinear system (6.9). Certain properties of the B–splines ensure that the Jacobian of (6.9) is almost block diagonal but having a more general structure than that of the matrix (6.5). These properties are (de Boor, 1978):

- On each subinterval $[x_{i-1}, x_i]$ only $r+1$ B–splines are non–zero, namely
$$B_{r(i-2)+1}, \ldots, B_{r(i-1)+1}.$$
- At $x = 0$, $B_i(0) = \delta_{i,1}$ and at $x = 1$, $B_i(1) = \delta_{i,s}$ where δ_{ij} denotes the Kronecker delta.

If the unknown coefficients in (6.8) are ordered by "columns", that is, first with respect to i and then with respect to l, the Jacobian of (6.9) has the form

$$\begin{bmatrix}
A & & & & & & & & \\
W_{11} & W_{12} & W_{13} & & & & & & \\
 & & W_{21} & W_{22} & W_{23} & & & & \\
 & & & & \cdot & & & & \\
 & & & & & \cdot & & & \\
 & & & & & & \cdot & & \\
 & & & & & & W_{N+1,1} & W_{N+1,2} & W_{N+1,3} \\
 & & & & & & & & B
\end{bmatrix}$$

where the matrices A and B are $m_1 \times m$ and $m_2 \times m$ respectively, and W_{i1}, W_{i2} and W_{i3} are $mr \times m$, $mr \times m(r-1)$ and $mr \times m$, respectively.

Under sufficient smoothness conditions, it can be shown (see, for example Ascher et al., 1979) that the error in $\mathbf{U}$ for $x \in [x_{i-1}, x_i]$ is given by

$$u_j(x) - U_j(x) = c(x) u_j^{(r+1)}(x_{i-1}) h_i^{r+1} + O(h^{r+2}), \quad j = 1, \ldots, m,$$

where $c(x)$ is a known bounded function of x, and at each mesh point x_i,

$$u_j(x_i) - U_j(x_i) = O(h^{2r}), \quad j = 1, \ldots, m, \quad i = 0, 1, \ldots, N+1.$$

6.4 Multiple Shooting

This procedure involves finding an approximation $\mathbf{U}_i$ to $\mathbf{u}(x_i)$, $i = 0, 1, \ldots, N+1$, in the following way (cf. Stoer and Bulirsch, 1980). Let $\mathbf{u}_i(x; \mathbf{U}_i)$ denote the solution of the initial value problem

$$\mathbf{u}' = \mathbf{f}(x, \mathbf{u}), \quad \mathbf{u}(x_i) = \mathbf{U}_i. \tag{6.10}$$

Then the method consists in finding the vectors $\mathbf{U}_i$, $i = 0, 1, \ldots, N+1$, such that

a) the function defined by

$$\begin{aligned} \mathbf{u}(x) &= \mathbf{u}_i(x; \mathbf{U}_i) \text{ on } [x_i, x_{i+1}],\ i = 0, 1, \ldots, N, \\ \mathbf{u}(1) &= \mathbf{U}_N \end{aligned}$$

is continuous on I, and is thus a solution of the differential equation (6.1), and

b) the function $\mathbf{u}$ satisfies the boundary conditions (6.1).

These requirements lead to the equations

$$\mathbf{g}_0(\mathbf{U}_0) = \mathbf{0}, \tag{6.11}$$

$$\mathbf{U}_{i+1} - \mathbf{u}_i(x_{i+1}; \mathbf{U}_i) = \mathbf{0},\ i = 0, \ldots, N, \tag{6.12}$$

$$\mathbf{g}_1(\mathbf{U}_{N+1}) = \mathbf{0}, \tag{6.13}$$

where equations (6.12) arise from the continuity constraints, and (6.11) and (6.13) from the boundary conditions. When Newton's method is used to solve this system, the Jacobian is again almost block diagonal, and, in this case, takes the form

$$\begin{bmatrix} A & & & & & \\ L_0 & I & & & & \\ & L_1 & I & & & \\ & & \cdot & \cdot & & \\ & & & \cdot & \cdot & \\ & & & & L_N & I \\ & & & & & B \end{bmatrix}$$

where

$$A = \frac{\partial \mathbf{g}_0}{\partial \mathbf{U}_1} \text{ is } m_1 \times m, \quad L_i = -\frac{\partial \mathbf{u}_i(x_{i+1}; \mathbf{U}_i)}{\partial \mathbf{U}_i} \text{ is } m \times m, \quad B = \frac{\partial \mathbf{g}_1}{\partial \mathbf{U}_N} \text{ is } m_2 \times m,$$

and I denotes the $m \times m$ unit matrix. In order to determine the $m \times m$ matrices L_i, $i = 0, 1, \ldots, N$, one must solve the initial value problems

$$L_i'(x) = J(x, \mathbf{u}_i(x; \mathbf{U}_i))L_i(x), \quad L_i(x_i) = I, \; i = 0, \ldots, N, \tag{6.14}$$

where J is the Jacobian matrix $J = \dfrac{\partial \mathbf{f}}{\partial \mathbf{u}}$. Then $L_i = L_i(x_{i+1})$. Given the function $\mathbf{f}$ and its Jacobian J, one can solve the initial value problems (6.10) and (6.14) simultaneously using the values of $\mathbf{U}$ calculated from the previous Newton iteration. In practice, these initial value problems are solved numerically using an initial value routine, usually a Runge–Kutta code.

6.5 Software Packages

(THIS SECTION IS OUTDATED) In recent years, several general–purpose, portable software packages capable of handling nonlinear boundary value problems involving mixed order systems of ordinary differential equations have been developed. The use of these packages has led to the solution of problems which were difficult and/or costly to solve, if not intractable, using standard solution procedures, and has removed much of the guesswork from the problem of solving boundary value problems for ordinary differential equations, (see, for example, Davis and Fairweather (1981), Denison et al. (1983), Muir et al. (1983), Fairweather and Vedha–Nayagam, (1987), Bhattacharya et al. (1986), Akyurtlu et al. (1986), Vedha–Nayagam et al. (1987)).

In this section, we shall discuss three packages which are widely available and are now in common use. In addition to implementing one of the basic numerical methods described in Sections 8.2-8.4, each package incorporates a sophisticated nonlinear equations solver, which involves a linear equations solver, and a complex, fully–automatic mesh selection and error estimation algorithm upon which the success of the code largely depends.

The *IMSL* package, *DVCPR*, called *BVPFD* in the new *IMSL* Library, implements a variable order finite difference method based on the trapezoidal rule with deferred corrections. Like the package *DD04AD* in the Harwell Subroutine Library (1978) and *D02RAF* in the NAG Library, this package

is a version of the code *PASVA3* (Pereyra, 1979). The code *PASVA3* has fairly long history and a series of predecessors has been in use for several years. The package *COLSYS* (Ascher et al., 1981), which implements spline collocation at Gauss points, and *IMSL*'s multiple shooting code, *DTPTB*, called *BVPMS* in the new *IMSL* Library, are of a more recent vintage. All three codes are documented elsewhere, and in this section we shall only briefly mention some of their similarities and differences, and other noteworthy features of the codes. It should be noted that Bader and Ascher (1987) have developed a new version of *COLSYS* called *COLNEW* which differs from *COLSYS* principally in its use of certain monomial basis functions instead of B-splines. This new code, which is reputed to be somewhat more robust that its predecessor, shares many of its features, and, in the remainder of this section, any comments referring to *COLSYS* apply equally well to *COLNEW*.

The codes *DVCPR* and *DTPTB* require that the boundary value problem be formulated as a first–order system, and they can handle nonseparated boundary conditions, i.e. boundary conditions for (1.1), for example, of the form

$$\mathbf{g}(\mathbf{u}(0), \mathbf{u}(1)) = \mathbf{0},$$

where $\mathbf{g}$ is a vector function of order m. On the other hand, *COLSYS* can handle a mixed–order system of multipoint boundary value problems without first reducing it to a first–order system, but requires that the boundary conditions be separated. This restriction is not a serious one as a boundary value problem with nonseparated boundary conditions can be reformulated so that only separated boundary conditions occur, as we shall see in Section 10.2. However this reformulation does increase the size of the problem.

The algebraic problem arising in the codes are similar but there is no uniformity in the manner in which they are solved. Each code uses some variant of Newton's method for the solution of the nonlinear systems, and a Gauss elimination–based algorithm to solve the almost block diagonal linear systems. Since the codes were written, the package *COLROW*, (Diaz et al., 1983) has been developed specifically for the solution of such linear systems; see Section 9.4. Its use in the existing codes has not been investigated, but may improve their efficiency significantly.

Both *DVCPR* and *COLSYS* solve the boundary value problem on a sequence of meshes until user–specified error tolerances are satisfied. Detailed descriptions and theoretical justification of the automatic mesh–selection procedures used in *DVCPR* and *COLSYS* are presented by Lentini and Pereyra (1974) and Ascher et al. (1979), respectively. It is worth noting

that *DVCPR* constructs meshes in such a way that each mesh generated contains the initial mesh, which is not necessarily the case in *COLSYS*.

In *DTPTB*, the "shooting points" $x_1, \ldots, x_N$, are chosen by the user or, for linear problems, by the code itself. Any adaptivity in the code appears in the initial value solver, which in this case is *IMSL*'s Runge–Kutta code, *DVERK*.

In *DVCPR*, one can specify only an absolute tolerance, ϵ, say, which is imposed on all components of the solution. The code attempts to find an approximate solution $\mathbf{U}$ such that the absolute error in each of its components is bounded by ϵ. On the other hand, *COLSYS* allows the user to specify a different tolerance for each component of the solution, and in fact allows one to impose no tolerance at all on some or all of the components. This code attempts to obtain an approximate solution $\mathbf{U}$ such that

$$\max_{x\in[x_i,x_{i+1}]} |u_l(x) - U_l(x)| \leq \epsilon_l + \max_{x\in[x_i,x_{i+1}]} |U_l(x)|\epsilon_l$$

for l^{th} component on which the tolerance ϵ_l is imposed. This criterion has a definite advantage in the case of components with different order of magnitude or when components differ in magnitude over the interval of definition. Moreover, it is often more convenient to provide an array of tolerances than to rescale the problem. On successful termination, both *DVCPR* and *COLSYS* return estimates of the errors in the components of approximate solution.

In *DTPTB*, the same absolute tolerance is imposed on all components of the solution. In addition, a boundary condition tolerance is specified, and, on successful termination, the solution returned will also satisfy the boundary conditions to within this tolerance.

Each code offers the possibility of providing an initial approximation to the solution and an initial subdivision of the interval of definition (the shooting points in the case of *DTPTB*). With *COLSYS*, one may also specify the number of collocation points per subinterval. In many instances, parameter continuation is used to generate initial approximations and subdivisions. That is, when solving a parameterized family of problems in increasing order of difficulty, the subdivision and initial approximation for a particular problem are derived from the final mesh and the approximate solution calculated in the previous problem. This is an exceedingly useful technique which improves the robustness of the packages. It should be noted that *COLSYS* requires a *continuous* initial approximation, which is sometimes an inconvenience.

It has been the authors' experience that *DTPTB* is the least robust of the three codes. However W. H. Enright and T. F. Fairgrieve (private communication) have made some modifications to this code which have greatly improved its performance, making it competitive with both *DVCPR* and *COLSYS*. In general, there is little to choose between *DVCPR* and *COLSYS*, and it is recommended that both be used to solve a given boundary value problem. Driver programs for these codes are straightforward to write. Moreover, there are definite advantages to using both codes. For example, each provides its own insights when solving a problem, simple programming errors can be detected more quickly by comparing results, and considerable confidence can be attached to a solution obtained by two different methods. With the wide availability of these software packages, it is rarely advisable or even necessary for the practitioner to develop *ad hoc* computer programs for solving boundary value problems ordinary differential equations.

7 Reformulation of Boundary Value Problems into Standard Form

7.1 Introduction

Boundary value problems arising in various applications are frequently not in the form required by existing general purpose software packages. However, many problems can be easily converted to such a form thus enabling the user to take advantage of the availability of this software and avoid the need to develop a special purpose code.

In this section, we describe the conversion to "standard" form of some rather common "nonstandard" problems which the author has encountered in applications in chemical engineering and mechanical engineering. Some of the material presented is adapted from the excellent survey article [5], where additional examples may be found.

7.2 Nonseparated Boundary Conditions

The standard form required by several boundary value problem codes is

$$\begin{aligned} &\mathbf{u}' = \mathbf{f}(x, \mathbf{u}), \quad x \in I, \\ &\mathbf{g}(\mathbf{u}(0), \mathbf{u}(1)) = \mathbf{0}, \end{aligned} \tag{7.1}$$

where $\mathbf{u}$, $\mathbf{f}$ and $\mathbf{g}$ have m components and $\mathbf{f}$ and/or $\mathbf{g}$ may be nonlinear. The boundary conditions in (7.1) are said to be **nonseparated**.

While *COLSYS/COLNEW* can handle mixed order systems directly, it does not accept nonseparated boundary conditions. A simple way to convert a *BVP* of the form (7.1) with nonseparated boundary conditions to one with separated boundary conditions, where each condition involves only one point, is the following. We introduce the constant function $\mathbf{v}$ such that

$$\mathbf{v}(x) = \mathbf{u}(1), \quad x \in I,$$

and obtain the equivalent boundary value problem

$$\mathbf{u}' = \mathbf{f}(x, \mathbf{u}), \ \mathbf{v}' = \mathbf{0}, \quad x \in I,$$
$$\mathbf{g}(\mathbf{u}(0), \mathbf{v}(0)) = \mathbf{0}, \quad \mathbf{u}(1) = \mathbf{v}(1).$$

The penalty in this approach is that the transformed problem is twice the order of the original problem. On the other hand, as we have seen, the application of standard techniques to boundary value problems with separated boundary conditions involves the solution of almost block diagonal linear systems. The algebraic problem is significantly more involved when these techniques are applied directly to the boundary value problem (7.1).

7.3 Singular Coefficients

In problems involving cylindrical or spherical geometry, ordinary differential equations with singular coefficients often occur. A prime example is the problem of diffusion and reaction in porous catalytic solids which gives rise to a boundary value problem of the form

$$u'' + \frac{s}{x}u' = R(u), \quad x \in I, \tag{7.2}$$

subject to the boundary conditions

$$u'(0) = 0, \quad u(1) = 1. \tag{7.3}$$

In (7.2), the quantity s, is a geometric factor which depends on the geometry of the catalyst pellets: $s = 0, 1, 2$ for rectangular, cylindrical, and spherical geometries, respectively. When $s = 1$ or 2, codes such as *DVCPR* and *DTPTB* cannot be used directly because of the presence of the singular coefficient in the differential equation (7.2). Since $u'(x) = 0$, and $\lim_{x\to 0} \dfrac{u'(x)}{x} = u''(0)$, the differential equation at $x = 0$ is replaced by

$$u'' = \frac{R(u)}{(1+s)}.$$

With this modification, the codes *DVCPR* (a predecessor of *BVPFD*) and *DTPTB* (a predessor of *BVPMS*) have been used successfully to solve problems of the form (7.2)–(7.3); see, for example, [15, 18].

7.4 Continuation

Many problems arising in engineering applications depend on one (or more) parameters. Consider, for example, the boundary value problem

$$\begin{aligned} &\mathbf{u}' = \mathbf{f}(x, \mathbf{u}; \lambda), \quad x \in I, \\ &\mathbf{g}(\mathbf{u}(0), \mathbf{u}(1); \lambda) = \mathbf{0}, \end{aligned} \tag{7.4}$$

where λ is a parameter. For the desired value of λ, λ_1 say, the problem (7.4) might be exceedingly difficult to solve without a very good initial approximation to the solution $\mathbf{u}$. If the solution of (7.4) is easily determined for another value of λ, λ_0 say, then a chain of continuation steps along the homotopy path $(\lambda, \mathbf{u}(\,\cdot\,;\lambda))$ may be initiated from $(\lambda_0, \mathbf{u}(\,\cdot\,;\lambda_0))$. At each step of the chain, standard software can be used to determine an approximation to $\mathbf{u}(\,\cdot\,;\lambda)$ which is then used, at the next step, as the initial approximation to $\mathbf{u}(\,\cdot\,;\lambda + \Delta\lambda)$. This procedure can also be used when the solution of (7.4) is required for a sequence of values of λ, and the problems are solved in increasing order of difficulty.

As an example, consider the boundary value problem

$$\begin{aligned} &u'' + \frac{s}{x}u' + \lambda e^u = 0, \quad x \in I, \\ &u'(0) = 0, \quad u(1) = 0, \end{aligned} \tag{7.5}$$

which describes the steady–state temperature distribution in the radial direction in the interior of a cylinder of unit radius when $s = 1$, and a sphere of unit radius when $s = 2$ with heat generation according to the exponential law [21]. There is no solution for $\lambda > 1.7$ when $s = 1$ and for $\lambda > 3.0$ when $s = 2$. The quantity of interest in this problem is the dimensionless center temperature $u(0)$. The boundary value problem (7.5) has been solved for various values of λ in [18] using *DVCPR* and *DTPTB* (replacing the differential equation in (7.5) by

$$u'' + \frac{\lambda e^u}{(1+s)} = 0,$$

when $x = 0$, as described in section 3) and *COLSYS*, and comparisons made with results obtained in [21]. The results for the case $s = 1$ are presented in

λ	[21]	*COLSYS*
0.1	0.252(−1)	0.254822(−1)
0.2	0.519(−1)	0.519865(−1)
0.3	0.796(−1)	0.796091(−1)
0.4	0.109	0.108461
0.5	0.138	0.138673
0.6	0.169	0.170397
0.7	0.2035	0.203815
0.8	0.238	0.239148
1.0	0.65	0.316694
1.5	1.1	0.575364
1.7	2.0	0.731577

Table 1: Dimensionless center temperature $u(0)$ for the case $s = 1$.

Table 1. For each value of λ, *COLSYS* converged without difficulty while it was necessary to use continuation with the other codes; *DVCPR* was run starting with $\lambda = 1.7$ and decreasing λ in steps of 0.1 to $\lambda = 0.1$ but did not converge for $\lambda = 1.7$, while *DTPTB* was run with λ increasing in steps of 0.1 from 0.1. In most cases, the results produced by *DVCPR* (except for $\lambda = 1.7$) and *DTPTB* agree to six significant figures with those obtained by *COLSYS*, casting doubt on the validity of the results [21] for $\lambda \geq 1$. Details of these numerical experiments and a discussion of the case $s = 2$ are given in [18].

7.5 Boundary Conditions at Infinity

Consider the following boundary value problem on the semi–infinite interval $[0, \infty)$:

$$\begin{aligned} &\mathbf{u}' = \mathbf{f}(x, \mathbf{u}), \quad x \in [0, \infty), \\ &\mathbf{g}(\mathbf{u}(0), \lim_{x\to\infty} \mathbf{u}(x)) = \mathbf{0}. \end{aligned} \tag{7.6}$$

Usually the components of the solution $\mathbf{u}$ approximate their asymptotic values quite accurately at a value x, L say, of moderate size. The boundary

value problem (7.6) can then be replaced by the boundary value problem

$$\begin{aligned} &\mathbf{u}' = \mathbf{f}(x, \mathbf{u}), \quad x \in [0, L], \\ &\mathbf{g}\left(\mathbf{u}(0), \mathbf{u}(L)\right) = \mathbf{0}. \end{aligned}$$

It is usually necessary to determine the value of L experimentally; this can be facilitated by using the transformation

$$t = \frac{x}{L}$$

to map the interval $[0, L]$ to the unit interval. The quantity L then appears as a parameter in the differential equation and continuation as described in Section 7.4 can be used to determine an appropriate value for L.

Consider the following example. In the study conducted by Na and Pop [20] of free–convective flow past a vertical plate embedded in a saturated porous medium, the governing equations reduce to the boundary value problem

$$\begin{aligned} &f''' + \tfrac{1}{3}(m+2)ff'' - \tfrac{1}{3}(2m+1)(f')^2 = 0, \quad x \in [0, \infty), \\ &f(0) = 0,\ f''(0) = -1,\ f'(\infty) = 0, \end{aligned} \tag{7.7}$$

where m is a parameter. The boundary condition at infinity is replaced by $f'(L) = 0$, and, with $z = x/L$, (7.7) becomes

$$\begin{aligned} &f''' + \tfrac{1}{3}L(m+2)ff'' - \tfrac{1}{3}L(2m+1)(f')^2 = 0, \quad z \in I, \\ &f(0) = 0,\ f''(0) = -L^2,\ f'(1) = 0, \end{aligned} \tag{7.8}$$

which can now be solved directly by *COLSYS* or reduced to a first order system and solved by *DVCPR* or *DTPTB*. The quantity of interest in this problem is $f'(0)$ and the boundary value problem (7.8) is solved for an increasing sequence of values of L until $f'(0)$ remains constant to the desired number of significant figures; see [18].

7.6 Eigenvalue Problems

As an example of problems in this category, consider the following boundary value problem which arises in the study of squeezing flow of a viscous fluid between elliptic plates [6]:

$$\begin{aligned} &f''' + k = SF(f, g), \quad \eta \in I, \\ &g''' + \beta k = SF(g, f), \quad \eta \in I, \end{aligned} \tag{7.9}$$

S	-0.5	0.0	1.0	25.0
(a)	1.3023	3.0000	6.2603	73.8652
(b)	1.3036	3.0018	6.2778	74.0414

Table 2: Values of k for $\beta = 1$ and various values of S.

where

$$F(\varphi, \psi) = 2\varphi' + \eta\varphi'' + \frac{1}{2}(\varphi')^2 - \frac{1}{2}\varphi''(\varphi + \psi),$$

subject to the boundary conditions

$$\begin{aligned} &f(0) = f''(0) = g(0) = g''(0) = 0, \\ &f(1) + g(1) = 2,\ f'(1) = g'(1) = 0. \end{aligned} \tag{7.10}$$

In (7.9), the parameters β and S are prescribed, and k is an unknown constant. If we add to the boundary value problem (7.9)–(7.10) the trivial differential equation

$$k' = 0,$$

the augmented boundary value problem can be solved using standard software. In Table 2, we give the values of k for $\beta = 1$ and various values of S determined by:

(a) solving the augmented boundary value problem using *COLSYS* and *DVCPR*;

(b) [6].

Since *COLSYS* and *DVCPR* produce the same results to the number of figures presented, it seems reasonable to assume that the results given by (a) are the correct values.

7.7 Integral Constraints

In many boundary value problems, a term of the form

$$\int_0^1 G(u(t), t)\, dt = R,$$

where R is a given constant, occurs as a constraint. In such cases, we define

$$w(x) = \int_0^x G(u(t), t)\, dt,$$

and replace the constraint by the differential equation

$$w'(x) = G(u(x), x)$$

and the boundary conditions

$$w(0) = 0, \quad w(1) = R.$$

As an example, consider the following boundary value problem which arises in a study of membrane separation processes:

$$(7.11)\quad u'' + \frac{1}{x}u' + c_1 + c_2(1 - e^{-\varphi(x,k)})\,[\psi(u,\varphi) - 1] + c_3\psi(u,\varphi)u = 0,$$

where

$$\varphi(x,k) = k/(c_4 - xk),$$
$$\psi(u,\varphi) = e^u/[1 + c_5 e^{\varphi}(e^u - 1)],$$

subject to

$$(7.12)\qquad u'(0) = 0, \qquad u(1) = 0,$$

$$(7.13)\qquad \int_0^1 \psi(u,\varphi)x\,dx \;=\; c_6\int_0^1 ux\,dx,$$

In the boundary value problem (7.11)–(7.13), the quantities $c_i,\ i = 1, \ldots, 6$, are prescribed constants and the constant k is to be determined in addition to the function u. To reformulate this problem, we first write (7.13) as

$$\int_0^1 F(u,x,k)x\,dx = 0,$$

and set

$$w(x) = \int_0^x F(u,x,k)x\,dx.$$

Then to the original boundary value problem we add

$$\begin{aligned} w' &= xF(u,x,k), \\ w(0) = 0, &\qquad w(1) = 0, \end{aligned}$$

and the trivial ordinary differential equation

$$k' = 0.$$

This augmented boundary value problem was solved successfully by [7] using *COLSYS*.

7.8 Interface Conditions

In problems involving layered media, for example, one has interface conditions at known interior points. Such a problem is

$$\begin{aligned} &\mathcal{L}\,u = 0, \quad x \in [0,\beta], \\ &\mathcal{L}\,u - \varphi_1^2 u = 0, \quad x \in [\beta,1], \end{aligned} \tag{7.14}$$

where

$$\mathcal{L}\,u = \frac{d^2u}{dx^2} + \frac{1}{x+\alpha}\frac{du}{dx},$$

and α and φ_1 are known constants, subject to

$$u(0) = 1,\ \frac{du}{dx}(1) = 0, \quad u(\beta^-) = u(\beta^+), \quad \frac{du}{dx}(\beta^-) = \frac{du}{dx}(\beta^+). \tag{7.15}$$

To obtain a standard boundary value problem, we map the problem (7.14)–(7.15) to $[0,1]$, by setting $z = x/\beta$ to map $[0,\beta]$ to $[0,1]$, and then $z = (1-x)/(1-\beta)$ maps $[\beta,1]$ to $[1,0]$. With

$$\mathcal{L}_0 u = \frac{d^2u}{dz^2} + \frac{\beta}{\beta^2+\alpha}\frac{du}{dz},$$

and

$$\mathcal{L}_1 u = \frac{d^2u}{dz^2} + \frac{\beta-1}{(\beta-1)^2+1+\alpha}\frac{du}{dz},$$

we obtain

$$\mathcal{L}_0 u_1 = 0 \qquad \mathcal{L}_1 u_2 - \varphi_1^2(\beta-1)^2 u_2 = 0 \tag{7.16}$$

and

$$\begin{aligned} &u_1(0) = 1, \quad \frac{du_2}{dz}(0) = 0, \\ &u_1(1) = u_2(1), \quad \frac{1}{\beta}\frac{du_1}{dz}(1) = \frac{1}{\beta-1}\frac{du_2}{dz}(1). \end{aligned} \tag{7.17}$$

An additional complication arises when the location of the interface, $x = \beta$, is unknown. Such a problem is discussed in [2], where in addition to (7.14), (7.15), we have

$$\mathcal{L}v - \varphi_2^2 u = 0, \quad x \in [\beta,1], \tag{7.18}$$

and the boundary conditions

$$v(1) = 0, \quad \frac{dv}{dx}(\beta) = 0, \quad v(1) = 1, \tag{7.19}$$

where φ_2 is a given constant. Equation (7.18) is transformed to

$$\mathcal{L}_1 v_1 - \varphi_2^2(\beta - 1)^2 u_2 = 0, \quad z \in I, \tag{7.20}$$

and (7.19) becomes

$$v_1(0) = 1, \quad v_1(1) = 0, \quad \frac{dv_1}{dz}(1) = 0. \tag{7.21}$$

Since β is unknown, to the boundary value problem consisting of (7.16), (7.17), (7.20) and (7.21), we add the trivial ordinary differential equation

$$\frac{d\beta}{dz} = 0, \quad z \in I.$$

8 Matrix Decomposition Algorithms for Poisson's Equation

8.1 Introduction

In this section, we consider the use of finite difference, finite element Galerkin and orthogonal spline collocation methods on uniform partitions for the solution of Poisson's equation in the unit square subject to Dirichlet boundary conditions:

$$\begin{aligned} -\Delta u &= f(x,y), \quad (x,y) \in \Omega, \\ u(x,y) &= 0, \quad (x,y) \in \partial\Omega, \end{aligned} \tag{8.1}$$

where Δ denotes the Laplacian and $\Omega = (0,1)^2$. Each method gives rise to a system of linear equations of the form

$$(A \otimes B + B \otimes A)\mathbf{u} = \mathbf{f}, \tag{8.2}$$

where A and B are square matrices of order M, say, $\mathbf{u}$ and $\mathbf{f}$ are vectors of order M^2 given by

$$\mathbf{u} = [u_{1,1}, \ldots, u_{1,M}, \ldots, u_{M,1}, \ldots, u_{M,M}]^T, \tag{8.3}$$

$$\mathbf{f} = [f_{1,1}, \ldots, f_{1,M}, \ldots, f_{M,1}, \ldots, f_{M,M}]^T, \tag{8.4}$$

and $\otimes$ denotes the tensor product; see Appendix D for the definition of $\otimes$ and its properties. A matrix decomposition algorithm is a fast direct method for solving systems of the form (8.2) which reduces the problem to one of solving a set of independent one-dimensional problems. We first develop a framework for matrix decomposition algorithms. To this end, suppose the real nonsingular matrix E is given and assume that a real diagonal matrix Λ and a real nonsingular matrix Z can be determined so that

$$AZ = BZ\Lambda \tag{8.5}$$

and

$$Z^T EBZ = I, \tag{8.6}$$

where I is the identity matrix of order M. Premultiplying (8.5) by $Z^T E$ and using (8.6), we obtain

$$Z^T EAZ = \Lambda. \tag{8.7}$$

The system of equations (8.2) can then be written in the form

$$(Z^T E \otimes I)(A \otimes B + B \otimes A)(Z \otimes I)(Z^{-1} \otimes I)\mathbf{u} = (Z^T E \otimes I)\mathbf{f}, \tag{8.8}$$

which becomes, on using the properties of $\otimes$, (8.6) and (8.7),

$$(\Lambda \otimes B + I \otimes A)(Z^{-1} \otimes I)\mathbf{u} = (Z^T E \otimes I)\mathbf{f}. \tag{8.9}$$

From the preceding, we obtain the following algorithm for solving (8.2):

MATRIX DECOMPOSITION ALGORITHM

1. Determine the matrices Λ and Z satisfying (8.5) and (8.6).
2. Compute $\mathbf{g} = (Z^T E \otimes I)\mathbf{f}$.
3. Solve $(\Lambda \otimes B + I \otimes A)\mathbf{v} = \mathbf{g}$.
4. Compute $\mathbf{u} = (Z \otimes I)\mathbf{v}$.

In the discretization methods considered in this section, the matrices Λ and Z are known explicitly. If $\Lambda = \text{diag}\{\lambda_i\}_{i=1}^M$, Step 2 reduces to a system of independent problems of the form

$$(A + \lambda_j B)\mathbf{v}_j = \mathbf{g}_j, \quad j = 1, \ldots, M.$$

It is shown that, for each method, these systems correspond to discretizations of two-point boundary value problems of the form

$$\begin{aligned} -u'' + \lambda u &= f(x), \quad x \in (0,1), \\ u(0) = u(1) &= 0, \end{aligned} \tag{8.10}$$

where λ is a positive constant. The elements of the matrix Z are sines and/or cosines and as a consequence multiplication by Z or Z^T can be done using fast Fourier transforms (FFTs). Moreover, the matrix E is sparse, so that multiplication by E can be done very efficiently. When all of these properties are exploited, the operation count for any of the matrix decomposition algorithms discussed in this section is $O(N^2 \log N)$, where $(N+1)h = 1$ and h is the mesh parameter.

8.2 Finite Difference Method

As a simple example, consider the basic five point difference approximation for Poisson's equation. To describe this method, suppose $h = 1/(N+1)$, where N is a positive integer, and set $x_m = mh, y_n = nh$. Denote by $u_{m,n}$ an approximation to $u(x_m, y_n)$ defined by the usual second order difference equations

$$-\frac{u_{m-1,n} - 2u_{m,n} + u_{m+1,n}}{h^2} - \frac{u_{m,n-1} - 2u_{m,n} + u_{m,n+1}}{h^2} = f(x_m, y_n), \quad m, n = 1, ..., N, \tag{8.11}$$

where $u_{0,n} = u_{N+1,n} = u_{m,0} = u_{m,N+1} = 0$. If we set $M = N$ and introduce vectors $\mathbf{u}$ and $\mathbf{f}$ as in (8.3) and (8.4), respectively, with $f_{m,n} = f(x_m, y_n)$, then the finite difference equations (8.11) may be written in the form

$$(A \otimes I + I \otimes A)\mathbf{u} = \mathbf{f}, \tag{8.12}$$

with $A = J$, where J is the tridiagonal matrix of order N given by (3.13). It is well known that (8.5) and (8.6) are satisfied if $B = E = I$,

$$\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_N), \tag{8.13}$$

where

$$\lambda_j = \frac{4}{h^2} \sin^2 \frac{j\pi h}{2}, \quad j = 1, \ldots, N, \tag{8.14}$$

and Z is the symmetric orthogonal matrix given by

$$Z = S, \tag{8.15}$$

where

$$S = \left(\frac{2}{N+1}\right)^{1/2} \left(\sin \frac{mn\pi}{N+1}\right)_{m,n=1}^{N}. \tag{8.16}$$

From the Matrix Decomposition Algorithm with Λ and Z defined by (8.13) and (8.15), respectively, we obtain the following matrix decomposition algorithm for solving (8.12).

FINITE DIFFERENCE ALGORITHM

1. Compute $\mathbf{g} = (S \otimes I)\mathbf{f}$.
2. Solve $(\Lambda \otimes I + I \otimes A)\mathbf{v} = \mathbf{g}$.
3. Compute $\mathbf{u} = (S \otimes I)\mathbf{v}$.

Note that Steps 1 and 3 can be carried out using FFTs at a cost of $O(N^2 \log N)$ operations. Step 2 consists of N tridiagonal linear systems each of which can be solved in $O(N)$ operations so that the total cost of the algorithm is $O(N^2 \log N)$ operations. Each tridiagonal system corresponds to the standard finite difference approximation to (8.10).

8.3 Finite Element Galerkin Methods with Piecewise Bilinear Elements

To obtain the weak form of the Poisson problem (8.1), let $H_0^1(\Omega)$ denote the space of all piecewise continuously differentiable functions defined on Ω which vanish on $\partial\Omega$. Then, for all $v \in H_0^1(\Omega)$, $u(x, y)$ satisfies

$$-\Delta u(x, y)v(x, y) = f(x, y)v(x, y),$$

and

$$-\int_\Omega \Delta u(x, y)v(x, y)dx\ dy = \int_\Omega f(x, y)v(x, y)dx\ dy. \tag{8.17}$$

On applying Green's formula to (8.17) and using the fact that $v = 0$ on $\partial\Omega$, we obtain

$$\int_\Omega \left[\frac{\partial u}{\partial x}\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\frac{\partial v}{\partial y}\right] dx\ dy = \int_\Omega f(x, y)v(x, y)dx\ dy. \tag{8.18}$$

For convenience, we introduce the notation

$$(g_1, g_2) = \int_\Omega g_1(x, y) \cdot g_2(x, y)dx\ dy$$

for functions g_1 and g_2 with the dot denoting scalar multiplication in the case of vector functions. Then (8.18) can be written in the form

$$(\nabla u, \nabla v) = (f, v) \quad \text{for all } v \in H_0^1(\Omega). \tag{8.19}$$

This is the weak form of (8.1) on which the finite element Galerkin method is based.

Now let $\{x_k\}_{k=0}^{N+1}$ be a uniform partition of the interval $[0, 1]$, so that $x_k = kh, k = 0, \ldots, N + 1$, where the stepsize $h = 1/(N + 1)$. Let $\{w_n\}_{n=1}^N$ denote the standard basis for the space of piecewise linear functions defined on this partition which vanish at 0 and 1; see (3.10). Then the C^0 piecewise bilinear Galerkin approximation

$$u_h(x, y) = \sum_{m=1}^{N}\sum_{n=1}^{N} u_{m,n}w_m(x)w_n(y)$$

to the solution u of (2.1) is obtained by requiring that

$$(\nabla u_h, \nabla v) = (f, v), \tag{8.20}$$

for all piecewise bilinear functions v which vanish on $\partial\Omega$. If $M = N$, and $\mathbf{u}$ and $\mathbf{f}$ are as in (8.3) and (8.4), respectively, with $f_{m,n} = (f, \phi_m\phi_n)$, then we obtain the linear system (8.2) in which the matrices A and B are given by (3.14). If $E = I$,

$$\Lambda = \operatorname{diag}\Big(\lambda_j/[1 - \tfrac{1}{6}\lambda_j]\Big)_{j=1}^{N}, \tag{8.21}$$

and

$$Z = S\ \operatorname{diag}\left(\Big\{h[1 - \tfrac{1}{6}\lambda_j]\Big\}^{-1/2}\right)_{j=1}^{N}, \tag{8.22}$$

where λ_j and S are given by (8.14) and (8.16), respectively, then (8.5) and (8.6) are satisfied. Thus, with Λ and Z defined by (8.21) and (8.22), respectively, we have the following matrix decomposition algorithm:

FINITE ELEMENT GALERKIN ALGORITHM

1. Compute $\mathbf{g} = (Z^T \otimes I)\mathbf{f}$.
2. Solve $(\Lambda \otimes B + I \otimes A)\mathbf{v} = \mathbf{g}$.
3. Compute $\mathbf{u} = (Z \otimes I)\mathbf{v}$.

As in the finite difference method, Step 2 involves the solution of N tridiagonal systems and consequently the computational cost of the algorithm is $O(N^2 \log N)$ operations. In this case, each tridiagonal system arises from the Galerkin approximation of (8.10).

8.4 Orthogonal Bicubic Spline Collocation Method

Again, let $\{x_k\}_{k=0}^{N+1}$ be a uniform partition of the interval $[0, 1]$, and let $\{\phi_n\}_{n=1}^{M}$, where $M = 2N + 2$ be a basis for $\mathcal{M}_1^{3,0}$ given by

$$\phi_n = v_n, \quad n = 1, \ldots, N, \quad \phi_{N+n+1} = s_n, \quad n = 0, \ldots, N + 1. \tag{8.23}$$

This is the standard basis but the ordering of the basis functions is nonstandard. The Hermite bicubic orthogonal collocation approximation

$$u_h(x,y) = \sum_{m=1}^{M}\sum_{n=1}^{M} u_{m,n}\phi_m(x)\phi_n(y)$$

to the solution u of (2.1) is obtained by requiring that

$$-\Delta u_h(\xi_m,\xi_n) = f(\xi_m,\xi_n), \qquad m,n = 1,\ldots,M. \tag{8.24}$$

With $\mathbf{u}$ and $\mathbf{f}$ as in (8.3) and (8.4), respectively, where $f_{m,n} = f(\xi_m,\xi_n)$, equations (8.24) can be written in the form (8.2) with

$$A = (a_{mn})_{m,n=1}^{M}, \qquad a_{mn} = -\phi_n''(\xi_m), \qquad B = (b_{mn})_{m,n=1}^{M}, \qquad b_{mn} = \phi_n(\xi_m). \tag{8.25}$$

Let

$$\Lambda = \mathrm{diag}(\lambda_1^-,\ldots,\lambda_N^-,\lambda_0,\lambda_1^+,\ldots,\lambda_N^+,\lambda_{N+1}), \tag{8.26}$$

where

$$\lambda_j^{\pm} = 12\left(\frac{8+\eta_j \pm \mu_j}{7-\eta_j}\right)h^{-2}, \qquad j = 1,\ldots,N, \qquad \lambda_0 = 36h^{-2}, \qquad \lambda_{N+1} = 12h^{-2},$$

and

$$\eta_j = \cos\left(\frac{j\pi}{N+1}\right), \qquad \mu_j = \sqrt{43+40\eta_j-2\eta_j^2}.$$

To describe the matrix Z, let $\Lambda_\alpha^{\pm}$, $\Lambda_\beta^{\pm}$ be diagonal matrices defined by

$$\Lambda_\alpha^{\pm} = \mathrm{diag}(\alpha_1^{\pm},\ldots,\alpha_N^{\pm}), \qquad \Lambda_\beta^{-} = \mathrm{diag}(\beta_1^{-},\ldots,\beta_N^{-}), \qquad \Lambda_\beta^{+} = \mathrm{diag}(1,\beta_1^{+},\ldots,\beta_N^{+},1/\sqrt{3}),$$

where

$$\alpha_j^{\pm} = (5+4\eta_j \mp \mu_j)\nu_j^{\pm}, \qquad \beta_j^{\pm} = 18\sin\left(\frac{j\pi}{N+1}\right)\nu_j^{\pm},$$

and

$$\nu_j^{\pm} = \left[27(1+\eta_j)(8+\eta_j\mp\mu_j)^2 + (1-\eta_j)(11+7\eta_j\mp 4\mu_j)^2\right]^{-1/2}.$$

Then

$$Z = 3\sqrt{3}\left[\begin{array}{c|c|c|c} S\Lambda_\alpha^- & \mathbf{0} & S\Lambda_\alpha^+ & \mathbf{0} \\ \hline \tilde{C}\Lambda_\beta^- & & C\Lambda_\beta^+ & \end{array}\right], \tag{8.27}$$

where $\mathbf{0}$ is the N-dimensional zero column vector, S is given by (8.16), and

$$C = \left(\frac{2}{N+1}\right)^{1/2} \left(\cos\frac{mn\pi}{N+1}\right)_{m,n=0}^{N+1}, \quad \tilde{C} = \left(\frac{2}{N+1}\right)^{1/2} \left(\cos\frac{mn\pi}{N+1}\right)_{m=0,n=1}^{N+1,N}.$$

It can be shown that if A, B, Λ, Z are given by (8.25), (8.26), (8.27), and $E = B^T$, then equations (8.5) and (8.6) are satisfied. Thus, from the Matrix Decomposition Algorithm, we obtain:

ORTHOGONAL SPLINE COLLOCATION ALGORITHM

1. Compute $\mathbf{g} = (Z^T B^T \otimes I)\mathbf{f}$.
2. Solve $(\Lambda \otimes B + I \otimes A)\mathbf{v} = \mathbf{g}$.
3. Compute $\mathbf{u} = (Z \otimes I)\mathbf{v}$.

Since there are at most four nonzero elements in each row of the matrix B^T the matrix-vector multiplications involving the matrix B^T in step 1 require a total of $O(N^2)$ arithmetic operations. From (8.27), it follows that FFT routines can be used to perform multiplications by the matrix Z^T in step 1 and by the matrix Z in step 3, the corresponding cost of each step being $O(N^2 \log N)$ operations. Step 2 involves the solution of M independent almost block diagonal linear systems with coefficient matrices of the form (4.7) arising from the orthogonal spline collocation approximation of (8.10), which can be solved in a total of $O(N^2)$ operations. Thus the total cost of this algorithm is also $O(N^2 \log N)$ operations.

References

1. S. Agmon, *Lectures on Elliptic Boundary–Value Problems*, Van Nostrand, Princeton, New Jersey, 1965.

2. A. Akyurtlu, J. F. Akyurtlu, C. E. Hamrin Jr. and G. Fairweather, *Reformulation and the numerical solution of the equations for a catalytic, porous wall, gas–liquid reactor*, Computers Chem. Eng., 10(1986), 361–365.

3. P. Amodio, J. R. Cash, G. Roussos, R. W. Wright, G. Fairweather, I. Gladwell, G. L. Kraut and M. Paprzycki, *Almost block diagonal linear systems: sequential and parallel solution techniques, and applications*, Numerical Linear Algebra with Applications, to appear.

4. E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov and D. Sorensen, *LAPACK Users' Guide*, SIAM Publications, Philadelphia, 1995.

5. U. Ascher and R. D. Russell, *Reformulation of boundary value problems into 'standard' form*, SIAM Rev., 23(1981), 238–254.

6. A. Aziz and T. Y. Na, *Squeezing flow of a viscous fluid between elliptic plates*, J. Comput. Appl. Math., 7(1981), 115–119.

7. D. Bhattacharyya, M. Jevtitch, J. T. Schrodt and G. Fairweather, *Prediction of membrane separation characteristics by pore distribution measurements and surface force–pore flow model*, Chem.Eng. Commun., 42(1986), 111–128.

8. C. de Boor, *A Practical Guide to Splines*, Applied Math. Sciences 27, Springer–Verlag, New York, 1978.

9. C. de Boor and R. Weiss, *SOLVEBLOK: A package for solving almost block diagonal linear systems*, ACM Trans. Math. Software, 6(1980), 80–87.

10. C. de Boor and R. Weiss, *Algorithm 546: SOLVEBLOK*, ACM Trans. Math. Software, 6(1980), 88–91.

11. J.C. Diaz, G. Fairweather and P. Keast *FORTRAN packages for solving certain almost block diagonal linear systems by modified alternate row and column elimination*, ACM Trans. Math. Software, 9(1983), 358–375.

12. J. C. Diaz, G. Fairweather and P. Keast, *Algorithm 603 COLROW and ARCECO: FORTRAN packages for solving certain almost block diagonal linear systems by modified alternate row and column elimination*, ACM Trans. Math. Software, 9(1983), 376–380.

13. J. Douglas Jr. and T. Dupont, *Collocation Methods for Parabolic Equations in a Single Space Variable*, Lecture Notes in Mathematics 385, Springer–Verlag, New York, 1974.

14. J. Douglas Jr., T. Dupont and L. Wahlbin, *Optimal L_∞ error estimates for Galerkin approximations to solutions of two–point boundary value problems* Math. Comp., 29(1975), 475–483.

15. K. S. Denison, C. E. Hamrin Jr. and G. Fairweather, *Solution of boundary value problems using software packages: DD04AD and COLSYS*, Chem. Eng. Commun., 22(1983), 1–9.

16. E. J. Doedel, *Finite difference collocation methods for nonlinear two point boundary value problems*, SIAM J. Numer. Anal., 16(1979), 173–185.

17. G. Fairweather, *Finite Element Galerkin Methods for Differential Equations*, Lecture Notes in Pure and Applied Mathematics, Vol. 34, Marcel Dekker, New York, 1978.

18. G. Fairweather and M. Vedha–Nayagam, *An assessment of numerical software for solving two–point boundary value problems arising in heat transfer*, Numerical Heat Transfer, 11(1987), 281-293.

19. H. B. Keller, *Numerical Methods for Two–Point Boundary Value Problems*, Dover, New York, 1992.

20. T. Y. Na and I. Pop, *Free convection flow past a vertical flat plate embedded in a saturated porous medium*, Int.J. Engrg. Sci., 21(1983), 517–526.

21. T. Y. Na and S. C. Tang, *A method for the solution of conduction heat transfer with non–linear heat generation*, Z. Angew. Math. Mech., 49(1969), 45–52.

22. R. S. Stepleman, *Tridiagonal fourth order approximations to general two–point nonlinear boundary value problems with mixed boundary conditions*, Math. Comp., 30(1976), 92–103.

23. J. M. Varah, *Alternate row and column elimination for solving certain linear systems*, SIAM J. Numer. Anal., 13(1976), 71–75.

24. M. F. Wheeler, *A Galerkin procedure for estimating the flux for two–point boundary value problems*, SIAM J. Numer. Anal., 11(1974), 764–768.

Appendix A. Galerkin Matrices for Piecewise Linear Functions. For the matrix, $A = ((w_i', w_j'))$:

$$\begin{aligned}
(w_i', w_i') &= \int_0^1 [w_i'(x)]^2\,dx \\
&= \int_{x_{i-1}}^{x_i} \frac{1}{(x_i - x_{i-1})^2}\,dx + \int_{x_i}^{x_{i+1}} \frac{1}{(x_{i+1} - x_i)^2}\,dx \\
&= \left[\frac{1}{(x_i - x_{i-1})} + \frac{1}{(x_{i+1} - x_i)}\right] \\
&= \frac{1}{h_i} + \frac{1}{h_{i+1}}; \\
(w_i', w_{i+1}') &= \int_{x_i}^{x_{i+1}} \frac{-1}{(x_{i+1} - x_i)^2}\,dx = -\frac{1}{(x_{i+1} - x_i)} = -\frac{1}{h_{i+1}}; \\
(w_i', w_{i-1}') &= -\frac{1}{h_i}.
\end{aligned}$$

For the matrix $B = ((w_i, w_j))$:

$$\begin{aligned}
(w_i, w_i) &= \int_{x_{i-1}}^{x_i} \left(\frac{x - x_{i-1}}{x_i - x_{i-1}}\right)^2 dx + \int_{x_i}^{x_{i+1}} \left(\frac{x - x_i}{x_i - x_{i+1}}\right)^2 dx \\
&= \frac{1}{3}[h_i + h_{i+1}]; \\
(w_i, w_{i+1}) &= \int_{x_i}^{x_{i+1}} \left(\frac{x_{i+1} - x}{x_{i+1} - x_i}\right)\left(\frac{x - x_i}{x_{i+1} - x_i}\right) dx \\
&= \frac{1}{h_{i+1}^2}\{\frac{1}{2}(x_{i+1} - x)(x - x_i)^2|_{x_i}^{x_{i+1}} + \frac{1}{2}\int_{x_i}^{x_{i+1}} (x - x_i)^2\,dx\} \\
&= \frac{1}{6}h_{i+1}.
\end{aligned}$$

Similarly,

$$(w_i, w_{i-1}) = \frac{1}{6}h_i.$$

Appendix B. Galerkin Matrices for Piecewise Hermite Cubic Functions. We consider the case in which the partition π of $\overline{I}$ is uniform and let $h = x_{i+1} - x_i$, $i = 0, \ldots, N$. The conditioning of the matrices is improved if the slope functions are normalized by dividing by h. Thus we define

$$\tilde{s}_i(x) = h^{-1} s_i(x), \quad i = 0, \ldots, N+1.$$

If the Galerkin solution u_h is expressed in the form

$$u_h(x) = \sum_{i=0}^{N+1} \{\alpha_i^{(1)} v_i(x) + \alpha_i^{(2)} \tilde{s}_i(x)\},$$

then

$$\alpha_i^{(1)} = u_h(x_i),$$

as before, but now

$$\alpha_i^{(2)} = h u_h'(x_i), \quad i = 0, \ldots, N+1.$$

If A and B are partitioned as in (3.16) then, with the normalization of the slope functions, we have, if $A = ((w_i', w_j')) \equiv (A_{ij})$ and $B = ((w_i, w_j)) \equiv (B_{ij})$,

$$\begin{aligned}
A_{00} &= (30h)^{-1} \begin{bmatrix} 36 & 3 \\ 3 & 4 \end{bmatrix}, \\
A_{ii} &= (15h)^{-1} \begin{bmatrix} 36 & 0 \\ 0 & 4 \end{bmatrix}, \quad i = 1, \ldots, N, \\
A_{N+1,N+1} &= (30h)^{-1} \begin{bmatrix} 36 & -3 \\ -3 & 4 \end{bmatrix}, \\
A_{i,i+1} &= A_{i+1,i}^T = (30h)^{-1} \begin{bmatrix} -36 & 3 \\ -3 & -1 \end{bmatrix}, \quad i = 0, \ldots, N,
\end{aligned}$$

and

$$\begin{aligned}
B_{00} &= \frac{h}{210} \begin{bmatrix} 78 & 11 \\ 11 & 2 \end{bmatrix}, \\
B_{ii} &= \frac{h}{105} \begin{bmatrix} 78 & 0 \\ 0 & 2 \end{bmatrix}, \quad i = 1, \ldots, N,
\end{aligned}$$

$$
\begin{aligned}
B_{N+1,N+1} &= \frac{h}{210}\begin{bmatrix} 78 & -11 \\ -11 & 2 \end{bmatrix}, \\
B_{i,i+1} &= B_{i+1,i}^T = \frac{h}{420}\begin{bmatrix} 54 & -13 \\ 13 & -3 \end{bmatrix}, \quad i = 0, \ldots, N.
\end{aligned}
$$

Appendix C. Elements of Piecewise Hermite Cubic Orthogonal Spline Collocation Matrices. On $[x_i, x_{i+1}]$, with $x_i = ih$,

$$
\begin{aligned}
v_i(x) &= -2\left[\frac{x_{i+1}-x}{h}\right]^3 + 3\left[\frac{x_{i+1}-x}{h}\right]^2, \\
v_i'(x) &= \frac{6}{h^3}(x_{i+1}-x)^2 - \frac{6}{h^2}(x_{i+1}-x), \\
v_i''(x) &= -\frac{12}{h^3}(x_{i+1}-x) + \frac{6}{h^2}.
\end{aligned}
$$

From (4.5), $\xi_{2i+1} = x_i + \frac{1}{2}h(1+\rho_1)$, where $\rho_1 = -1/\sqrt{3}$. Then, if $x = \xi_{2i+1}$

$$
x_{i+1} - x = \frac{1}{2}h(1-\rho_1)
$$

and

$$
\begin{aligned}
v_i(\xi_{2i+1}) &= -\frac{1}{4}(1-\rho_1)^3 + \frac{3}{4}(1-\rho_1)^2 \\
&= \frac{1}{4}(1-\rho_1)^2(2+\rho_1) \\
v_i'(\xi_{2i+1}) &= \frac{3}{2h}(1-\rho_1)^2 - \frac{3}{h}(1-\rho_1) \\
&= -\frac{3}{2h}(1-\rho_1^2) \\
v_i''(\xi_{2i+1}) &= -\frac{6}{h^2}(1-\rho_1) + \frac{6}{h^2} = \frac{6\rho_1}{h^2}
\end{aligned}
$$

Similar expressions hold for $v_i(\xi_{2i+2})$, $v_i'(\xi_{2i+2})$, $v_i''(\xi_{2i+2})$, where $\xi_{2i+2} = x_i + \frac{1}{2}h(1+\rho_2)$, $\rho_2 = -\rho_1$.

On $[x_i, x_{i+1}]$,

$$
\begin{aligned}
v_{i+1}(x) &= -2\left[\frac{x-x_i}{h}\right]^3 + 3\left[\frac{x-x_i}{h}\right]^2, \\
v_{i+1}'(x) &= -\frac{6}{h^3}(x-x_i)^2 + \frac{6}{h^2}(x-x_i) \\
v_{i+1}''(x) &= -\frac{12}{h^3}(x-x_i) + \frac{6}{h^2}.
\end{aligned}
$$

If $x = \xi_{2i+1}$,

$$x - x_i = \frac{1}{2}h(1+\rho_1).$$

Therefore,

$$\begin{aligned}
v_{i+1}(\xi_{2i+1}) &= -\frac{1}{4}(1+\rho_1)^3 + \frac{3}{4}(1+\rho_1)^2 \\
&= \frac{1}{4}(1+\rho_1)^2(2-\rho_1), \\
v'_{i+1}(\xi_{2i+1}) &= -\frac{3}{2h}(1+\rho_1)^2 + \frac{3}{h}(1+\rho_1) \\
&= \frac{3}{2h}(1-\rho_1^2), \\
v''_{i+1}(\xi_{2i+1}) &= -\frac{6}{h^2}(1+\rho_1) + \frac{6}{h_{i+1}^2} \\
&= -\frac{6\rho_1}{h^2},
\end{aligned}$$

with similar expressions for $v_{i+1}(\xi_{2i+2})$, $v'_{i+1}(\xi_{2i+2})$, $v''_{i+1}(\xi_{2i+2})$.

With $\tilde{s}_i = h^{-1}s_i$ as in Appendix B, on $[x_i, x_{i+1}]$,

$$\begin{aligned}
\tilde{s}_i(x) &= -\left\{\left[\frac{x_{i+1}-x}{h}\right]^3 - \left[\frac{x_{i+1}-x}{h}\right]^2\right\}, \\
\tilde{s}'_i(x) &= -\left\{-\frac{3}{h^3}(x_{i+1}-x)^2 + \frac{2}{h^2}(x_{i+1}-x)\right\}, \\
\tilde{s}''_i(x) &= -\left\{\frac{6}{h^3}(x_{i+1}-x) - \frac{2}{h^2}\right\}.
\end{aligned}$$

With $x_{i+1} - x = \frac{1}{2}h(1-\rho_1)$, we obtain

$$\begin{aligned}
\tilde{s}_i(\xi_{2i+1}) &= -\left\{\frac{1}{8}(1-\rho_1)^3 - \frac{1}{4}(1-\rho_1)^2\right\} \\
&= -\frac{1}{8}(1-\rho_1)^2\{1-\rho_1-2\} \\
&= \frac{1}{8}(1-\rho_1)^2(1+\rho_1), \\
\tilde{s}'_i(\xi_{2i+1}) &= -\left\{-\frac{3}{h}\frac{(1-\rho_1)^2}{4} + \frac{1}{h}(1-\rho_1)\right\} \\
&= -\frac{1}{4h}(1-\rho_1)\{-3(1-\rho_1)+4\}
\end{aligned}$$

$$
\begin{aligned}
&= -\frac{1}{4h}(1-\rho_1)(1+3\rho_1),\\
\tilde{s}_i''(\xi_{2i+1}) &= -\left\{\frac{3}{h^2}(1-\rho_1)-\frac{2}{h^2}\right\}\\
&= -\frac{1}{h^2}(1-3\rho_1),
\end{aligned}
$$

with similar expressions for $\tilde{s}_i(\xi_{2i+2})$, $\tilde{s}_i'(\xi_{2i+2})$, and $\tilde{s}_i''(\xi_{2i+2})$.

On $[x_i, x_{i+1}]$,

$$
\begin{aligned}
\tilde{s}_{i+1}(x) &= \left[\frac{x-x_i}{h}\right]^3 - \left[\frac{x-x_i}{h}\right]^2,\\
\tilde{s}_{i+1}'(x) &= \frac{3}{h^3}(x-x_i)^2 - \frac{2}{h^2}(x-x_i),\\
\tilde{s}_{i+1}''(x) &= \frac{6}{h^3}(x-x_i) - \frac{2}{h^2}.
\end{aligned}
$$

Then, with $x - x_i = \frac{1}{2}h(1+\rho_1)$, we have

$$
\begin{aligned}
\tilde{s}_{i+1}(\xi_{2i+1}) &= \left\{\frac{1}{8}(1+\rho_1)^3 - \frac{1}{4}(1+\rho_1)^2\right\}\\
&= -\frac{1}{8}(1+\rho_1)^2(1-\rho_1),\\
\tilde{s}_{i+1}'(\xi_{2i+1}) &= \left\{\frac{3}{4h}(1+\rho_1)^2 - \frac{1}{h}(1+\rho_1)\right\}\\
&= -\frac{1}{4h}(1+\rho_1)(1-3\rho_1),\\
\tilde{s}_{i+1}''(\xi_{2i+1}) &= \frac{1}{h^2}\{3(1+\rho_1)-2\}\\
&= \frac{1}{h^2}(1+3\rho_1),
\end{aligned}
$$

with similar expressions for $\tilde{s}_{i+1}(\xi_{2i+2})$, $\tilde{s}_{i+1}'(\xi_{2i+2})$, $\tilde{s}_{i+1}''(\xi_{2i+2})$.

Appendix D. Properties of the Tensor Product. If $A = (a_{ij})$ is an $M \times M$ matrix and B is an $N \times N$ matrix, then the tensor product of A and B, $A \otimes B$, is the $MN \times MN$ block matrix whose (i,j) element is $a_{ij}B$. The tensor product has the following properties:

$$
\begin{aligned}
A \otimes (B+C) &= A \otimes B + A \otimes C;\\
(B+C) \otimes A &= B \otimes A + C \otimes A;\\
(A \otimes B)(C \otimes D) &= AC \otimes BD, \text{ provided the matrix products are defined.}
\end{aligned}
$$