# 3

# Boundary-Value Problems for Ordinary Differential Equations: Finite Element Methods

## INTRODUCTION

The numerical techniques outlined in this chapter produce approximate solutions that, in contrast to those produced by finite difference methods, are continuous over the interval. The approximate solutions are piecewise polynomials, thus qualifying the techniques to be classified as finite element methods [1]. Here, we discuss two types of finite element methods: collocation and Galerkin.

## BACKGROUND

Let us begin by illustrating finite elements methods with the following BVP:

$$y'' = y + f(x), \qquad 0 < x < 1 \tag{3.1a}$$

$$\begin{aligned} y(0) &= 0 \\ y(1) &= 0 \end{aligned} \tag{3.1b}$$

Finite element methods find a piecewise polynomial (pp) approximation, $u(x)$, to the solution of (3.1). A piecewise polynomial is a function defined on a partition such that on the subintervals defined by the partition, it is a polynomial. The pp-approximation can be represented by

$$u(x) = \sum_{j=1}^{m} a_j \phi_j(x) \tag{3.2}$$

where $\{\phi_j(x)|j=1, \ldots, m\}$ are specified functions that are piecewise continuously differentiable, and $\{a_j|j=1, \ldots, m\}$ are as yet unknown constants. For now, assume that the functions $\phi_j(x)$, henceforth called basis functions (to be explained in the next section), satisfy the boundary conditions. The finite element methods differ only in how the unknown coefficients $\{a_j|j=1, \ldots, m\}$ are determined.

In the collocation method, the set $\{a_j|j=1, \ldots, m\}$ is determined by satisfying the BVP exactly at $m$ points, $\{x_i|i=1, \ldots, m\}$, the collocation points, in the interval. For (3.1):

$$u''(x_i) - u(x_i) - f(x_i) = 0, \qquad i = 1, \ldots, m \tag{3.3}$$

If $u(x)$ is given by (3.2), then (3.3) becomes

$$\sum_{j=1}^{m} a_j[\phi_j''(x_i) - \phi_j(x_i)] - f(x_i) = 0, \qquad i = 1, \ldots, m \tag{3.4}$$

or in matrix notation,

$$A^C\mathbf{a} = \mathbf{f} \tag{3.5}$$

where

$$A_{ij}^C = \phi_j''(x_i) - \phi_j(x_i)$$

$$\mathbf{a} = [a_1, a_2, \ldots, a_m]^T$$

$$\mathbf{f} = [f(x_1), f(x_2), \ldots, f(x_m)]^T$$

The solution of (3.5) then yields the vector $\mathbf{a}$, which determines the collocation approximation (3.2).

To formulate the Galerkin method, first multiply (3.1) by $\phi_i$ and integrate the resulting equation over [0, 1]:

$$\int_0^1 [y''(x) - y(x) - f(x)]\phi_i(x)\,dx = 0, \qquad i = 1, \ldots, m \tag{3.6}$$

Integration of $y''(x)\phi_i(x)$ by parts gives

$$\int_0^1 y''(x)\phi_i(x)\,dx = y'(x)\phi_i(x)\Big|_0^1 - \int_0^1 y'(x)\phi_i'(x)\,dx, \qquad i = 1, \ldots, m$$

Since the functions $\phi_i(x)$ satisfy the boundary conditions, (3.6) becomes

$$\int_0^1 y'(x)\phi_i'(x)\,dx + \int_0^1 [y(x) + f(x)]\phi_i(x)\,dx = 0, \qquad i = 1, \ldots, m \tag{3.7}$$

For any two functions $\eta$ and $\psi$ we define

$$(\eta, \psi) = \int_0^1 \eta(x)\psi(x)\,dx \tag{3.8}$$

With (3.8), Eq. (3.7) becomes

$$(y', \phi_i') + (y, \phi_i) + (f, \phi_i) = 0, \qquad i = 1, \ldots, m \tag{3.9}$$

and is called the weak form of (3.1). The Galerkin method consists of finding $u(x)$ such that

$$(u', \phi_i') + (u, \phi_i) + (f, \phi_i) = 0, \qquad i = 1, \ldots, m \tag{3.10}$$

If $u(x)$ is given by (3.2), then (3.10) becomes:

$$\left(\sum_{j=1}^{m} a_j \phi_j', \phi_i'\right) + \left(\sum_{j=1}^{m} a_j \phi_j, \phi_i\right) + (f, \phi_i) = 0, \qquad i = 1, \ldots, m \tag{3.11}$$

or, in matrix notation,

$$A^G \mathbf{a} = -\mathbf{g} \tag{3.12}$$

where

$$A_{ij}^G = (\phi_j', \phi_i') + (\phi_j, \phi_i)$$
$$\mathbf{a} = [a_1, \ldots, a_m]^T$$
$$\mathbf{g} = [\bar{f}_1, \ldots, \bar{f}_m]^T$$
$$\bar{f}_i = (f, \phi_i)$$

The solution of (3.12) gives the vector $\mathbf{a}$, which specifies the Galerkin approximation (3.2).

Before discussing these methods in further detail, we consider choices of the basis functions.

## PIECEWISE POLYNOMIAL FUNCTIONS

To begin the discussion of pp-functions, let the interval partition $\pi$ be given by:

$$a = x_1 < x_2 < \ldots < x_{\ell+1} = b \tag{3.13}$$

with

$$h = \max_{1 \leq j \leq \ell} h_j = \max_{1 \leq j \leq \ell} (x_{j+1} - x_j)$$

Also let $\{P_j(x) | j = 1, \ldots, \ell\}$ be any sequence of $\ell$ polynomials of order $k$ (degree $\leq k - 1$). The corresponding pp-function, $F(x)$, of order $k$ is defined by

$$F(x) = P_j(x), \qquad x_j < x < x_{j+1} \tag{3.14}$$
$$j = 1, \ldots, \ell$$

where $x_j$ are called the breakpoints of $F$. By convention

$$F(x) = \begin{cases} P_1(x), & x \leq x_1 \\ P_\ell(x), & x \geq x_{\ell+1} \end{cases} \tag{3.15}$$

and

$$F(x_i) = P_i(x_i) \qquad \text{(right continuity)} \tag{3.16}$$

A portion of a pp-function is illustrated in Figure 3.1. The problem is how to conveniently represent the pp-function.

Let $S$ be a set of functions:

$$S = \{\lambda_j(x) | j = 1, \ldots, L\} \tag{3.17}$$

The class of functions denoted by $\mathscr{P}$ is defined to be the set of all functions $f(x)$ of the form

$$f(x) = \sum_{j=1}^{L} \alpha_j \lambda_j(x) \tag{3.18}$$

where the $\alpha_j$'s are constants. This class of functions $\mathscr{P}$ defined by (3.18) is called a linear function space. This is analogous to a linear vector space, for if vectors $\mathbf{x}_j$ are substituted for the functions $\lambda_j(x)$ in (3.18), we have the usual definition of an element $\mathbf{x}$ of a vector space. If the functions $\lambda_j$ in $S$ are linearly independent, then the set $S$ is called a basis for the space $\mathscr{P}$, $L$ is the dimension of the space $\mathscr{P}$, and each function $\lambda_j$ is called a basis function.

Define $\mathscr{P}_k(\pi)$ (subspace of $\mathscr{P}$) to be the set of all pp-functions of order $k$ on the partition $\pi$. The dimension of this space is

$$\dim \mathscr{P}_k(\pi) = k\ell \tag{3.19}$$

Let $\boldsymbol{\nu}$ be a sequence of nonnegative integers $\nu_j$, that is, $\boldsymbol{\nu} = \{\nu_j | j = 2, \ldots, \ell\}$, such that

$$\text{jump}_{x_j} \frac{d^{i-1}}{dx^{i-1}} [f(x)] = 0 \tag{3.20}$$

$$i = 1, \ldots, \nu_j, \qquad j = 2, \ldots, \ell$$

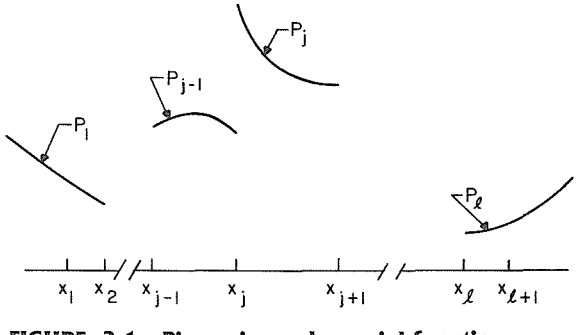


**FIGURE 3.1 Piecewise polynomial function.**

where

$$\text{jump}_{x_j} \frac{d^{i-1}}{dx^{i-1}}[f(x)] = \frac{d^{i-1}}{dx^{i-1}}[f(x_j^+)] - \frac{d^{i-1}}{dx^{i-1}}[f(x_j^-)] \tag{3.21}$$

or in other words, $\nu$ specifies the continuity (if any) of the function and its derivative at the breakpoints. Define the subspace $\mathscr{P}_k^{\nu}(\pi)$ of $\mathscr{P}_k(\pi)$ by

$$\mathscr{P}_k^{\nu}(\pi) = \left\{ \begin{matrix} f(x) \text{ is in } \mathscr{P}_k(\pi) \text{ and satisfies the jump} \\ \text{conditions specified by } \nu \end{matrix} \right\} \tag{3.22}$$

The dimension of the space $\mathscr{P}_k^{\nu}(\pi)$ is

$$\dim \mathscr{P}_k^{\nu}(\pi) = \sum_{j=1}^{\ell} (k - \nu_j) \tag{3.23}$$

where $\nu_1 = 0$.

We now have a space, $\mathscr{P}_k^{\nu}(\pi)$, that can contain pp-functions such as $F(x)$. Since the $\lambda_i$'s can be a basis for $\mathscr{P}_k^{\nu}(\pi)$, then $F(x)$ can be represented by (3.18). Next, we illustrate various spaces $\mathscr{P}_k^{\nu}(\pi)$ and bases for these spaces. When using $\mathscr{P}_k^{\nu}(\pi)$ as an approximating space for solving differential equations by finite element methods, we will not use variable continuity throughout the interval. Therefore, notationally replace $\nu$ by $\nu$, where $\{\nu_j = \nu | j = 2, \ldots, \ell\}$.

The simplest space is $\mathscr{P}_2^1(\pi)$, the space of piecewise linear functions. A basis for this space consists of straight-line segments (degree = 1) with discontinuous derivatives at the breakpoints ($\nu = 1$). This basis is given in Table 3.1 and is shown in Figure 3.2*a*. Notice that the dimension of $\mathscr{P}_2^1(\pi)$ is $\ell + 1$ and that there are $\ell + 1$ basis functions given in Table 3.1. Thus, (3.18) can be written as

$$f(x) = \sum_{j=1}^{\ell+1} \alpha_j w_j \tag{3.24}$$

**TABLE 3.1 Linear Basis Functions**

$$w_1 = \begin{cases} \dfrac{x_2 - x}{x_2 - x_1}, & \text{for } x_1 \leq x \leq x_2 \\ 0, & \text{for } x \geq x_2 \end{cases}$$

$$w_j = \begin{cases} \dfrac{x - x_{j-1}}{x_j - x_{j-1}}, & \text{for } x_{j-1} \leq x \leq x_j \\ \dfrac{x_{j+1} - x}{x_{j+1} - x_j}, & \text{for } x_j \leq x \leq x_{j+1} \\ 0, & \text{for } x_{j-1} \leq x,\ x \geq x_{j+1} \end{cases}$$

$$w_{\ell+1} = \begin{cases} 0, & \text{for } x \leq x_\ell \\ \dfrac{x - x_\ell}{x_{\ell+1} - x_\ell}, & \text{for } x_\ell \leq x \leq x_{\ell+1} \end{cases}$$

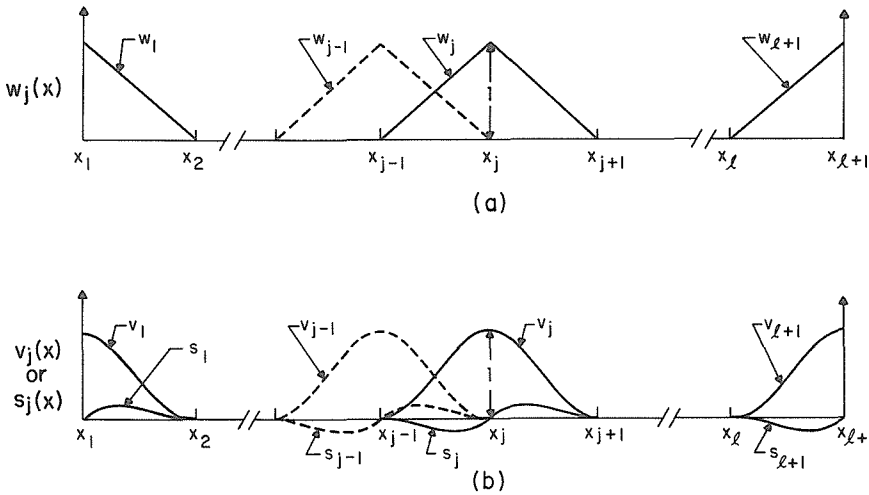


**FIGURE 3.2 Schematic of basis functions. (*a*) Piecewise linear functions. (*b*) Piecewise hermite cubic functions.**

Frequently, one is interested in numerical approximations that have continuity of derivatives at the interior breakpoints. Obviously, $\mathscr{P}_2^1(\pi)$ does not possess this property, so one must resort to a high-order space.

A space possessing continuity of the first derivative is the Hermite cubic space, $\mathscr{P}_4^2(\pi)$. A basis for this space is the "value" $v_j$ and the "slope" $s_j$ functions given in Table 3.2 and shown in Figure 3.2*b*. Some important properties of this basis are

$$
\begin{aligned}
v_j &= \begin{cases} 0 & \text{at all } x_i \neq x_j \\ 1 & \text{at } x_j \end{cases} \\
\frac{dv_j}{dx} &= 0 \quad \text{at all } x_i \\
s_j &= 0 \quad \text{at all } x_i \\
\frac{ds_j}{dx} &= \begin{cases} 0 & \text{at all } x_i \neq x_j \\ 1 & \text{at } x_j \end{cases}
\end{aligned}
\tag{3.25}
$$

The dimension of this space is $2(\ell + 1)$; thus (3.18) can be written as

$$
f(x) = \sum_{j=1}^{\ell+1} [\alpha_j^{(1)} v_j + \alpha_j^{(2)} s_j] \tag{3.26}
$$

where $\alpha_j^{(1)}$ and $\alpha_j^{(2)}$ are constants. Since $\nu = 2$, $f(x)$ is continuous and also possesses a continuous first derivative. Notice that because of the properties

**TABLE 3.2 Hermite Cubic Basis Functions**

$h_j = x_{j+1} - x_j$ $\qquad g_1(x) = -2x^3 + 3x^2, \quad 0 \leq x \leq 1$

$\xi_j(x) = \dfrac{x - x_j}{h_j}$ $\qquad g_2(x) = x^3 - x^2, \quad 0 \leq x \leq 1$

| Value Functions | Slope Functions |
|---|---|
| $v_1 = \begin{cases} g_1(1-\xi_1(x)), & 0 \leq x \leq x_2 \\ 0, & x_2 \leq x \leq 1 \end{cases}$ | $s_1 = \begin{cases} -h_1 g_2(1-\xi_1(x)), & 0 \leq x \leq x_2 \\ 0, & x_2 \leq x \leq 1 \end{cases}$ |
| $v_j = \begin{cases} g_1(\xi_{j-1}(x)), & x_{j-1} \leq x \leq x_j \\ g_1(1-\xi_j(x)), & x_j \leq x \leq x_{j+1} \\ 0, & 0 \leq x \leq x_{j-1}, x_{j+1} \leq x \leq 1 \end{cases}$ | $s_j = \begin{cases} h_{j-1} g_2(\xi_{j-1}(x)), & x_{j-1} \leq x \leq x_j \\ -h_j g_2(1-\xi_j(x)), & x_j \leq x \leq x_{j+1} \\ 0, & 0 \leq x \leq x_{j-1}, x_{j+1} \leq x \leq 1 \end{cases}$ |
| $v_{\ell+1} = \begin{cases} 0, & 0 \leq x \leq x_\ell \\ g_1(\xi_\ell(x)), & x_\ell \leq x \leq 1 \end{cases}$ | $s_{\ell+1} = \begin{cases} 0, & 0 \leq x \leq x_\ell \\ h_\ell g_2(\xi_\ell(x)), & x_\ell \leq x \leq 1 \end{cases}$ |

shown in (3.25) the vector

$$\alpha^{(1)} = [\alpha_1^{(1)}, \alpha_2^{(1)}, \ldots, \alpha_{\ell+1}^{(1)}]^T$$

give the values of $f(x_i)$, $i = 1, \ldots, \ell + 1$ while the vector

$$\alpha^{(2)} = [\alpha_1^{(2)}, \alpha_2^{(2)}, \ldots, \alpha_{\ell+1}^{(2)}]^T$$

gives the values of $df(x_i)/dx$, $i = 1, \ldots, \ell + 1$. Also, notice that the Hermite cubic as well as the linear basis have limited or local support on the interval; that is, they are nonzero over a small portion of the interval.

A suitable basis for $\mathscr{P}_k^{\nu}(\pi)$ given any $\nu$, $k$, and $\pi$ is the B-spline basis [2]. Since this basis does not have a simple representation like the linear or Hermite cubic basis, we refer the reader to Appendix D for more details on B-splines. Here, we denote the B-spline basis functions by $B_j(x)$ and write (3.18) as:

$$f(x) = \sum_{j=1}^{N} \alpha_j B_j(x) \tag{3.27}$$

where

$$N = \dim \mathscr{P}_k^{\nu}(\pi)$$

Important properties of the $B_j$'s are:

1. They have local support.
2. $B_1(a) = 1$, $B_N(b) = 1$.
3. Each $B_j(x)$ satisfies $0 \leq B_j(x) \leq 1$ (normalized B-splines).
4. $\sum_{j=1}^{N} B_j(x) = 1$ for $a \leq x \leq b$.

## THE GALERKIN METHOD

Consider (3.1) and its weak form (3.9). The use of (3.2) in (3.9) produces the matrix problem (3.12). Since the basis $\phi_i$ is local, the matrix $A^G$ is sparse.

### EXAMPLE 1

Set up the matrix problem for

$$\begin{aligned} -y''(x) &= 1, \qquad 0 < x < 1, \\ y(0) &= 0 \\ y(1) &= 0 \end{aligned}$$

using $\mathscr{P}\frac{1}{2}(\pi)$ as the approximating space.

*SOLUTION*

Using $\mathscr{P}\frac{1}{2}(\pi)$ gives

$$u(x) = \sum_{j=1}^{\ell+1} a_j w_j$$

Since we have imposed the condition that the basis functions satisfy the boundary conditions, the first and last basis function given in Table 3.1 are excluded. Therefore, the pp-approximation is given by

$$u(x) = \sum_{j=1}^{\ell-1} a_j w_j$$

where the $w_j$'s are as shown in Figure 3.3. The matrix $A^G$ is given by

$$A_{ij}^G = \int_0^1 \phi_j' \phi_i' \, dx$$

Because each basis function is supported on only two subintervals [see Figure 3.2(a)],

$$A_{ij}^G = 0 \quad \text{if} \quad |i - j| > 1$$

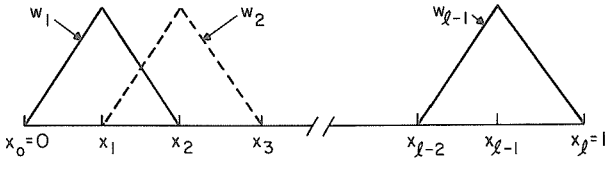


**FIGURE 3.3 Numbering of basis functions for Example 1.**

Thus, $A^G$ is tridiagonal and

$$A_{ii}^G = \int_0^1 (\phi_i')^2 dx = \int_{x_{i-1}}^{x_i} \left[\frac{1}{x_i - x_{i-1}}\right]^2 dx + \int_{x_i}^{x_{i+1}} \left[\frac{-1}{x_{i+1} - x_i}\right]^2 dx$$

$$= \frac{1}{h_i} + \frac{1}{h_{i+1}}, \qquad h_i = x_i - x_{i-1}$$

$$A_{i,i+1}^G = \int_0^1 \phi_i' \phi_{i+1}' dx = \int_{x_i}^{x_{i+1}} \left[\frac{-1}{x_{i+1} - x_i}\right]\left[\frac{1}{x_{i+1} - x_i}\right] dx$$

$$= -\frac{1}{h_{i+1}}$$

$$A_{i,i-1}^G = -\frac{1}{h_i}$$

The vector $\mathbf{g}$ is given by

$$\bar{f}_i = \int_0^1 f(x)\phi_i(x)\,dx = \int_0^1 \phi_i(x)\,dx = \tfrac{1}{2}[h_i + h_{i+1}]$$

Therefore, the matrix problem is:

$$\begin{bmatrix} \left(\frac{1}{h_1} + \frac{1}{h_2}\right) & -\frac{1}{h_2} & & & 0 \\ -\frac{1}{h_2} & \left(\frac{1}{h_2} + \frac{1}{h_3}\right) & -\frac{1}{h_3} & & \\ & \ddots & \ddots & \ddots & \\ & & & -\frac{1}{h_{\ell-2}} & \\ & & -\frac{1}{h_{\ell-2}} & \left(\frac{1}{h_{\ell-1}}\right) & \\ 0 & & & & \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_{\ell-2} \\ a_{\ell-1} \end{bmatrix} = \begin{bmatrix} \frac{1}{2}(h_1 + h_2) \\ \frac{1}{2}(h_2 + h_3) \\ \vdots \\ \frac{1}{2}(h_{\ell-2} + h_{\ell-1}) \end{bmatrix}$$

From Example 1, one can see that if a uniform mesh is specified using $\mathscr{P}_2^1(\pi)$, the standard second-order correct finite difference method is obtained. Therefore, the method would be second-order accurate. In general, the Galerkin method using $\mathscr{P}_k^\nu(\pi)$ gives an error such that [1]:

$$\|y - u\| \leq Ch^k \tag{3.28}$$

where

$$\begin{aligned} y &= \text{true solution} \\ u &= \text{pp-approximation} \\ C &= \text{a constant} \\ h &= \text{uniform partition} \\ \|Q\| &= \max_x |Q| \end{aligned}$$

provided that $y$ is sufficently smooth. Obviously, one can increase the accuracy by choosing the approximating space to be of higher order.

EXAMPLE 2

An insulated metal rod is exposed at each end to a temperature, $T_0$. Within the rod, heat is generated according to the following function:

$$\xi[(T - T_0) + \cosh(1)]$$

where

$$\begin{aligned} \xi &= \text{constant} \\ T &= \text{absolute temperature} \end{aligned}$$

The rod is illustrated in Figure 3.4. The temperature profile in the rod can be calculated by solving the following energy balance:

$$\begin{aligned} K\frac{d^2T}{dz^2} &= \xi[(T - T_0) + \cosh(1)] \\ T &= T_0 \quad \text{at} \quad z = 0 \\ T &= T_0 \quad \text{at} \quad z = L \end{aligned} \tag{3.29}$$

where $K$ is the thermal conductivity of the metal. When $(\xi L^2)/K = 4$, the solution of the BVP is

$$y = \cosh(2x - 1) - \cosh(1)$$

where $y = T - T_0$ and $x = z/L$. Solve (3.29) using the Hermite cubic basis, and show that the order of accuracy is $0(h^4)$ (as expected from 3.28).

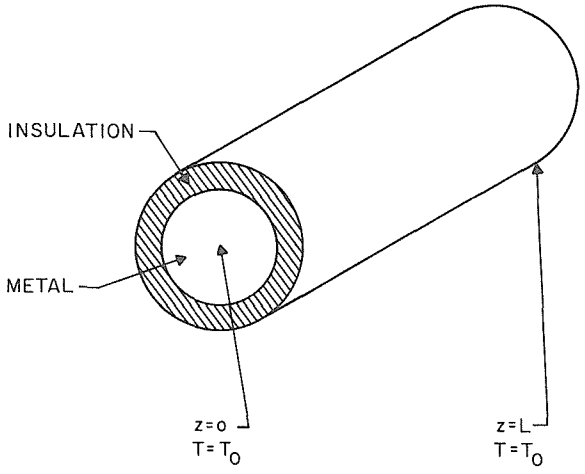


**FIGURE 3.4 Insulated metal rod.**

*SOLUTION*

First put (3.29) in dimensionless form by using $y = T - T_0$ and $x = z/L$.

$$\frac{d^2y}{dx^2} = \frac{\xi L^2}{K}[y + \cosh(1)]$$

Since $(\xi L^2)/K = 4$, the ordinary differential equation (ODE) becomes

$$\frac{d^2y}{dx^2} = 4[y + \cosh(1)]$$

Using $\mathscr{P}_4^2(\pi)$ (with $\pi$ uniform) gives the piecewise polynomial approximation

$$u(x) = \sum_{j=1}^{\ell+1} [\alpha_j^{(1)} v_j + \alpha_j^{(2)} s_j]$$

As with Example 1, $y(0) = y(1) = 0$ and, since $v_1(0) = 1$ and $v_{\ell+1}(1) = 1$,

$$u(x) = \alpha_1^{(2)} s_1 + \alpha_2^{(1)} v_2 + \alpha_2^{(2)} s_2, \ldots, \alpha_\ell^{(1)} v_\ell + \alpha_\ell^{(2)} s_\ell + \alpha_{\ell+1}^{(2)} s_{\ell+1}$$

The weak from of the ODE is

$$-(y', \phi_i') - 4(y, \phi_i) = 4(1, \phi_i)\cosh(1), \qquad i = 1, \ldots, 2(\ell + 1) - 2$$

Substitution of $u(x)$ into the above equation results in

$$-(u', \phi_i') - 4(u, \phi_i) = 4(1, \phi_i)\cosh(1), \qquad i = 1, \ldots, 2(\ell + 1) - 2$$

In matrix notation the previous equation is

$$[A + 4B]\,\boldsymbol{\alpha} = -4\cosh(1)\mathbf{F}$$

where

$$A = \begin{bmatrix} (s_1', s_1') & (s_1', v_2') & (s_1', s_2') & & & & 0 \\ (v_2', s_1') & (v_2', v_2') & (v_2', s_2') & (v_2', v_3') & (v_2', s_3') & & \\ (s_2', s_1') & (s_2', v_2') & (s_2', s_2') & (s_2', v_3') & (s_2', s_3') & & \\ & \cdot & \cdot & \cdot & \cdot & & \\ & & \cdot & & & \cdot & \\ & & & \cdot & & & \cdot \\ 0 & & & \cdot & & & \cdot \\ & & (v_\ell', v_{\ell-1}') & (v_\ell', s_{\ell-1}') & (v_\ell', v_\ell') & (v_\ell', s_\ell') & (v_\ell', s_{\ell+1}') \\ & & (s_\ell', v_{\ell-1}') & (s_\ell', s_{\ell-1}') & (s_\ell', v_\ell') & (s_\ell', s_\ell') & (s_\ell', s_{\ell+1}') \\ & & & & (s_{\ell+1}', v_\ell') & (s_{\ell+1}', s_\ell') & (s_{\ell+1}', s_{\ell+1}') \end{bmatrix}$$

$\boldsymbol{B}$ = the same as $\boldsymbol{A}$ except for no primes on the basis functions

$\mathbf{F} = [(1, s_1), (1, v_2), (1, s_2), \ldots, (1, v_\ell), (1, s_\ell), (1, s_{\ell+1})]^T$

$\boldsymbol{\alpha} = [\alpha_1^{(2)}, \alpha_2^{(1)}, \alpha_2^{(2)}, \ldots, \alpha_\ell^{(1)}, \alpha_\ell^{(2)}, \alpha_{\ell+1}^{(2)}]^T$

Each of the inner products ( , ) shown in $\boldsymbol{A}$, $\boldsymbol{B}$, and $\mathbf{F}$ must be evaluated. For example

$$v_i(x) = g_1(\xi_{i-1}(x)) + g_1(1 - \xi_i(x))$$

with

$$\xi_{i-1}(x) = \frac{x - x_{i-1}}{x_i - x_{i-1}}, \qquad 1 - \xi_i(x) = \frac{x_{i+1} - x}{x_{i+1} - x_i}$$

Therefore

$$v_i(x) = -2\delta_1 \left[\frac{x - x_{i-1}}{x_i - x_{i-1}}\right]^3 + 3\delta_1 \left[\frac{x - x_{i-1}}{x_i - x_{i-1}}\right]^2 - 2\delta_2 \left[\frac{x_{i+1} - x}{x_{i+1} - x_i}\right]^3 + 3\delta_2 \left[\frac{x_{i+1} - x}{x_{i+1} - x_i}\right]^2$$

where

$$\delta_1 = \begin{cases} 1, & x_{i-1} \leq x \leq x_i \\ 0, & \text{otherwise} \end{cases}$$

$$\delta_2 = \begin{cases} 1, & x_i \leq x \leq x_{i+1} \\ 0, & \text{otherwise} \end{cases}$$

and

$$(1, v_i) = \frac{h_i}{2} + \frac{h_{i+1}}{2}$$

or for a uniform partition,

$$(1, v_i) = h$$

Once all the inner products are determined, the matrix problem is ready to be solved. Notice the structure of $A$ or $B$ (they are the same). These matrices are block-tridiagonal and can be solved using a well-known block version of Gaussian elimination (see page 196 of [3]). The results are shown below.

| $t$ | $h$ (uniform partition) | $\|y - u\|$ |
|---|---|---|
| 1 | 0.1250 | $0.6011 \times 10^{-5}$ |
| 2 | 0.0556 | $0.2707 \times 10^{-6}$ |
| 3 | 0.0357 | $0.4872 \times 10^{-7}$ |
| 4 | 0.0263 | $0.1475 \times 10^{-7}$ |

Since $\|y - u\| \leq Ch^p$, take the logarithm of this equation to give

$$\ln\|y - u\| \leq \ln C + pLnh$$

Let $e(h) = \|y - u\|$ ($u$ calculated with a uniform partition; subinterval size $h$), and calculate $p$ by

$$p = \frac{\ln\left(\frac{e(h_{t-1})}{e(h_t)}\right)}{\ln\left(\frac{h_{t-1}}{h_t}\right)}$$

From the above results,

| $t$ | $p$ |
|---|---|
| 1 | — |
| 2 | 3.83 |
| 3 | 3.87 |
| 4 | 3.91 |

which shows the fourth-order accuracy of the method.

Thus using $\mathscr{D}_4^2(\pi)$ as the approximating space gives a Galerkin solution possessing a continuous first derivative that is fourth-order accurate.

## Nonlinear Equations

Consider the nonlinear ODE:

$$y'' = f(x, y, y'), \qquad 0 < x < 1$$
$$y(0) = y(1) = 0 \tag{3.30}$$

Using the B-spline basis gives the pp-approximation

$$u(x) = \sum_{j=1}^{N} \alpha_j B_j(x) \tag{3.31}$$

Substitution of (3.31) into the weak form of (3.30) yields

$$\left(\sum_{j=1}^{N} \alpha_j B_j', B_i'\right) + \left(f\left(x, \sum_{j=1}^{N} \alpha_j B_j, \sum_{j=1}^{N} \alpha_j B_j'\right), B_i\right) = 0, \qquad (3.32)$$

$$i = 1, \ldots, N$$

The system (3.32) can be written as

$$A\alpha + \mathbf{H}(\alpha) = \mathbf{0} \qquad (3.33)$$

where the vector $\mathbf{H}$ contains inner products that are nonlinear functions of $\alpha$. Equation (3.33) can be solved using Newton's method, but notice that the vector $\mathbf{H}$ must be recomputed after each iteration. Therefore, the computation of $\mathbf{H}$ must be done efficiently. Normally, the integrals in $\mathbf{H}$ do not have closed form, and one must resort to numerical quadrature. The rule of thumb in this case is to use at least an equal number of quadrature points as the degree of the approximating space.

### Inhomogeneous Dirichlet and Flux Boundary Conditions

The Galerkin procedures discussed in the previous sections may easily be modified to treat boundary conditions other than the homogeneous Dirichlet conditions, that is, $y(0) = y(1) = 0$. Suppose that the governing ODE is

$$(a(x)y'(x))' + b(x)y(x) + c(x) = 0, \qquad 0 < x < 1 \qquad (3.34)$$

subject to the boundary conditions

$$y(0) = \psi_1, \qquad y(1) = \psi_2 \qquad (3.35)$$

where $\psi_1$ and $\psi_2$ are constants. The weak form of (3.34) is

$$a(x)y'(x)B_i(x)\Big|_0^1 - (a(x)y'(x), B_i'(x)) + (b(x)y(x), B_i(x)) + (c(x), B_i(x)) = 0 \qquad (3.36)$$

Since

$$B_1(0) = 1, \qquad \sum_{j=2}^{N} B_j(0) = 0$$

and

$$B_N(1) = 1, \qquad \sum_{j=1}^{N-1} B_j(1) = 0$$

then

$$\alpha_1 = \psi_1, \qquad \alpha_N = \psi_2 \qquad (3.37)$$

to match the boundary conditions. The value of $i$ in (3.36) goes from 2 to $N-1$ so that the basis functions satisfy the homogeneous Dirichlet conditions [eliminates the first term in (3.36)]. Thus (3.36) becomes:

$$\sum_{j=2}^{N-1} \alpha_j[(a(x)B_j', B_i') - (b(x)B_j, B_i)] = (c(x), B_i) + \psi_1[-(a(x)B_1', B_i') + (b(x)B_1, B_i)] + \psi_2[-(a(x)B_N', B_i') + (b(x)B_N, B_i)], \quad i = 2, \ldots, N-1 \tag{3.38}$$

If flux conditions are prescribed, they can be represented by

$$\eta_1 y + \beta_1 y' = \gamma_1 \quad \text{at} \quad x = 0$$

$$\eta_2 y + \beta_2 y' = \gamma_2 \quad \text{at} \quad x = 1 \tag{3.39}$$

where $\eta_1$, $\eta_2$, $\beta_1$, $\beta_2$, $\gamma_1$, and $\gamma_2$ are constants and satisfy

$$|\eta_1| + |\beta_1| > 0$$

$$|\eta_2| + |\beta_2| > 0$$

Write (3.39) as

$$y' = \frac{\gamma_1}{\beta_1} - \frac{\eta_1}{\beta_1} y \quad \text{at} \quad x = 0$$

$$y' = \frac{\gamma_2}{\beta_2} - \frac{\eta_2}{\beta_2} y \quad \text{at} \quad x = 1 \tag{3.40}$$

Incorporation of (3.40) into (3.36) gives:

$$\sum_{j=1}^{N} \alpha_j \left[ (a(x)B_j', B_i') - (b(x)B_j, B_i) - \delta_{i1}\delta_{j1} a(0) \frac{\eta_1}{\beta_1} + \delta_{iN}\delta_{jN} a(1) \frac{\eta_2}{\beta_2} \right]$$

$$= (c(x), B_i) + \delta_{iN} a(1) \frac{\gamma_2}{\beta_2} - \delta_{i1} a(0) \frac{\gamma_1}{\beta_1}, \quad i = 1, \ldots, N \tag{3.41}$$

where

$$\delta_{st} = \begin{cases} 1, & s = t \\ 0, & s \neq t \end{cases}$$

Notice that the subscript $i$ now goes from 1 to $N$, since $y(0)$ and $y(1)$ are unknowns.

## Mathematical Software

In light of the fact that Galerkin methods are not frequently used to solve BVPs (because of the computational effort as compared with other methods, e.g., finite differences, collocation), it is not surprising that there is very limited

software that implements Galerkin methods for BVPs. Galerkin software for BVPs consists of Schryer's code in the PORT library developed at Bell Laboratories [4]. There is a significant amount of Galerkin-based software for partial differential equations, and we will discuss these codes in the chapters concerning partial differential equations. The purpose for covering Galerkin methods for BVPs is for ease of illustration, and because of the straightforward extension into partial differential equations.

## COLLOCATION

Consider the nonlinear ODE

$$y'' = f(x, y, y'), \qquad a < x < b \tag{3.42a}$$

$$\eta_1 y + \beta_1 y' = \gamma_1 \quad \text{at} \quad x = a$$

$$\eta_2 y + \beta_2 y' = \gamma_2 \quad \text{at} \quad x = b \tag{3.42b}$$

where $\eta_1$, $\eta_2$, $\beta_1$, $\beta_2$, $\gamma_1$, and $\gamma_2$ are constants. Let the interval partition be given by (3.13), and let the pp-approximation in $\mathscr{P}_k^{\nu}(\pi)$ ($\nu \geq 2$) be (3.31). The collocation method determines the unknown set $\{\alpha_j|_j = 1, \ldots, N\}$ by satisfying the ODE at $N$ points. For example, if $k = 4$ and $\nu = 2$, then $N = 2\ell + 2$. If we satisfy the two boundary conditions (3.42b), then two collocation points are required in each of the $\ell$ subintervals. It can be shown that the optimal position of the collocation points are the $k - M$ ($M$ is the degree of the ODE; in this case $M = 2$) Gaussian points given by [5]:

$$\tau_{ji} = x_j + \frac{h_j}{2} + \left(\frac{h_j}{2}\right)\omega_i, \qquad j = 1, \ldots, \ell, \quad i = 1, \ldots, k - M \tag{3.43}$$

where

$$\omega = k - M \text{ Gaussian points in } [-1, 1]$$

The $k - M$ Gaussian points in $[-1, 1]$ are the zeros of the Legendre polynomial of degree $k - M$. For example, if $k = 4$ and $M = 2$, then the two Gaussian points are the zeros of the Legendre polynomial

$$-\tfrac{1}{2} + \tfrac{3}{2}x^2, \qquad -1 \leq x \leq 1$$

or

$$\omega_1 = -\frac{1}{\sqrt{3}}, \qquad \omega_2 = \frac{1}{\sqrt{3}}$$

Thus, the two collocation points in each subinterval are given by

$$\tau_{j1} = x_j + \frac{h_j}{2} - \frac{h_j}{2}\left(\frac{1}{\sqrt{3}}\right)$$

$$\tau_{j2} = x_j + \frac{h_j}{2} + \frac{h_j}{2}\left(\frac{1}{\sqrt{3}}\right) \tag{3.44}$$

The $2\ell$ equations specified at the collocation points combined with the two boundary conditions completely determines the collocation solution $\{a_j | j = 1, \ldots, 2\ell + 2\}$.

**EXAMPLE 3**

Solve Example 2 using spline collocation at the Gaussian points and the Hermite cubic basis. Show the order of accuracy.

*SOLUTION*

The governing ODE is:

$$\frac{d^2y}{dx^2} = 4[y + \cosh(1)], \qquad 0 < x < 1$$
$$y(0) = y(1) = 0$$

Let

$$Ly = -y'' + 4y = -4\cosh(1)$$

and consider a general subinterval $[x_j, x_{j+1}]$ in which there are four basis functions—$v_j$, $v_{j+1}$, $s_j$, and $s_{j+1}$—that are nonzero. The "value" functions are evaluated as follows:

$$v_j = g_1(1 - \xi_j(x)), \qquad [x_j, x_{j+1}]$$

$$v_j = -2\left[\frac{x_{j+1} - x}{h}\right]^3 + 3\left[\frac{x_{j+1} - x}{h}\right]^2, \qquad h = x_{j+1} - x_j$$

$$v_j'' = -\frac{12}{h^3}(x_{j+1} - x) + \frac{6}{h^2}$$

$$Lv_j = \frac{12}{h^3}(x_{j+1} - x) - \frac{6}{h^2} - \frac{8}{h^3}(x_{j+1} - x)^3 + \frac{12}{h^2}(x_{j+1} - x)^2$$

$$v_{j+1} = g_1(\xi_j(x)), \qquad [x_j, x_{j+1}]$$

$$v_{j+1} = -2\left[\frac{x - x_j}{h}\right]^3 + 3\left[\frac{x - x_j}{h}\right]^2$$

$$v_{j+1}'' = -\frac{12}{h^3}(x - x_j) + \frac{6}{h^2}$$

$$Lv_{j+1} = \frac{12}{h^3}(x - x_j) - \frac{6}{h^2} - \frac{8}{h^3}(x - x_j)^3 + \frac{12}{h^2}(x - x_j)^2$$

The two collocation points per subinterval are

$$\tau_{j1} = x_j + \frac{h}{2} - \frac{h}{2}\left(\frac{1}{\sqrt{3}}\right) = x_j + \frac{h}{2}\left[1 - \frac{1}{\sqrt{3}}\right] = x_j + hp_1$$

$$\tau_{j2} = x_j + \frac{h}{2} + \frac{h}{2}\left(\frac{1}{\sqrt{3}}\right) = x_j + \frac{h}{2}\left[1 - \frac{1}{\sqrt{3}}\right] = x_j + hp_2$$

Using $\tau_{j1}$ and $\tau_{j2}$ in $Lv_j$ and $Lv_{j+1}$ gives

$$Lv_j(\tau_{j1}) = \frac{12}{h^2}(1 - \rho_1) - \frac{6}{h^2} - 8(1 - \rho_1)^3 + 12(1 - \rho_1)^2$$

$$Lv_j(\tau_{j2}) = \frac{12}{h^2}(1 - \rho_2) - \frac{6}{h^2} - 8(1 - \rho_2)^3 + 12(1 - \rho_2)^2$$

$$Lv_{j+1}(\tau_{j1}) = \frac{12}{h^2}\rho_1 - \frac{6}{h^2} - 8\rho_1^3 + 12\rho_1^2$$

$$Lv_{j+1}(\tau_{j2}) = \frac{12}{h^2}\rho_2 - \frac{6}{h^2} - 8\rho_2^3 + 12\rho_2^2$$

The same procedure can be used for the "slope" functions to produce

$$Ls_j(\tau_{j1}) = \frac{6}{h}(1 - \rho_1) - \frac{2}{h} - 4h[(1 - \rho_1)^3 - (1 - \rho_1)^2]$$

$$Ls_j(\tau_{j2}) = \frac{6}{h}(1 - \rho_2) - \frac{2}{h} - 4h[(1 - \rho_2)^3 - (1 - \rho_2)^2]$$

$$Ls_{j+1}(\tau_{j1}) = -\frac{6}{h}\rho_1 + \frac{2}{h} + 4h[\rho_1^3 - \rho_1^2],$$

$$Ls_{j+1}(\tau_{j2}) = -\frac{6}{h}\rho_2 + \frac{2}{h} + 4h[\rho_2^3 - \rho_2^2].$$

For notational convenience let

$$F1 = Ls_j(\tau_{j1}) = -Ls_{j+1}(\tau_{j2})$$

$$F2 = Ls_j(\tau_{j2}) = -Ls_{j+1}(\tau_{j1})$$

$$F3 = Lv_j(\tau_{j2}) = Lv_{j+1}(\tau_{j1})$$

$$F4 = Lv_j(\tau_{j1}) = Lv_{j+1}(\tau_{j2})$$

At $x = 0$ and $x = 1$, $y = 0$. Therefore,

$$\alpha_1^{(1)} = \alpha_{\ell+1}^{(1)} = 0$$

Thus the matrix problem becomes:

$$\begin{bmatrix} F1 & F3 & -F2 & & & & & \\ F2 & F4 & -F1 & & & & & \\ & F4 & F1 & F3 & -F2 & & 0 & \\ & F3 & F2 & F4 & -F1 & & & \\ & & \cdot & & & & & \\ & & & \cdot & & & & \\ & & & & \cdot & & & \\ & & & & & F4 & F1 & -F2 \\ & 0 & & & & F3 & F2 & -F1 \end{bmatrix} \begin{bmatrix} \alpha_1^{(2)} \\ \alpha_2^{(1)} \\ \alpha_2^{(2)} \\ \alpha_3^{(1)} \\ \cdot \\ \cdot \\ \cdot \\ \alpha_\ell^{(2)} \\ \alpha_{\ell+1}^{(2)} \end{bmatrix} = -4\cosh(1) \begin{bmatrix} 1 \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ \\ \\ 1 \\ 1 \end{bmatrix}$$

This matrix problem was solved using the block version of Gaussian elimination (see page 196 of [4]). The results are shown below.

| $t$ | $h$ (uniform partition) | $\|y - u\|$ |
|---|---|---|
| 1 | 0.100 | $0.2830 \times 10^{-6}$ |
| 2 | 0.050 | $0.1764 \times 10^{-7}$ |
| 3 | 0.033 | $0.3483 \times 10^{-8}$ |
| 4 | 0.250 | $0.1102 \times 10^{-8}$ |

From the above results

| $t$ | $p$ |
|---|---|
| 1 | — |
| 2 | 4.00 |
| 3 | 3.90 |
| 4 | 4.14 |

which shows fourth-order accuracy.

In the previous example we showed that when using $\mathscr{L}_k^{\nu}(\pi)$, the error was $0(h^4)$. In general, the collocation method using $\mathscr{L}_k^{\nu}(\pi)$ gives an error of the same order as that in Galerkin's method [Eq. (3.28)] [5].

## EXAMPLE 4

The problem of predicting diffusion and reaction in porous catalyst pellets was discussed in Chapter 2. In that discussion the boundary condition at the surface was the specification of a known concentration. Another boundary condition that can arise at the surface of the pellet is the continuity of flux of a species as a result of the inclusion of a boundary layer around the exterior of the pellet. Consider the problem of calculating the concentration profile in an isothermal catalyst pellet that is a slab and is surrounded by a boundary layer. The conservation of mass equation is

$$D\frac{d^2c}{dx^2} = k\mathscr{R}(c), \qquad 0 < x < x_p$$

where

$D$ = diffusivity

$x$ = spatial coordinate ($x_p$ = half thickness of the plate)

$c$ = concentration of a given species

$k$ = rate constant

$\mathscr{R}(c)$ = reaction rate function

The boundary conditions for this equation are

$$\frac{dc}{dx} = 0 \quad \text{at} \quad x = 0 \qquad \text{(symmetry)}$$

$$-D\frac{dc}{dx} = S_h(c - c_0) \quad \text{at} \quad x = x_p \qquad \text{(continuity of flux)}$$

where

$c_0$ = known concentration at the exterior of the boundary layer

$S_h$ = mass transfer coefficient

Set up the matrix problem to solve this ODE using collocation with $\mathscr{P}_4^2(\overline{\pi})$, where

$$\overline{\pi}: \quad 0 = x_1 < x_2 < \ldots < x_{\ell+1} = x_p$$

and

$$h = x_{i+1} - x_i, \qquad \text{for } 1 \leqslant i \leqslant \ell \quad \text{(i.e., uniform)}$$

*SOLUTION*

First, put the conservation of mass equation in dimensionless form by defining

$$C = \frac{c}{c_0}$$

$$z = \frac{x}{x_p}$$

$$\Phi = x_p\sqrt{\frac{k}{D}} \qquad \text{(Thiele modulus)}$$

$$\text{Bi} = \frac{S_h x_p}{D} \qquad \text{(Biot number)}$$

With these definitions, the ODE becomes

$$\frac{d^2C}{dz^2} = \Phi^2\left[\frac{\mathscr{R}(c)}{c_0}\right]$$

$$\frac{dC}{dz} = 0 \quad \text{at} \quad z = 0$$

$$\frac{dC}{dz} = \text{Bi}\,(1 - C) \quad \text{at} \quad z = 1$$

The dimension of $\mathscr{L}_4^2(\overline{\pi})$ is $2(\ell + 1)$, and there are two collocation points in each subinterval.

The pp-approximation is

$$u(x) = \sum_{j=1}^{\ell+1} (\alpha_j^{(1)} v_j + \alpha_j^{(2)} s_j)$$

With $C'(0) = 0$, $\alpha_1^{(2)}$ is zero since $s_1' = 1$ is the only nonzero basis function in $u'(0)$. For each subinterval there are two equations such that

$$\sum_{j=1}^{\ell+1} [\alpha_j^{(1)} v_j''(\tau_{i1}) + \alpha_j^{(2)} s_j''(\tau_{i1})] = \frac{\Phi^2}{c_0} \mathscr{R} \left\{ c_0 \left[ \sum_{j=1}^{\ell+1} \alpha_j^{(1)} v_j(\tau_{i1}) + \alpha_j^{(2)} s_j(\tau_{i1}) \right] \right\}$$

$$\sum_{j=1}^{\ell+1} [\alpha_j^{(1)} v_j''(\tau_{i2}) + \alpha_j^{(2)} s_j''(t_{i2})] = \frac{\Phi^2}{c_0} \mathscr{R} \left\{ c_0 \left[ \sum_{j=1}^{\ell+1} \alpha_j^{(1)} v_j(\tau_{i2}) + \alpha_j^{(2)} s_j(\tau_{i2}) \right] \right\}$$

for $i = 1, \ldots, \ell$. At the boundary $z = 1$ we have

$$\alpha_{\ell+1}^{(2)} = \text{Bi}\ (1 - \alpha_{\ell+1}^{(1)})$$

since $s_{\ell+1}' = 1$ is the only nonzero basis function in $u'(1)$ and $v_{\ell+1} = 1$ is the only nonzero basis funciton in $u(1)$.

Because the basis is local, the equations at the collocation points can be simplified. In matrix notation:

$$\begin{bmatrix} v_1''(\tau_{11}),\ v_2''(\tau_{11}),\ s_2''(\tau_{11}) & & \\ v_1''(\tau_{12}),\ v_2''(\tau_{12}),\ s_2''(\tau_{12}) & & \\ v_2''(\tau_{21}),\ s_2''(\tau_{21}),\ v_3''(\tau_{21}),\ s_3''(\tau_{21}) & & 0 \\ v_2''(\tau_{22}),\ s_2''(\tau_{22}),\ v_3''(\tau_{22}),\ s_3''(\tau_{22}) & & \\ & \cdot & \\ & \cdot & \\ & \cdot & \\ & \cdot & \\ 0 & v_\ell''(\tau_{\ell 1}),\ s_\ell''(\tau_{\ell 1}),\ v_{\ell+1}''(\tau_{\ell 1}) - \text{Bi}\ s_{\ell+1}''(\tau_{\ell 1}) & \\ & v_\ell''(\tau_{\ell 2}),\ s_\ell''(\tau_{\ell 2}),\ v_{\ell+1}''(\tau_{\ell 2}) - \text{Bi}\ s_{\ell+1}''(\tau_{\ell 2}) & \end{bmatrix} \begin{bmatrix} \alpha_1^{(1)} \\ \alpha_2^{(1)} \\ \alpha_2^{(2)} \\ \alpha_3^{(1)} \\ \alpha_3^{(2)} \\ \cdot \\ \cdot \\ \cdot \\ \alpha_\ell^{(1)} \\ \alpha_\ell^{(2)} \\ \alpha_{\ell+1}^{(1)} \end{bmatrix} = \frac{\Phi^2}{c_0} \mathbf{F}$$

where

$$\mathbf{F} = \begin{bmatrix} \mathscr{R}\{c_0[\alpha_1^{(1)}v_1(\tau_{11}) + \alpha_2^{(1)}v_2(\tau_{11}) + \alpha_2^{(2)}s_2(\tau_{11})]\} \\ \vdots \\ \mathscr{R}\{c_0[\alpha_j^{(1)}v_j(\tau_{j1}) + \alpha_j^{(2)}s_j(\tau_{j1}) + \alpha_{j+1}^{(1)}v_{j+1}(\tau_{j1}) + \alpha_{j+1}^{(2)}s_{j+1}(\tau_{j1})]\} \\ \mathscr{R}\{c_0[\alpha_j^{(1)}v_j(\tau_{j2}) + \alpha_j^{(2)}s_j(\tau_{j2}) + \alpha_{j+1}^{(1)}v_{j+1}(\tau_{j2}) + \alpha_{j+1}^{(2)}s_{j+1}(\tau_{j2})]\} \\ \vdots \\ -\dfrac{c_0\mathrm{Bi}}{\Phi^2}s''_{\ell+1}(\tau_{\ell 1}) + \mathscr{R}\{c_0[\alpha_\ell^{(1)}v_\ell(\tau_{\ell 1}) + \alpha_\ell^{(2)}s_\ell(\tau_{\ell 1}) \\ \qquad + \alpha_{\ell+1}^{(1)}(v_{\ell+1}(\tau_{\ell 1}) - \mathrm{Bi}s_{\ell+1}(\tau_{\ell 1})) + \mathrm{Bi}s_{\ell+1}(\tau_{\ell 1})]\} \\ -\dfrac{c_0\mathrm{Bi}}{\Phi^2}s''_{\ell+1}(\tau_{\ell 2}) + \mathscr{R}\{c_0[\alpha_\ell^{(1)}v_\ell(\tau_{\ell 2}) + \alpha_\ell^{(2)}s_\ell(\tau_{\ell 2}) \\ \qquad + \alpha_{\ell+1}^{(1)}(v_{\ell+1}(\tau_{\ell 2}) - \mathrm{Bi}s_{\ell+1}(\tau_{\ell 2})) + \mathrm{Bi}s_{\ell+1}(\tau_{\ell 2})]\} \end{bmatrix}$$

This problem is nonlinear, and therefore Newton's method or a variant of it would be used. At each iteration the linear system of equations can be solved efficiently by the alternate row and column elimination procedure of Varah [6]. This procedure has been modified and a FORTRAN package was produced by Diaz et al. [7].

As a final illustration of collocation, consider the $m$ nonlinear ODEs

$$\mathbf{y}'' = \mathbf{f}(x, \mathbf{y}, \mathbf{y}'), \qquad a < x < b \tag{3.45a}$$

with

$$\mathbf{g}(\mathbf{y}(a), \mathbf{y}(b), \mathbf{y}'(a), \mathbf{y}'(b)) = \mathbf{0} \tag{3.45b}$$

The pp-approximations ( $\mathscr{P}_k^{\nu}(\pi)$) for this system can be written as

$$\mathbf{u}(x) = \sum_{j=1}^{N} \boldsymbol{\alpha}_j B_j(x) \tag{3.46}$$

where each $\boldsymbol{\alpha}_j$ is a constant vector of length $m$. The collocation equations for (3.45) are

$$\sum_{j=1}^{N} \boldsymbol{\alpha}_j B_j''(\tau_{si}) = \mathbf{f}\left(x, \sum_{j=1}^{N} \boldsymbol{\alpha}_j B_j(\tau_{si}), \sum_{j=1}^{N} \boldsymbol{\alpha}_j B_j'(\tau_{si})\right)$$
$$i = 1, \ldots, k-2, \qquad s = 1, \ldots, \ell \tag{3.47}$$

and,

$$\mathbf{g}\left(\sum_{j=1}^{N} \boldsymbol{\alpha}_j B_j(a), \sum_{j=1}^{N} \boldsymbol{\alpha}_j B_j(b), \sum_{j=1}^{N} \boldsymbol{\alpha}_j B_j'(a), \sum_{j=1}^{N} \boldsymbol{\alpha}_j B_j'(b)\right) = \mathbf{0} \tag{3.48}$$

If there are $m$ ODEs in the system and the dimension of $\mathscr{P}_k^{\nu}(\pi)$ is $N$, then there are $mN$ unknown coefficients that must be obtained from the nonlinear algebraic system of equations composed of (3.47) and (3.48). From (3.23)

$$N = k + \sum_{j=2}^{\ell} (k - \nu) \tag{3.49}$$

and the number of coefficients is thus

$$mk + m(\ell - 1)(k - \nu) \tag{3.50}$$

The number of equations in (3.47) is $m(k - 2)\ell$, and in (3.48) is $2m$. Therefore the system (3.47) and (3.48) is composed of $2m + m\ell(k - 2)$ equations. If we impose continuity of the first derivative, that is, $\nu = 2$, then (3.50) becomes

$$mk + m(\ell - 1)(k - 2)$$

or

$$2m + m\ell(k - 2) \tag{3.51}$$

Thus the solution of the system (3.47) and (3.48) completely specifies the pp-approximation.

## Mathematical Software

The available software that is based on collocation is rather limited. In fact, it consists of one code, namely COLSYS [8]. Next, we will study this code in detail.

COLSYS uses spline collocation to determine the solution of the mixed-order system of equations

$$u_s^{(M_s)}(x) = f_s(x; \mathbf{z}(\mathbf{u})), \qquad s = 1, \ldots, d$$
$$a < x < b \tag{3.52}$$

where

$$\begin{aligned} M_s &= \text{order of the } s \text{ differential equation} \\ \mathbf{u} &= [u_1, u_2, \ldots, u_d]^T \text{ is the vector of solutions} \\ \mathbf{z}(\mathbf{u}) &= (u_1, u_1', \ldots, u_1^{(M_1-1)}, \ldots, u_d, u_d', \ldots, u_d^{(M_d-1)}) \end{aligned}$$

It is assumed that the components $u_1, u_2, \ldots, u_d$ are ordered such that

$$M_1 \leq M_2 \leq \ldots \leq M_d \leq 4 \tag{3.53}$$

Equations (3.52) are solved with the conditions

$$g_j(\xi_j, \mathbf{z}(\mathbf{u})) = 0, \qquad j = 1, \ldots, M^* \tag{3.54}$$

where

$$M^* = \sum_{s=1}^{d} M_s$$

and

$$a \leq \xi_i \leq \xi_2 \leq \ldots \leq \xi_{M^*} \leq b$$

Unlike the BVP codes in Chapter 2, COLSYS does not convert (3.52) to a first-order system. While (3.54) does not allow for nonseparated boundary conditions,

such problems can be converted to the form (3.54) [9]. For example, consider the BVP

$$\begin{aligned} y'' &= f(x, y, y'), \qquad a < x < b \\ y'(a) &= \alpha, \qquad g(y(a), y(b)) = 0 \end{aligned} \tag{3.55}$$

Introducing a (constant) $V(x)$ gives an equivalent BVP

$$\begin{aligned} y'' &= f(x, y, y'), \qquad a < x < b \\ V' &= 0, \\ y'(a) &= \alpha, \qquad y(a) = V(a), \qquad g(V(b), y(b)) = 0 \end{aligned} \tag{3.56}$$

which does not contain a nonseparated boundary condition.

COLSYS implements the method of spline collocation at Gaussian points using a B-spline basis (modified versions of deBoor's algorithms [2] are used to calculate the B-splines and their derivates). The pp-solutions are thus in $\mathscr{P}_k^{\nu^*}(\pi)$ where COLSYS sets $k$ and $\nu^*$ such that

$$u_s \quad \text{is in } \mathscr{P}_{q+M_s}^{\nu^*}(\pi), \qquad s = 1, \ldots, d \tag{3.57}$$

where

$$\nu^* = \{\nu_j = M_s \mid j = 2, \ldots, \ell\}$$

q = number of collocation points per subintervals

The matrix problem is solved using an efficient implementation of Gaussian elimination with partial pivoting [10], and nonlinear problems are "handled" by the use of a modified Newton method. Algorithms are included for estimating the error, and for mesh refinement. A redistribution of mesh points is automatically performed (if deemed worthwhile) to roughly equidistribute the error. This code has proven to be quite effective for the solution of "difficult" BVPs arising in chemical engineering [11].

To illustrate the use of COLSYS we will solve the isothermal effectiveness factor problem with large Thiele moduli. The governing BVP is the conservation of mass in a porous plate catalyst pellet where a second-order reaction rate is occurring, i.e.,

$$\begin{aligned} \frac{d^2c}{dx^2} &= \Phi^2 c^2, \qquad 0 < x < 1, \\ c'(0) &= 0 \\ c(1) &= 1 \end{aligned} \tag{3.58}$$

where

$c$ = dimensionless concentration of a given species

$x$ = dimensionless coordinate

$\Phi$ = Thiele modulus (constant)

The effectiveness factor (defined in Chapter 2) for this problem is

$$E = \int_0^1 c^2 \, dx \tag{3.59}$$

For large values of $\Phi$, the "exact" solution can be obtained [12] and is

$$E = \frac{1}{\Phi}\sqrt{\frac{2}{3}}(1 - c_0^3)^{1/2} \tag{3.60}$$

where $c_0$ is given by

$$\Phi\sqrt{\frac{2}{3}}c_0 = \int_0^{1/c_0} \frac{d\xi}{\sqrt{\xi^3 - 1}} \tag{3.61}$$

This problem is said to be difficult because of the extreme gradient in the solution (see Figure 3.5). We now present the results generated by COLSYS.

COLSYS was used to solve (3.58) with $\Phi = 50$, 100, and 150. A tolerance was set on the solution and the first derivative, and an initial uniform mesh of five subintervals was used with initial solution and derivative profiles of 0.1 and 0.001 for $0 \leq x \leq 1$, respectively. The solution for $\Phi = 50$ was used as the initial profile for calculating the solution with $\Phi = 100$, and subsequently this solution was used to calculate the solution for $\Phi = 150$. Table 3.3 compares

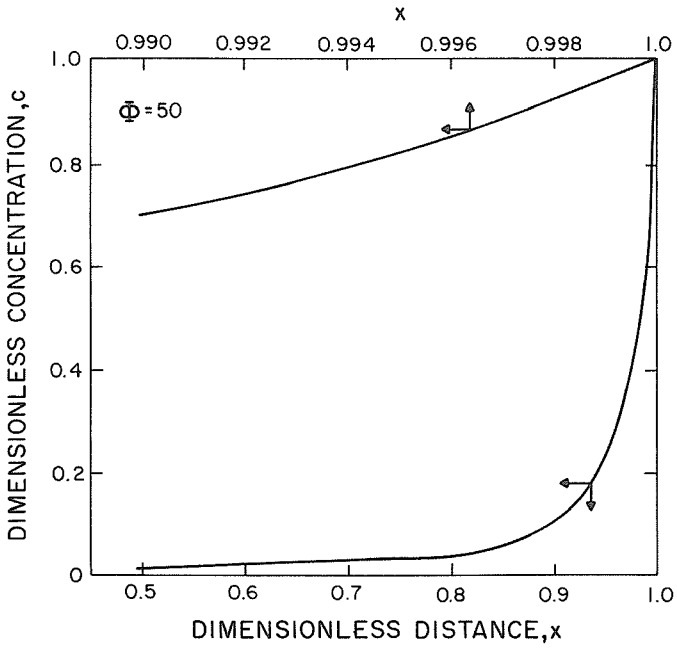


**FIGURE 3.5 Solution of Eq. (3.58).**

**TABLE 3.3 Results for Eq. (3.58)**
**Tolerance = $10^{-4}$**
**Collocation Points Per Subinterval = 3**

| $\Phi$ | COLSYS | "Exact" |
|---|---|---|
| 50 | 0.1633(−1) | 0.1633(−1) |
| 100 | 0.8165(−2) | 0.8165(−2) |
| 150 | 0.5443(−2) | 0.5443(−2) |

the results computed by COLSYS with those of (3.60) and (3.61). This table shows that COLSYS is capable of obtaining accurate results for this "difficult" problem.

COLSYS incorporates an error estimation and mesh refinement algorithm. Figure 3.6 shows the redistribution of the mesh for $\Phi = 50$, $q = 4$, and the tolerance = $10^{-4}$. With the initial uniform mesh (mesh redistribution number = 0; i.e., a mesh redistribution number of $\eta$ designates that COLSYS has automatically redistributed the mesh $\eta$ times), COLSYS performed eight Newton iterations on the matrix problem to achieve convergence. Since the computations continued, the error exceeded the specified tolerance. Notice that the mesh was then redistributed such that more points are placed in the region of the steep gradient (see Figure 3.5). This is done to "equidistribute" the error throughout the $x$ interval. Three additional redistributions of the mesh were required to provide an approximation that met the specified error tolerance. Finally, the effect of the tolerance and $q$, the number of collocation points per subinterval, were tested. In Table 3.4, one can see the results of varying the aforementioned parameters. In all cases shown, the same solution, $u(x)$, and value of $E$ were

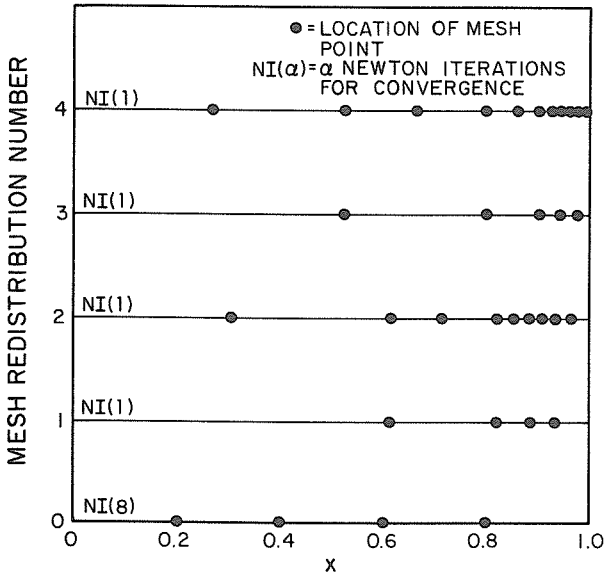


**FIGURE 3.6 Redistribution of mesh.**

**TABLE 3.4 Further Results for Eq. (3.58)**
**$\Phi = 50$**

| Collocation Points Per Subinterval | Number of Subintervals | Tolerance on Solution and Derivative | E.T.R.* |
|---|---|---|---|
| 3 | 20 | $10^{-4}$ | 1.0 |
| 3 | 114 | $10^{-6}$ | 4.6 |
| 2 | 80 | $10^{-4}$ | 1.9 |
| 4 | 12 | $10^{-4}$ | 1.1 |

* E.T.R. = execution time ratio.

obtained. As the tolerance is lowered, the number of subintervals and the execution time required for solution increase. This is not unexpected since we are asking the code to calculate a more accurate solution. When $q$ is raised from 3 to 4, there is a slight decrease in the number of subintervals required for solution, but this requires approximately the same execution time. If $q$ is reduced from 3 to 2, notice the quadrupling in the number of subintervals used for solution and also the approximate doubling of the execution time. The drastic changes in going from $q = 2$ to $q = 3$ and the relatively small changes when increasing $q$ from 3 to 4 indicate that for this problem one should specify $q \geq 3$.

In this chapter we have outlined two finite element methods and have discussed the limited software that implements these methods. The extension of these methods from BVPs to partial differential equations is shown in later chapters.

## PROBLEMS

1. A liquid is flowing in laminar motion down a vertical wall. For $z < 0$, the wall does not dissolve in the fluid, but for $0 < z < L$, the wall contains a species A that is slightly soluble in the liquid (see Figure 3.7, from [13]). In this situation, the change in the mass convection in the $z$ direction equals the change in the diffusion of mass in the $x$ direction, or

$$\frac{\partial}{\partial z}(u_z c_A) = D\frac{\partial^2 c_A}{\partial x^2}$$

where $u_z$ is the velocity and $D$ is the diffusivity. For a short "contact time" the partial differential equation becomes (see page 561 of [13]):

$$\hat{a}x\frac{\partial c_A}{\partial z} = D\frac{\partial^2 c_A}{\partial x^2}$$

$$c_A = 0 \quad \text{at} \quad z = 0$$

$$c_A = 0 \quad \text{at} \quad x = \infty$$

$$c_A = c_A^0 \quad \text{at} \quad x = 0$$

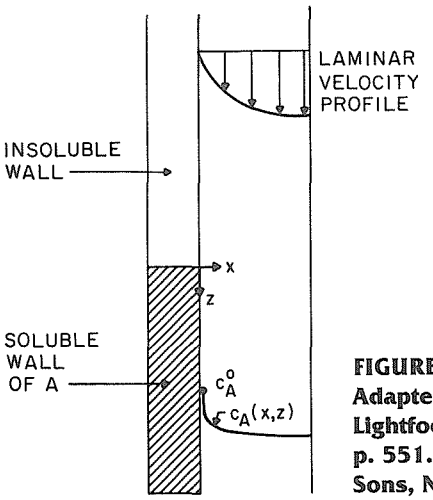


**FIGURE 3.7 Solid dissolution into falling film. Adapted from R. B. Bird, W. E. Stewart, and E. N. Lightfoot, *Transport Phenomena*, copyright © 1960, p. 551. Reprinted by permission of John Wiley and Sons, New York.**

where $\hat{a}$ is a constant and $c_A^0$ is the solubility of A in the liquid. Let

$$f = \frac{c_A}{c_A^0} \quad \text{and} \quad \xi = x\left(\frac{\hat{a}}{9Dz}\right)^{1/3}$$

The PDE can be transformed into a BVP with the use of the above dimensionless variables:

$$\frac{d^2f}{d\xi^2} + 3\xi^2\frac{df}{d\xi} = 0$$

$$f = 0 \quad \text{at} \quad \xi = \infty$$

$$f = 1 \quad \text{at} \quad \xi = 0$$

Solve this BVP using the Hermite cubic basis by Galerkin's method and compare your results with the closed-form solution (see p. 552 of [13]):

$$f = \frac{\int_\xi^\infty \exp(-\xi^3)d\xi}{\Gamma(\frac{4}{3})}$$

where $\Gamma(n) = \int_0^\infty \beta^{n-1}e^{-\beta}d\beta$, $(n > 0)$, which has the recursion formula

$$\Gamma(n+1) = n\Gamma(n)$$

The solution of the Galerkin matrix problem should be performed by calling an appropriate matrix routine in a library available at your installation.

2. Solve Problem 1 using spline collocation at Gaussian points.

3.* Solve problem 5 of Chapter 2 and compare your results with those obtained with a discrete variable method.

4.* The following problem arises in the study of compressible boundary layer flow at a point of attachment on a general curved surface [14].

$$f''' + (f + cg)f'' + (1 + S_w h - (f')^2) = 0$$

$$g''' + (f + cg)g'' + c(1 + S_w h - (f')^2) = 0$$

$$h'' + (f + cg)h' = 0$$

with

$$f = g = f' = g' = 0 \quad \text{at} \quad \eta = 0$$

$$h = 1 \quad \text{at} \quad \eta = 0$$

$$f' = g' = 1 \quad \text{at} \quad \eta \to \infty$$

$$h = 0 \quad \text{at} \quad \eta \to \infty$$

where $f$, $g$, and $h$ are functions of the independent variable $\eta$, and $c$ and $S_w$ are constants. As initial approximations use

$$f(\eta) = g(\eta) = \frac{\eta^2}{2\eta_\infty}$$

$$h(\eta) = \frac{\eta_\infty - \eta}{\eta_\infty}$$

where $\eta_\infty$ is the point at which the right-hand boundary conditions are imposed. Solve this problem with $S_w = 0$ and $c = 1.0$ and compare your results with those given in [11].

5.* Solve Problem 4 with $S_w = 0$ and $c = -0.5$. In this case there are two solutions. Be sure to calculate both solutions.

6.* Solve Problem 3 with $\beta = 0$ but allow for a boundary layer around the exterior of the pellet. The boundary condition at $x = 1$ now becomes

$$\frac{dy}{dx} = \text{Bi}\,(1 - y)$$

Vary the value of Bi and explain the effects of the boundary layer.

## REFERENCES

1. Fairweather, G., *Finite Element Galerkin Methods for Differential Equations,* Marcel Dekker, New York (1978).

2. deBoor, C., *Practical Guide to Splines,* Springer-Verlag, New York (1978).

3. Varga, R. S., *Matrix Iterative Analysis,* Prentice-Hall, Englewood Cliffs, N.J. (1962).
4. Fox, P. A., A. D. Hall, and N. L. Schryer, "The PORT Mathematical Subroutine Library," ACM TOMS, *4,* 104 (1978).
5. deBoor, C., and B. Swartz, "Collocation at Gaussian Points," SIAM J. Numer. Anal., *10,* 582 (1973).
6. Varah, J. M., "Alternate Row and Column Elimination for Solving Certain Linear Systems," SIAM J. Numer. Anal., *13,* 71 (1976).
7. Diaz, J. C., G. Fairweather, and P. Keast, "FORTRAN Packages for Solving Almost Block Diagonal Linear Systems by Alternate Row and Column Elimination," Tech. Rep. No. 148/81, Department of Computer Science, Univ. Toronto (1981).
8. Ascher, U., J. Christiansen, and R. D. Russell, "Collocation Software for Boundary Value ODEs," ACM TOMS, *7,* 209 (1981).
9. Ascher, U., and R. D. Russell, "Reformulation of Boundary Value Problems Into "Standard" Form," SIAM Rev. *23,* 238 (1981).
10. deBoor, C., and R. Weiss, "SOLVEBLOK: A Package for Solving Almost Block Diagonal Linear Systems," ACM TOMS, *6,* 80 (1980).
11. Davis, M., and G. Fairweather, "On the Use of Spline Collocation for Boundary Value Problems Arising in Chemical Engineering," Comput. Methods. Appl. Mech. Eng., *28,* 179 (1981).
12. Aris, R., *The Mathematical Theory of Diffusion and Reaction in Permeable Catalysts,* Clarendon Press, Oxford (1975).
13. Bird, R. B., W. E. Stewart, and E. N. Lightfoot, *Transport Phenomena,* Wiley, New York (1960).
14. Poots, J., "Compressible Laminar Boundary-Layer Flow at a Point of Attachment," J. Fluid Mech., *22,* 197 (1965).

## BIBLIOGRAPHY

*For additional or more detailed information, see the following:*

Becker, E. B., G. F. Carey, and J. T. Oden, *Finite Elements: An Introduction,* Prentice-Hall, Englewood Cliffs, N.J. (1981).

deBoor, C., *Practical Guide to Splines,* Springer-Verlag, New York (1978).

Fairweather, G., *Finite Element Galerkin Methods for Differential Equations,* Marcel Dekker, New York (1978).

Russell, R. D., *Numerical Solution of Boundary Value Problems,* Lecture Notes, Universidad Central de Venezuela, Publication 79-06, Caracas (1979).

Strang, G., and G. J. Fix, *An Analysis of the Finite Element Method,* Prentice-Hall, Englewood Cliffs, N.J. (1973).