Chapter 1

Boundary Value Problems for Ordinary Differential Equations

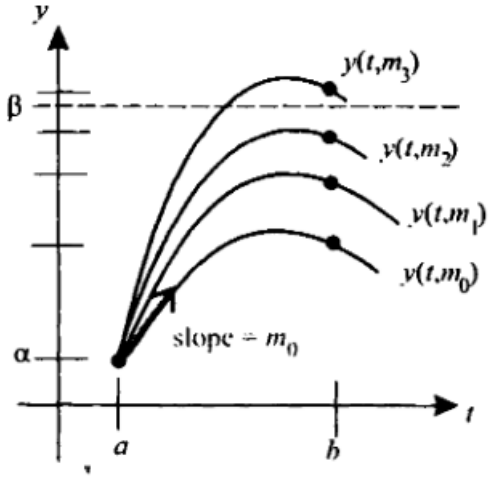
1.1. THE NONLINEAR SHOOTING METHOD

This method will really appear more like we are "shooting" at the solution than was the case with the linear shooting method of the last section. We once again turn to the general BVP (1):

$$\begin{cases} y''(t) = f(t,y,y') & (DE) \\ y(a) = \alpha, y(b) = \beta & (BCs) \end{cases}$$

Recall that when the DE was linear, we obtained the solution of the BVP as a linear combination of two solutions of (just) two specially associated IVPs. In the nonlinear case, we will solve a sequence of related IVPs:

$$(IVP)_k \begin{cases} y''(t) = f(t,y,y') \\ y(a) = \alpha, y'(a) = m_k \end{cases} \qquad \begin{matrix} same\,(DE) \\ (ICs) \end{matrix} \Rightarrow \; solution\; y_k(t) \equiv y(t,m_k)$$



Each of the IVPs above is identical, except for the second initial condition $y(a)' = m_k$,where the parameter will be appropriately adjusted ("aimed') at each iteration. We have denoted the solution of $(IVP)_k$ as $y_k(t)$ and , since it depends on m_k, we have introducet the function of two variables $y(t,m_k) \equiv y_k(t)$. The method is roughly illustrated and explained in Figure 10.3.

Figure 1.1: Illustration of the nonlinear shooting method for a BVP:
$\begin{cases} y''(t) = f(t,y,y') \\ y(a) = \alpha, y(b) = \beta \end{cases}$ The initial approximation $y_0(t) = y(t,m_0)$ is the solution of the corresponding IVP having the same DE, the same first condition, and satisfying the initial slope $y_0'(t) = m_0$,obtained numerically by methods of the last chapter. The desired second boundary condition is compared with $y_0(b)$, and ,if neccesery, this process is repeated with adjusted initial slopes $m_1, m_2, \cdots$ until we arrive at a solution that satisfies the second boundary condition(within a desired tolerance).
The only detail left to tend to is the important issue of how best to choose our initial slopes. It turns out to be a bit complicated; indeed, figuring out subsequent initial slopes will require solving an additional IVP. We outline the procedure now, give a specific example, and afterwards give a theoretical explanation of it.

The Nonlinear Shooting Method:

1. Start with an estimate (or guess) for the initial slope of the first IVP $m_0 = y_0'(a)$; a good default is the difference quotient $m_0 = \dfrac{\beta-\alpha}{b-a}$

2. Solve the associated $(IVP)_k \begin{cases} y''(t) = f(t,y,y') \\ y(a) = \alpha, y'(a) = m_k \end{cases}$ $(k=0)$ *on* $a \le t \le b$ using, say, the Runge-Kutta method; denote the solution as $y_k(t) = y(t,m_k)$ $(k=0)$

3. Check for accuracy by evaluating: $Diff \equiv y(b,m_k) - \beta$. If $|Diff| <$ tolerance, accept $y_k(t) = y(t,m_k)$ as solution to BVP, otherwise update

$$m_{k+1} = m_k - \frac{Diff}{z(b,m_k)},$$

where $z(t,m_k)$ solves the IVP:

$$\begin{cases} z''(t) = zf_y(t,y,y') + z'f_y(t,y,y') \\ z(a) = 0, z'(a) = 1 \end{cases}$$

Increase k and return to step 2 to iterate this procedure.

NOTE: To numerically solve the IVP for z in step 3, we will need to do it in conjunction with the concurrent IVP $y(y_k)$ in step 2 since, in general, the DE of z involves y. Thus, we will have to solve the two IVPs simultaneously by writing them into an equivalent four-dimensional first-order system.

Example 1.1. Numerically solve the BVP: ■

$$\begin{cases} y''(t) = -2(yy' + ty' + y + t) & (DE) \\ y(1) = 0, y(2) = -2 & (BCs) \end{cases}$$

by using the nonlinear shooting method in conjunction with the Runge-Kutta method with step size $h = 0.01$.

(a) Do it first with a tolerance of 0.01. How many "shots" were required? Get MATLAB to display the totality of graphs of the functions $y_k(t) = y(t,m_k)$("*shots*") in the same plot with the final one in a different color from the rest.

(b) Next do it for a tolerance of 10^7. How many "shots" were required? This time, using the subplot command, display the plots of the successive difference errors $|y_{k+1}(t) - y_k(t)|\, for k = 0,1,2,3,\ldots$

SOLUTION: We point out the DE is nonlinear (because of the yy' term) and so the linear shooting method would not be applicable. The associated initial value problems for y are

$$(IVP)_k \begin{cases} y''(t) = f(t,y,y') \equiv -2(yy' + ty' + y + t) \\ y(1) = 0, y'(1) = k_k \end{cases}$$

By introducing the new function $yp(t) = y'(t)$ we can translate this IVP into the following equivalent system:

$$(IVP)_k' \begin{cases} y'(t) = yp, & y(1) = 0 \\ yp'(t) = -2(y(yp) + t(yp) + y + t), & yp(1) = m_k \end{cases}$$

To get the IVP for the auxiliary function z, we compute

$$f_y(t,y,y') = -2y' - 2 \quad f_{y'}(t,y,y') = -2y - 2t,$$

which brings us to the following companion IVP for z:

$$\begin{cases} z''(t) = z(-2y' - 2) + z'(-2y - 2t) \\ z(1) = 0, z'(1) = 1 \end{cases}$$

By introducing the new function $zp(t) = z'(t)$ and combining this IVP with the previous one, we arrive at the following four-dimensional system:

$$\begin{cases} y'(t) = yp, & y(1) = 0 \\ yp'(t) = -2(y(yp)+t(yp)+y+t), & yp(1) = m_k \\ z'(t) = zp, & z(1) = 0 \\ zp'(t) = -2(yp+1)z-2(y+t)zp, & zp(1) = 1 \end{cases}$$

For the initial slope we use the suggested default $m_0 = \dfrac{y(2)-y(1)}{2-1} = \dfrac{-2-0}{2-1} = -2$.
In turning the problem over to MATLAB, since we plan to make use of the `rksys` routine of the last chapter, we must first construct the vector-valued function corresponding to the right sides of the four-dimensional system above. We do this in the following rather generic way that can be easily mimicked for any other nonlinear shooting problem:

```
function xp=nlshoot(t,x)
xp(1)=x(2);
xp(2)=-2*(x(2)*x(2)+t*x(2)+x(1)+t);
xp(3)=x(4);
xp(4)=-2*(x(2)+1)*x(3)-2*(x(1)+t)*z(4);
```

Note that we have identified $x(1)$ with y, $x(2)$ with yp, $x(3)$ with z and $x(4)$ with zp.

Part (a): We can now perform the desired plots using the following while loop.

```
>>mk=-2; \%initialize\\
>> while 1 \%since 1 is true, loopwill continue to execute
  [t,x]=rksys('nlshoot',1,2,[0 \ mk \ 0 \ 1],0.01);
  y=X(:1); z=X(:3); \%peel off the vectors we need
  Diff=y(101)+2; \%y(101) (MATLAB) corresponds to y(2) (Math)
  if abs(Diff)<0.01
    plot(t,y,'r'), return
    end
  plot(t,y,'b'), \ \ hold on
  n=n+1; \%bump counter up one
  mk=mk-Diff/z(101); \%update slope
end
```

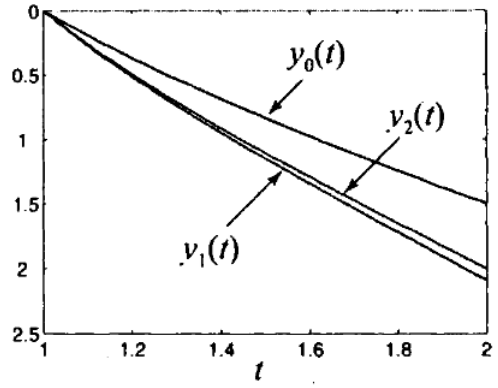


Figure 1.2: Illustration of the nonlinear shooting method applied to the BVP of Example 10.4(a). The successive approx- imations $y_k(t)$ are shown until the value of $y_k(2)$ is within a tolerance of 0.01 to - 2 , at which point the process grinds to a halt. The code is set up to graph the final approximation in red.

The plot shown in Figure 10.4 clearly shows that 3 iterations ("shots") were done: The first was too high, the second too low, and the third about right (within tolerance). Alternatively, by the way that the loop was set up, we could just enter n to query MATLAB to tell us how many iterations were done.

Part (b): We can easily modify the above loop to get the desired information and plots. We leave this as an exercise, but include the plot in Figure 10.5. We point out that in order to use the `subplot` command to get a decent plot, we first found out the number of shots needed and then ran through the loop again with an appropriately dimensioned "subplot" window. We also comment that a plot like the one in part (a) would be not quite so useful here since all approximations from the third onward are essentially indistinguishable using the graphs. This is why we look at successive differences. This is also a good way to check global errors.

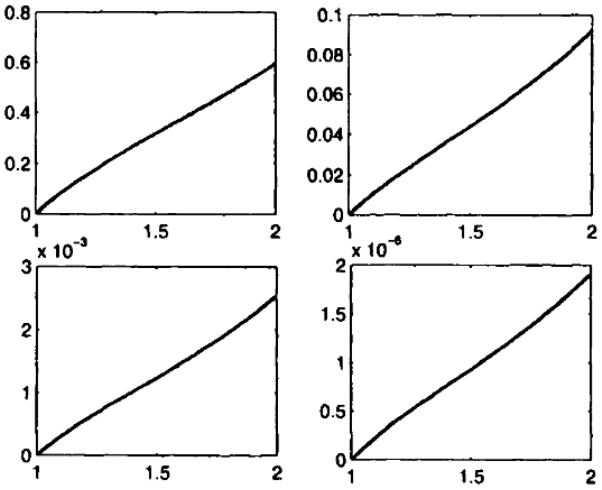


Figure 1.3: These graphs display the successive differences $|y_{k+1}(t) - y_k(t)|$ for nonlinear shooting method in part (b) of Example 10.4; this time the iterations continue until $y_k(2)$ gets within 10^{-7} of the desired value -2 and only 5 approximations ("shots") are needed.

EXERCISE FOR THE READER 1.1.

(a) Write a function M-file that will perform the nonlinear shooting method to numerically solve the BVP (1):

$\begin{cases} y''(t) = f(t,y,y') & (DE) \\ y(a) = \alpha, y(b) = \beta) & (BCs) \end{cases}$ The inputs and outputsshould be as follows:

```
[t, y, nshots] = nonlinshoot(a, alpha, b, beta, f, fy, fyp, tol, hstep)
```

The input and output variables more or less correspond to the data of the problem, but we leave the syntax open (see the suggestion for Exercise for the Reader 10.7). The variable *tol* will provide a stopping criterion for the iterations: $|y(b) - \beta| < tol$.[1] The last output variable is a positive integer giving the number of iterations (shots) that were executed. (b) Test your program by applying it to the two sets of data of Example 10.4. (c) Does Theorem 10.1 tell us anything about solutions for the following BVP?

$$\begin{cases} y''(t) = te^y - \sin(\cos(t))y' & (DE) \\ y(0) = 0, y(6) = 10 & (BCs) \end{cases}$$

Apply the nonlinear shooting method with tolerance $h = 0.01$ to solve this problem. Then repeat with tolerance 10^{-7} . Plot the final numerical solution and record the number of iterations.
Note: It is possible to write this program using MATLAB's Symbolic Toolbox capabilities and in this case the input variables fy and fyp could be dispensed with. The program we write in Appendix B does not use the Symbolic Toolbox; we leave such a construction to the interested reader.

As promised, we will now give a theoretical explanation of what has motivated the nonlinear shooting method. We assume that for any initial slope m, the IVP associated with the BVP (1),

$$\begin{cases} y''(t) = f(t,y,y') & same(DE) \\ y(a) = \alpha, y'(a) = m & (ICs)) \end{cases}$$

always has a unique solution on the time interval $[a,b]$, and we denote it by $y(t,m)$, which is a function of two variables. Our goal is to make m be a root of the equation

$$y(b,m) - \beta = 0 \tag{1.1}$$

In this equation we have held the t-variable of $y(t,m)$ to be fixed at $t = b$ so that the left side of (14) is a function of a single variable (namely m). Assuming it is differentiate, Newton's recursion formula for rootfinding (Chapter 6) suggests that it would be a good idea to define our sequence recursively using the following scheme:

$$m_k = m_{k-1} - \frac{y(b,m_{k-1}) - \beta}{\partial / \partial m\{y(b,m_{k-1})\}} \tag{1.2}$$

To go from one iteration to the next, after having (numerically) found $y(t,m_{k-1})$, the only difficult part of the formula (15) to obtain is the partial derivative $\partial / \partial m\{y(b,m_{k-1})\}$. This can be done (in an at first seemingly roundabout way) by finding an IVP for which the function

[1] Creation of this M-file will require features from MATLAB's Symbolic Toolbox; see Appendix A. Without the symbolic toolbox features, a similar program could be constructed but it would need more input variables, for example, $f_y(t,y,y')$ and $f_{y'}(t,y,y')$

$$z(t) = z(t,m) \equiv \frac{\partial y}{\partial m}(t,m)$$

is a solution and then numerically solving this IVP and evaluating it at $t = b$ to get the needed partial derivative. We can get a DE for $z(t,m)$ by differentiation of the DE for $y(t,m)$ and using the chain rule as follows (wherein we reserve primes (') for differentiations in the t-variable):

$y''(t,m) = f(t,y(t,m),y'(t,m)) \Rightarrow$
$\frac{\partial y''}{\partial m}(t,m) = f_t(t,y,y')\frac{\partial t}{\partial m} + f_y(t,y,y')\frac{\partial y}{\partial m}(t,m) + f_{y'}(t,y,y')\frac{\partial y'}{\partial m}(t,m)$

Since t and m are independent variables, $\partial t/\partial m = 0$ and so the first term on the right vanishes. We now simply replace $\partial t/\partial m(t,m)$ by $z(t,m)$ and the above becomes the following DE for z:

$$z''(t,m) = f_y z + f_{y'} z' \tag{1.3}$$

If we differentiate the initial conditions for $y(t,m)$: $\begin{cases} y(a,m) = \alpha \\ y'(a,m) = m \end{cases}$ we obtain corresponding initial conditions for $z(t,m)$:

$$\begin{cases} z(a,m) = 0 \\ z'(a,m) = 1 \end{cases} \tag{1.4}$$

Replacing the partial derivative in (15) by $z(b,m_{k-1})$ we see at once that (15), (16), and (17) yield the nonlinear shooting algorithm.

EXERCISES 1.1.

1. For each of the linear BVPs in parts (a) through (d) of Exercise 1, Section 10.2, apply the nonlinear shooting method to solve it via the Runge-Kutta method with step size $h = 0.01$ by following the outline below (if possible):

   (i) Write down the associated IVPs both for y and for the auxiliary function z.

   (ii) Translate both IVPs for y and z into a single four-dimensional IVP system of first-order DEs.

   (iii) Use MATLAB to apply the nonlinear shooting method to solve the BVP with a tolerance of 10^4. Display all of the approximations ("shots") in a single plot with the final one being displayed in a different color or plot style.

2. For each of the linear BVPs in parts (a) through (d) of Exercise 2, Section 10.2, repeat the instructions of the last exercise, but change item (iii) to: (iii'): Use MATLAB to apply the nonlinear shooting method to solve the BVP with a tolerance of 10^-6 . How many "shots" were required? Plot the final graph and also in a separate window and using the `subplot` command, get MATLAB to display the plots of the successive differences of the "shots": $|y_{k+1}(t) - y_k(t)|$ for $k = 0,1,2,3,...$

3. For each of the nonlinear BVPs given, perform the following tasks (if possible):

   (i) Write down the associated IVPs both for » and for the auxiliary function z.

   (ii) Translate both IVPs for y and z into a single four-dimensional IVP system of first-order DEs.

   (iii) Use MATLAB to apply the nonlinear shooting method to solve the BVP with a tolerance of 10^4. Display all of the approximations ("shots") in a single plot with the final one being displayed in a different color or plot style.

   (iv) Along with the BVP, an exact solution $f(t)$ is given; verify that this function actually solves the BVP.

   (v) Using a subplot window if you prefer, plot the errors of each of the successive shots with the exact solution given.

   (a) $\begin{cases} y'' = 12y^{5/3} \\ y(0) = 1, y(2) = 1/27 \end{cases}$, $f(t) = \frac{1}{(t+1)^3}$

   (b) $\begin{cases} y'' = -[y']^2/y \\ y(0) = 1, y(5) = 4 \end{cases}$, $(t) = \sqrt{3t+1}$

   (c) $\begin{cases} y'' = y' + 2(y - \ln t)^3 \\ y(1) = 1/2, y(2) = 1/2 + \ln 2 \end{cases}$, $f(t) = \frac{1}{t} + \ln t$

4. Repeat the instructions for Exercise 3 on the following BVPs.

   (a) $\begin{cases} y'' = y'\cos t - y\sin t \\ y(0) = 1, y(8\pi) = 1 \end{cases}$, $f(t) = \exp(\sin t)$

(b) $\begin{cases} y'' = y^3 - yy' \\ y(1) = 1/2, y(2) = 1/3 \end{cases}$, $f(t) = \frac{1}{t+1}$

(c) $\begin{cases} y'' = y'(\ln t + 1) + y(1 + 1/t) \\ y(1) = 1, y(5) = 3125 \end{cases}$, $f(t) = t^t$

5. Use the nonlinear shooting method to solve the following BVP. Your IVP solver should be Runge-Kutta with step size $h = 0.01$. Your tolerance (for the right BC) should be 0.0001. How many iterations did this take? Plot your solution. Also, what is the value of the solution when $x = 0.4$?

$$\begin{cases} y''(t) = -t(y')^3 \\ y(0) = 0, y(1) = \pi/2 \end{cases}$$

6. *(Physics: Flight of a Well-Hit Baseball)* This problem deals with the flight of a baseball in two dimensions (which we take for convenience as the xy-plane). We consider a ball that is hit so it lands 300 feet from home plate after 3 seconds. Many factors influence the flight of the ball. We assume that the air resistance acts only against the horizontal velocity, and for this particular baseball it is proportional to the horizontal velocity with exponent 1.2. Assuming the of home plate are $(x, y) = (0, 0)$, we let $x(t)$ and $y(t)$ denote the x and y coordinates of the position of the ball t seconds after it is hit. Thus, at time t, the coordinates of the baseball are $(x(t), y(t))$, $0 \le t \le 3$. The air resistance assumption and Newton's law from basic physics give the following system of second-order DEs:

$$\begin{cases} x''(t) = -cx(t)^(1.2) \\ y''(t) = -g \end{cases},$$

where for the ball being used the constant $c = 0.44$. The initial position of the ball is $(x(0), y(0)) = (0, 3)$ <feet>. Since the ball lands after 3 seconds we also get $(x(3), y(3)) = (300, 0)$.

(a) Explicitly find the function $y(t)$ just using basic calculus.

(b) Numerically find $x(t)$ by using the shooting method with step size $h = 0.01$ (and implementing the Runge-Kutta method), and sketch a plot of the path of the ball (i.e., of y vs. x). (For this part, you need not print the graph of x vs. t, or give any explicit values for $x(t)$.)

(c) After how many seconds does the ball reach its maximum height? At this time what is the x- coordinate?

(d) With the same hit, how far would the ball have gone (on the x-axis), if, as in the imaginary assumptions of physics courses, there was no air resistance?

7. *(Civil Engineering: Deflection of a Beam)* Use the nonlinear shooting method with $h = 0.01$ in the Runge-Kutta method to solve the exact beam-deflection model BVP:

$$\begin{cases} y''(t)/[1 + (y')^2]^{3/2} = \dfrac{T}{EI} y + \dfrac{wx(x-L)}{2EI} \\ y(0) = 0 = y(L) \end{cases},$$

having the parameters: L = 50 feet (length), T = 300 lb (tension at ends), w = 50 lb/ft (vertical load), $E = 1.2 \times 10^7$ lb/ft^2 (modulus of elasticity), and $I = 4$ ft^4 (central moment of inertia).
(a) Graph the resulting numerical solution, (b) How does the solution compare with that obtained for the corresponding linear approximating BVP of Example 10.3?

1.2. THE FINITE DIFFERENCE METHOD FOR LINEAR BVP'S

The method we present next is philosophically quite different from the shooting methods. It immediately discretizes the BVP by approximating the derivatives with difference quotients. The problem is then translated into a linear system that is easily solved directly. This method will pave the way for the corresponding finite difference methods that we will employ in the next two chapters for solving PDEs. There are analogues of this method for nonlinear BVPs; the discretization is done in the same way but the resulting system of equations will no longer be linear.[2]

All finite difference methods are based on approximating derivatives of a function by certain difference quotients. These difference quotient formulas can always be obtained using Taylor's theorem. We will be needing them only for first and second derivatives, and we now present them in the following lemma. To describe the error bounds, we employ the "big O" notation that was introduced in Section 8.3.

LEMMA 1.1. *(Central Difference Formulas)*

[2]For the nonlinear analogues of the finite difference method, we cite the reference: [BuFa-01] (see Section 11.3 therein.)

(a) If $f(x)$ is a function having a continuous second derivative in the interval $a-h \le x \le a+h$, then we have

$$f'(a) \approx \frac{f(a+h)-f(a-h}{2h}, \tag{1.5}$$

where the error of the approximation is $O(h^2)$.

(b) If, furthermore, f(x) has a continuous fourth derivative throughout $a-h \le x \le a+h$, then we also have the approximation

$$f''(a) \approx \frac{f(a+h)-2f(a)+f(a-h)}{h^2}, \tag{1.6}$$

and the error of this approximation is also $O(h^2)$.

Proof of part (a): Taylor's theorem allows us to write

$$f(a+h) = f(a)+hf'(a)+h^2f''(a)/2+O(h^3), \text{and}$$

$$f(a-h) = f(a)-hf'(a)+h^2f''(a)/2+O(h^3).$$

Subtracting the second of these equations from the first gives $f(a+h)-f(a-h) = 2hf'(a)+O(h^3)$ and solving this for $f'(a)$ produces (18). We have used the facts that $O(h^3)+O(h^3) = O(h^3)$ and $O(h^3)/h = O(h^2)$. The proof of part (b) is similar and is left as the next exercise for the reader.

EXERCISE FOR THE READER 1.2.

Prove part (b) of Lemma 10.3.

We now explain the finite difference method in more detail. Consider the linear BVP (5):

$$\begin{cases} y''(t) = p(t)y'+q(t)y+r(t) & (DE) \\ y(a) = \alpha, y(b) = \beta & (BCs) \end{cases}$$

Choose a positive integer N, and subdivide the interval $a < t < b$ into N equal subintervals using $N-1$ interior grid values: $t_1 = a+h, t_2 = a+2h, \cdots, t_{N-1} = a+(N-1)h$, where $h = (b-a)/N$. We also write $t_0 = a$ and $t_N = b$; see Figure 10.6. We let,

$$y_i = y(t_i) \ \ (0 \le i \le N),$$

and similarly,

$$p_i = p(t_i),\ q_i = q(t_i),\ r_i = r(t_i)\ (0 \le i \le N).$$

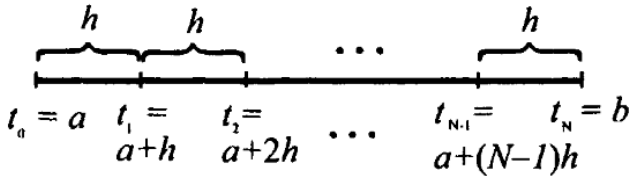


Figure 1.4: Grid value notation for the finite difference method for a BVP.

At each internal grid value $f_i(0 < i < N)$, we approximate the DE (5)

$$y''(t_i) = p(t_i)y'(t_i)+q(t_i)y(t_i)+r(t_i)$$

using the central difference formulas of Lemma 10.3, to obtain the approximation with local truncation error $O(h^2)$:

$$\frac{y_{i+1}-2y_i+y_{i-1}}{h^2} = p_i\frac{y_{i+1}-y_{i-1}}{2h}+q_iy_i+r_i \quad (0 \le i \le N) \tag{1.7}$$

Multiplying by h^2 , and then regrouping, we can rewrite each equation in (20) as:

$$y_{i+1}-2y_i+y_{i-1} = hp_i(y_{i+1}-y_{i-1})/2+h^2q_iy_i+h^2r_i, \text{or}$$

$$(1+p_ih/2)y_{i-1}-(2+h^2q_i)y_i+(1-p_ih/2)y_{i+1} = h^2r_i \quad (0 \le i \le N) \tag{1.8}$$

Since we know from the two BCs of (5) that

$$y_0 = \alpha, \quad y_N = \beta,$$

the equations of (21), form a linear system in the $N-1$ unknowns $y_1, y_2, \cdots, y_{N-1}$, which, when put in matrix form $AY = C$, has

$$A = \begin{bmatrix} -(2+h^2q_1) & 1-p_1h/2 & 0 & \cdots & 0 \\ 1+p_2h/2 & -(2+h^2q_2) & 1-p_2h/2 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & & 1+p_{N-2}h/2 & -(2+h^2q_{N-2}) & 1-p_{N-2}h/2 \\ 0 & \cdots & 0 & 1+p_{N-1}h/2 & -(2+h^2q_{N-2}) \end{bmatrix}$$

and

$$C = \begin{bmatrix} h^2r_1 - (1+p_1h/2)\alpha \\ h^2r_2 \\ \vdots \\ h^2r_{N-2h} \\ h^2r_{N-1} - (1-p_{N-1}h/2)\beta \end{bmatrix} \tag{1.9}$$

Notice the special form of the coefficient matrix A. Often in finite difference methods and in many other applications, the coefficient matrices that arise are of a similar **banded** form (i.e., nonzero entries lie entirely on a few diagonal bands). This special type of banded matrix is called a **tridiagonal matrix**. Banded matrices are special cases of what are called **sparse** matrices, which are matrices having the majority of the entries being zero. An $n \times n$ tridiagonal matrix has at most $3n-2$ nonzero entries (among its n^2 entries). Since large matrices often eat up a lot of memory with storage, it is often more expedient to deal with sparse matrices of specialized forms using specialized methods. To solve such tridiagonal systems, rather than Gaussian elimination, we will be using the so-called Thomas method, whose algorithm is given below:[3]

PROGRAM 1.1. *The Thomas method for solving tridiagonal systems of the form:*

$$\begin{bmatrix} d_1 & a_1 & 0 & 0 & 0 & \cdots & & 0 & 0 \\ b_2 & d_2 & a_2 & 0 & 0 & \cdots & & 0 & 0 \\ 0 & b_3 & d_3 & a_3 & 0 & & & \vdots & \vdots \\ 0 & 0 & b_4 & d_4 & a_4 & 0 & & & \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & & & \\ 0 & 0 & & & 0 & b_{n-2} & d_{n-2} & a_{n-2} & 0 \\ 0 & 0 & \cdots & & & 0 & b_{n-1} & d_{n-1} & a_{n-1} \\ 0 & 0 & \cdots & & & & 0 & b_n & d_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \\ \vdots \\ x_{n-2} \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \\ \vdots \\ c_{n-2} \\ c_{n-1} \\ c_n \end{bmatrix}.$$

```
function x = thomas(a,d,b,c)
%solves matrix equation Ax=c, where A is a tridiagonal matrix
%inputs: a=upper diagonal of matrix A a(n)=0, d=diagonal of A,
%b=lower diagonal of A, b(1)=0, c=right-hand side of equation
n=length(d);
a(1)=a(1)/d(1);
c(1)=c(1)/d(1);
for i=2:n-1
  denom=d(i)-b(i)*a(i-1);
  if (denom==0), error('zero in denominator'), end
  a(i)=a(i)/denom;
  c(i)=(c(i)-b(i)*c(i-1))/denom;
end
c(n)=(c(n)-b(n)*c(n-1))/(d(n)-b(n)*a(n-1));
x(n)=c(n);
for i=n-1:-1:1
  x(i)=c(i)-a(i)*x(i+1);
end
```

Example 1.2. Use the thomas program above to solve the tridiagonal system: ■

$$\begin{array}{rcrcrcrcl} 2x_1 & - & x_2 & & & & & = & 1 \\ -x_1 & + & 2x_2 & - & x_3 & & & = & 0 \\ & - & x_2 & + & 2x_3 & - & x_4 & = & 0 \\ & & & - & x_3 & + & 2x_4 & = & 1 \end{array}$$

[3] Banded and sparse matrices were studied in Chapter 7; in particular, the Thomas method was introduced in Exercise 9 of Section 7.5

```
>> a=[-1 -1 -1 0]; d=[2 2 2 2]; b=[0 -1 -1 -1]; c=[1 0 0 1];
>> format rat;
>> thomas(a,d,,b,c)
```

ans → 1 1 1 1 (This answer is easily checked.)

We now make some technical comments on implementing the finite difference method. In order to solve the system $AY = C$, we will need the coefficient matrix A of (22) to be nonsingular. In general this can fail, but the following theorem gives sufficient conditions to guarantee A's invertibility.

THEOREM 1.1.

Suppose that the functions $p(t), q(t),$ and$r(t)$ are continuous on $a \le t \le b$ and that $q(t) \ge 0$on this time interval. Then the linear system $AY = C$ where A and C are as in (22) will have a unique solution provided that $h < 2/M$, where $M = max\{|p(t)| : a \le t \le b\}$. The hypotheses guarantee that the coefficient matrix A will be **strictly diagonally dominant**, which means that

$$|a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}|. \tag{1.10}$$

In other words, the absolute value of any diagonal entry dominates the sum of the absolute values of all other entries in its row. Diagonally dominant matrices are always invertible [4], 4 and furthermore, it can be shown that the Thomas algorithm works very well in their presence. There are instances, however, where the Thomas method will fail for tridiagonal nonsingular matrices (e.g., this happens if $a_{11} = 0$), but there are ways to modify the method to deal with such cases.
Generally speaking, the linear shooting method is more efficient for solving a linear BVP when the former is coupled with the Runge-Kutta method. This is because the Runge-Kutta method has a local truncation error of $O(h^4)$ while that for the finite difference method is $O(h^2)$. Our main reason for introducing it here is to prime the way for its generalization to solving partial differential equations; the shooting methods do not naturally extend to the setting of PDEs.

EXERCISE FOR THE READER 1.3.

Show that under the conditions of Theorem 10.4, the matrix A of (22) is diagonally dominant.

Example 1.3. Use the finite difference method with $h = 0.1$ to solve the beam-deflection BVP of Example 10.3: ■

$$\begin{cases} y''(x) = \frac{T}{EI} y + \frac{wx(x-L)}{2EI} \\ y(0) = 0 = y(L) \end{cases}, 0 \le x \le L,$$

having the parameters L = 50 feet (length), T = 300 lb (tension at ends), w = 50 lbs/ft (vertical load), E = 1.2x$\times 10^7$ lb/ft^2 (modulus of elasticity), and I = 4 ft^4 (central moment of inertia).

(a) Do it first for N = 20 subdivisions to obtain the approximate solution yl and plot its graph.

(b) Redo it for both N = 40, and N = 80 subdivisions, to get approximate solutions y2,and y3.

SOLUTION: Since the scripts are similar, we present only the one for obtaining $y1$ when N = 20. The script is written in such a way as to be easily modified to work for any linear BVP. The graphs of the solutions look identical, so we present only the graph of $y1$, but give plots of the differences $y1 - y2$ and $y2 - y3$.

```
%MATLAB script for finite difference method for above problem.
xa=0; xb=50; n=20; h=(xb-xa)/n;x=h:h:(xb-h);
for i=1:n-1, a(i)=0;end, b=a;
a(1:n-2)=1-p(1:n-2)*h/2; %above diagonal band
d=-(2+h*h*q); %diagonal
b(2:n-1)=1+p(2:n-1)*h/2; %below diagonal band
c(2:n-2)=h*h*r(2:n-2);
c(1)=h*h*r(1)-(1+p(1)*h/2)*ya; c(n-1)=h*h*r(n-1)-(1-p(n-1)*h/2)*yb;
y=thomas(a,d,b,c);
X=(xa x xb);
Y=(ya y yb);
plot(X,Y), grid on
```

[4]Proof: Suppose that A is diagonally dominant. If A were not invertible, then there would exist a nonzero vector x such that $Ax = 0$. Let k be an index so that the absolute value $|x_k|$ is as large as possible (in the norm notation from Section 7.6, this would mean $|x_k| = ||x||_\infty$) Take the kth equation of $Ax = b : \sum_{j=1}^{n} a_{kj} x_j = 0$, divide by x_k, and solve for a_{kk} to get a $a_{kk} = -\sum_{j=1, j\neq k}^{n} a_{kj} \cdot (x_j/x_k)$. Now take absolute values and use the triangle inequality to get $|a_{kk}| \le \sum_{j=1, j\neq k}^{n} |a_{kj} \cdots (x_j/x_k)| \le \sum_{\substack{j=1 \\ j\neq k}}^{n}$. Wh at we now have contradicts diagonal dominance of the matrix A, so we have proved that A must indeed be invertible.

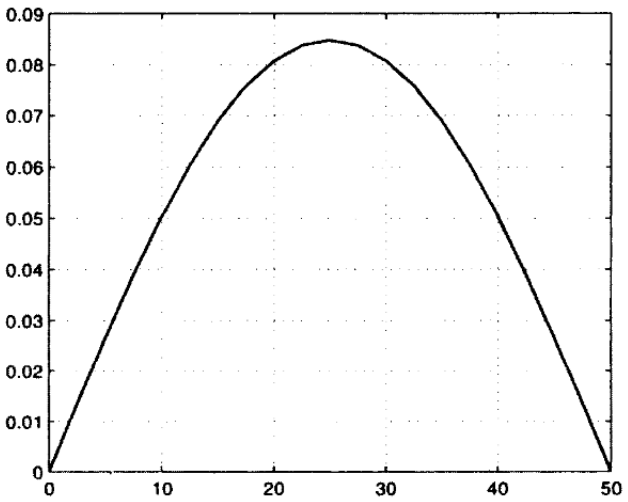


Figure 1.5: Graph of the solution of the beam-deflection problem of Example 10.6 using N=20 subdivisions.

We can now store these x- and y-values as xl and yl by entering:

```
>> xl=X; yl=Y;
```

and next go on to slightly change the script to do $N = 40$ iterations and then $N = 80$ iterations and store the corresponding x and y-values as x2, y2 and x3, y3 respectively. You will notice that the graphs look quite identical. To plot the differences: yl - y2 , and y2 - y3 , one must be a bit careful since yl , y2, and y3 all have different lengths. Each has $N + 1$ components (N - 1 grid points + the two boundary points). Here is one strategy to plot yl - y2 versus x. The grid points for y2 consist of those of yl plus one extra grid point between each adjacent pair of grid points for yl (located at the midpoint). We must reformulate y2, only at the grid values for yl (throwing away the extra ones at the midpoints). Let's call this "trimmed down" version of y2 by y2 trim . To form y2 trim in MATLAB, we could use the following line:

```
>> for i=1:21, y2trim(i)=y2(2*i-1); end %alternatively: y2trim = y2
   (1:2:41)
```

Now we can plot the difference of yl - y2 by simply entering:

```
>> plot (x1,y1-y2trim)
```

In a similar fashion, the following commands give the plot of the difference y2 - y3:

```
>> for i=1:41, y3trim(i)=y3(2*i-1); end
>> plot(x1,y2-y3trim)
```

See Figure 10.8 for both of these error plots. Both scripts finished on the author's computer in less than a second, and the differences are quite small. We leave it to you to see what happens if one continues this by repeatedly doubling the number on subintervals N. N= 160, N=320, N=640,....

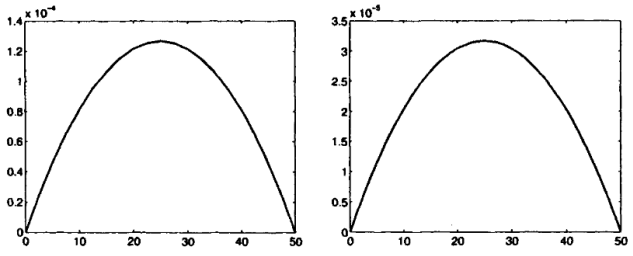


Figure 1.6: Plots of differences of finite difference approximated solutions to the deflected beam problem of Example 10.6. (a) The left graph is of the difference of the $N = 20$ and $N = 40$ interior grid point solutions and (b) the right one is the graph of the difference of the $N = 40$ and $N = 80$ interior grid point solutions.

EXERCISES 1.2.

1. For each of the linear BVPs of parts (a) through (d) of Exercise 1 of Section 10.2, use the finite difference method with $N = 100$ to solve and then plot the solution. Whenever possible, use the Thomas method to solve the tridiagonal system. If the coefficient matrix fails to be invertible (so that errors will come up with both the Thomas and the Gaussian methods), try bumping N up to 500.

2. For each of the linear BVPs of parts (a) through (d) of Exercise 2 of Section 10.2, use the finite difference method with $N = 100$ to solve and then plot the solution. Whenever possible, use the Thomas method to solve the tridiagonal system. If the coefficient matrix fails to be invertible (so that errors will come up with both the Thomas and the Gaussian methods), try bumping N up to 500.

3. For each of the BVPs and corresponding general solutions for the DEs given in parts (a) through (c) of Exercise 3 of Section 10.2, do the following (if possible): (i) Use the finite difference method with $N = 100$ to solve and store the solution in vectors $t1$, $y1$. (ii) Repeat with $N = 500$ and store the solution in vectors $t2$, $y2$. (iii) Repeat once again with $N = 2500$ and store the solution in the vectors $t3$,$y3$. (iv) Determine the constants in the general solution given so that it solves the given BVP. (v) Plot the four curves in the same graph using different plot colors/styles for each. In situations where graphs are indistinguishable, plot also the errors (differences of approximations with exact solutions). Whenever possible, use the Thomas method to solve the tridiagonal system.

4. Repeat all parts of the previous exercise for each of the BVPs and general solutions given in parts (a) through (c) of Exercise 4 of Section 10.2.

5. A thin rod of length L is insulated along the lateral surface but kept at temperature $T = 0$ at both ends $x = 0$ and $x = L$. The rod has a heat source which is proportional to the temperature at cross-section x with proportionality constant Q, The steady-state temperature function $T(x)$ $0 \leq x \leq L$ then satisfies the DE:

$$T_{xx} + QT = 0, T = T(x), 0 \leq x \leq L.$$

(See Section 11.2 for a derivation of more general heat equations.) Solve this DE with the given BC's $T(0) = T(L) = 0$ using the finite difference method with $N = 20$. Repeat with $N = 40, N = 60$, and $N = 120$. Plot these four approximations together (using different plot styles/colors). In cases where two are indistinguishable, plot the corresponding successive differences.
(See Section 11.2 for a derivation of more general heat equations.) Solve this DE with the given BC$'$ s $T(0) = T(L) = 0$ using the finite difference method with $N = 20$. Repeat with $N = 40, N = 60$, and $N = 120$. Plot these four approximations together (using different plot styles/colors). In cases where two are indistinguishable, plot the corresponding successive differences.

6. The general solution of the DE $y'' = -y$ is $y = A\sin t + B\cos t$.

   (a) What restriction (on the parameters A and B) does the condition $y(0) = 0$ place?

   (b) For which values of $L > 0$ does the BVP consisting of the DE and the BC's $y(0) = y(L) = 0$ have a solution (existence)? For such values of L, show that the solution is not unique.

   (c) Use $L = 1$ in the BVP of part (b) and apply the finite difference method with $N = 20$. What happens? Does the Thomas algorithm work? If not, try Gaussian elimination. Is the coefficient matrix nonsingular?

   (d) Repeat part (c) using $L = \pi$.

7. (a) Use Taylor's theorem to establish the following fourth-order central difference formula:

$$f'(a) \approx \frac{-f(a+2h) + 8f(a+h) - 8f(a-h) + f(a-2h)}{12h}$$

with error $O\left(h^4\right)$, provided that $f^{(5)}(x)$ is continuous in the interval $a - 2h \leq x \leq a + 2h$.

   (b) In the same fashion, derive the fourth-order central difference formula

$$f''(a) \approx \frac{-f(a+2h) + 16f(a+h) - 30f(a) + 16f(a-h) - f(a-2h)}{12h}$$

with error $= O\left(h^4\right)$, provided that $f^{(6)}(x)$ is continuous in the interval $a - 2h \leq x \leq a + 2h$.

1.3. THE RAYLEIGH-RITZ METHOD

Figure 1.7: Johny William Strutt (Lord Rayleigh) (1942-1919), English physicist and mathematician.

The material of this section contains much more theory than a typical section of the text. The ideas contained herein come from an important and very beautiful area of mathematics which blends linear algebra and analysis. It is fair to say that this area gave birth to the subject of functional analysis. Furthermore, the generalization of the Rayleigh-Ritz method to higher dimensions gives rise to the very important finite element method (Chapter 13) for numerical solution of PDEs. As the language in the development will indicate, many of the concepts leading to the Rayleigh-Ritz method are motivated by concepts in physics. Indeed, this was the motivational setting that led to its development. Despite the fact that the RayleighRitz method [5] dates back to the beginning of the twentieth century, it took another half century before the finite element method came to fruition. The basic idea of the Rayleigh-Ritz method is that a boundary value problem can be recast as a certain minimization problem.

Rather than strive for generality, our purpose in this section will be to understand the concepts behind the Rayleigh-Ritz method so we begin by focusing our attention on the following boundary value problem:

$$\text{(BVP)} \begin{cases} -u''(x) = f(x), 0 < x < 1 \\ u(0) = 0, u(1) = 0 \end{cases} \tag{1.11}$$

Here $f(x)$ is a continuous function. This problem has, by itself, numerous physical interpretations, as we have seen in previous chapters. As examples we mention the steady-state heat distribution on a thin rod (Chapter 11) with ends maintained at temperature zero, or the deflection of an elastic beam (Section 10.1) whose ends are fixed.

We introduce the **inner product** $\langle u, v \rangle$ for a pair of piecewise continuous bounded functions on $[0,1]$:

$$\langle u, v \rangle = \int_0^1 u(x)v(x)dx. \tag{1.12}$$

Recall that for a function $u(x)$ to be piecewise continuous on $[0,1]$, it means that the domain can be broken up into subintervals: $0 = a_0 < a_1 < \cdots < a_n = 1$ such that $u(x)$ is continuous on each open subinterval (a_{i-1}, a_i). We point out the following simple yet very important properties of this inner product. By linearity of the integral, it immediately follows that the inner product is linear in each variable, i.e.,

$$\begin{aligned} \langle \alpha u_1 + \beta u_2, v \rangle &= \alpha \langle u_1, v \rangle + \beta \langle u_2, v \rangle \\ \langle u, \alpha v_1 + \beta v_2 \rangle &= \alpha \langle u, v_1 \rangle + \beta \langle u, v_2 \rangle \end{aligned}, \tag{1.13}$$

where the u, u_i, v, v_i denote arbitrary (piecewise continuous bounded) functions and α, β denote arbitrary real numbers. Even clearer is the following symmetry property:

$$\langle u, v \rangle = \langle v, u \rangle \tag{1.14}$$

[5] Despite family attempts to dissuade him from vigorously pursuing a career as a full-time scientist, Lord Rayleigh (who succeeded to the title at age 30) was so intrigued by the mysteries of physics and the power of mathematics, that he made a firm commitment not to let his official diplomatic and social functions interfere too much with his dedication to scientific inquiry. For most of his life, he was financially independent, and this allowed him to set up a personal laboratory in his estate and gave him more time to focus on his research without the distraction of the other duties associated with an academic post. For the periods that he did hold academic posts at Cambridge, he took his duties with utmost conscientiousness and made some very lasting improvements in the university's scientific programs. Lord Rayleigh was a model scientist; his work touched upon and connected many areas (the Rayleigh-Ritz method is a good example inside mathematics) and was extensive (446 publications), and he won numerous prizes and recognitions for his work. Beside his scientific prowess, he was also a kind, modest, and generous man. When he won the Nobel Prize in physics in 1904, he donated his prize money to Cambridge University for the purpose of building more laboratories. In 1902, in his acceptance speech for the National Order of Merit, he stated "... the only merit of which I personally am conscious was that of having pleased myself by my studies, and any results that may be due to my researches were owing to the fact that it has been a pleasure for me to become a physicist."

Walter Ritz (1878-1909) was a Swiss/German mathematician/physicist. After entering the Polytechnic University of Zurich in an engineering program, he found that he was not satisfied with the compromises and lack of rigor in his engineering courses, so he switched to physics. He was a classmate of Albert Einstein. For health reasons, he needed to move away from the humid climate of ZÃijrich, and went on to the University GÃűttingen to complete his studies. There he was influenced by the teachings of David Hubert. Despite his short life and career, he was able to accomplish quite a lot of scientific research. Actually, Lord Rayleigh and Ritz never met. Rayleigh first developed a mathematical method for predicting the first natural frequency of simple structures by minimizing the distributed energy. Ritz subsequently extended the method to solve (numerically) associated displacement and stress functions.

In light of properties (26) and (27), the inner product is said to be a **symmetric bilinear form**. Another property of the inner product is that it is **positive definite**: If $u(x)$ is a piecewise continuous function on $[0,1]$ that is not zero on some open interval (a_{i-1},a_i), then $\langle u,u\rangle > 0$ (see Exercise 17).

We consider the following rather large class of **admissible functions** on $[0,1]$ which obey the boundary conditions of our problem (24) :

$$\mathscr{A} = \{v : [0,1] \to \mathbb{R} : v(x) \text{ is continuous, } v'(x) \text{is piecewise continuous and bounded, and } v(0) = 0, v(1) = 0\}. \quad (1.15)$$

EXERCISE FOR THE READER 1.4.

Show that the space $\mathscr{A}$ is closed under the operations of addition of functions and scalar multiplication. More precisely, if $v,w \in \mathscr{A}$ and α is any real number, show that the functions $v+w$, and αv also belong to $\mathscr{A}$.
For functions in this class we further define the following functional: $F : \mathscr{A} \to \mathbb{R}$ by the formula:

$$F(v) = \frac{1}{2}\langle v', v'\rangle - \langle f, v\rangle \quad (1.16)$$

In the setting where (24) models the deflection of an elastic beam, certain physical interpretations can be given to some of these quantities. For a given displacement $v(x)$, the inner product $\langle f,v\rangle$ represents the so-called **load potential** and the term $\frac{1}{2}\langle v', v'\rangle$ represents the internal **elastic energy**. The functional $F(v)$ then represents the **total potential energy**. Using physics it can be proved that the solution of (24) will have minimal total potential energy over all possible admissible functions $v \in \mathscr{A}$. This fact is known as the **Principle of Minimum Potential Energy (MPE)** and we will prove it mathematically in Theorem 10.5 below. Thus, the variational problem which turns out to be equivalent to the boundary value problem (24) is the following:

$$\text{(MPE) Find } u \in \mathscr{A} \text{ satisfying } F(u) \le F(v) \text{ for all } v \in \mathscr{A}. \quad (1.17)$$

Another equivalent, but very different looking problem whose equivalence to the boundary value problem is known in physics as **Principle of Virtual Work (PVW)**, is the following:

$$\text{(PVW) Find } u \in \mathscr{A} \text{ satisfying } \langle u', v'\rangle = \langle f, v\rangle \text{ for all } v \in \mathscr{A} \quad (1.18)$$

It is quite a surprising fact that the three seemingly different problems (24), (30), and (31) have equivalent solutions. The precise result is stated in the following theorem.

THEOREM 1.2. *(Variational Equivalences of a Boundary Value Problem)*

Suppose that $f(x)$ is any continuous and bounded function on $0 < x < 1$, and tha $u(x)$ is an admissible function of the class $\mathscr{A}$ defined in (28). Then the followin are equivalent:

(a) The function $u(x)$ is a solution of the (BVP) (24) $\begin{cases} -u''(x) = f(x), 0 < x < 1 \\ u(0) = 0, u(1) = 0 \end{cases}$.

(b) The function $u(x)$ is a solution of the (MPE) (30): $F(u) \le F(v)$ for all $v \in \mathscr{A}$.

(c) The function $u(x)$ is a solution of the (PVW) (31) : $\langle u', v'\rangle = \langle f, v\rangle$ for all $v \in \mathscr{A}$.

Furthermore, each of these three problems has unique solutions.

Proof: The proof is rather long, so we break it up into several pieces. The proof that (24) has a unique solution can be accomplished quite easily (see Exercise 22). We point out that Theorem 10.1 does not apply.[6]

Step 1: We first show that (b) implies (c). To this end, suppose that $u(x)$ solves the (MPE), so that $F(u) \le F(v)$ for all $v \in \mathscr{A}$. Letting ε denote any real number, we may conclude that $F(u) \le F(u+\varepsilon v)$, where $v \in \mathscr{A}$ is arbitrary. If we hold the functions u and v fixed, we can view the function on the right $\phi(\varepsilon) \equiv F(u+\varepsilon v)$ as a real-valued function of

[6] A general result shows that existence and uniqueness questions about general BVPs can be reduced to questions about homogeneous BVPs. The following is taken from page 197 of [Sta-79]: **Theorem** For a pair of 2×2 matrices $\mathscr{A}$ and B, and continuous functions $f(x), g(x), h(x)$ on an interval $[a,b]$, the BVP consisting of the DE $y'' = h(x)y' + g(x)y + f(x)$ $(y = y(x))$ and the general boundary conditions $A\begin{bmatrix} y(a) \\ y'(a) \end{bmatrix} + B\begin{bmatrix} y(b) \\ y'(b) \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$ has a unique solution if and only if the corresponding homogeneous problem with $f(x) = 0$, and $\alpha, \beta = 0$ has only the trivial solution $y(x) = 0$. For our special problem (24) we need only take $h(x) = g(x) = 0$ to get the DE and $\mathscr{A} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$, $B = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$ to get the BC 's. The corresponding homogeneous problem is just: $y'' = 0$ and $y(0) = y(1) = 0$. Integrating this DE and using the BC's easily shows $y(x) = 0$ is the only solution. Thus, this theorem implies our problem (24) has a unique solution.

ε. Using bilinearity and then symmetry of the inner product, we may expand this function as follows:

$$\begin{aligned}\phi(\varepsilon) &\equiv \frac{1}{2}\left\langle (u+\varepsilon v)', (u+\varepsilon v)'\right\rangle - \langle f, u+\varepsilon v\rangle \\ &= \frac{1}{2}\left\langle u'+\varepsilon v', u'+\varepsilon v'\right\rangle - \langle f,u\rangle - \varepsilon\langle f,v\rangle \\ &= \frac{1}{2}\left\langle u',u'\right\rangle + \frac{\varepsilon}{2}\left\langle u',v'\right\rangle + \frac{\varepsilon}{2}\left\langle v',u'\right\rangle + \frac{\varepsilon^2}{2}\left\langle v',v'\right\rangle - \langle f,u\rangle - \varepsilon\langle f,v\rangle \\ &= \frac{1}{2}\left\langle u',u'\right\rangle + \varepsilon\left\langle u',v'\right\rangle + \frac{\varepsilon^2}{2}\left\langle v',v'\right\rangle - \langle f,u\rangle - \varepsilon\langle f,v\rangle.\end{aligned}$$

Since each of the inner products in the last expression is simply a real number, the function $\phi(\varepsilon)$ is just a second-degree polynomial (in the variable ε). Since we know this function has a minimum value at $\varepsilon = 0$, we must have $\phi'(0) = 0$. Differentiating the last expression for $\phi(\varepsilon)$ in the above expansion, this gives $\langle u', v'\rangle - \langle f, v\rangle = 0$, and since $v \in \mathscr{A}$ was arbitrary, this shows (PVW). *Step 2*: We show that (c) implies (b). So assume that $u(x)$ solves the (PVW), i.e., $\langle u', v'\rangle = \langle f, v\rangle$ for all $v \in \mathscr{A}$. Fix now an admissible function $v \in \mathscr{A}$. Our task is to show that $F(v) \geq F(u)$. Setting $w = v - u$ so $v = u + w$, we may use bilinearity and symmetry as above to write:

$$\begin{aligned}F(v) &= F(u+w) \\ &= \frac{1}{2}\left\langle u'+w', u'+w'\right\rangle - \langle f, u+w\rangle \\ &= \underbrace{\frac{1}{2}\left\langle u',u'\right\rangle - \langle f,u\rangle}_{=F(u)} + \underbrace{\left\langle u',w'\right\rangle - \langle f,w\rangle}_{=0 \text{ by (PVW)}} + \underbrace{\frac{1}{2}\left\langle w',w'\right\rangle}_{\geq 0} \\ &\geq F(u),\end{aligned}$$

as desired. *Step 3*: We show that (a) implies (c). We thus assume that the function $u(x)$ solves the BVP (24). From the differential equation $-u''(x) = f(x), 0 < x < 1$, the second derivative of $u(x)$ exists (and is continuous) so it follows that the first derivative $u'(x)$ is continuous (from calculus, differentiability implies continuity). Furthermore, since $f(x)$ is assumed to be bounded, so must be $u'(x)$, and from the boundary conditions stipulated by (24), it follows that $u(x)$ is an admissible function (i.e., $u \in \mathscr{A}$). We now fix an admissible function $v \in$ and proceed to integrate by parts. Doing this and translating into inner products gives:

$$\langle f, v\rangle = \left\langle -u'', v\right\rangle = -\int_0^1 u^n(x)v(x)dx = \underbrace{u'(x)v(x)\big]_{x=1}^{x=0}}_{=0 \text{ by (BC)}} + \int_0^1 u'(x)v'(x)dx = \left\langle u', v'\right\rangle$$

It follows that $u(x)$ solves the (PVW), as asserted.

Up to this point we have rigorously shown the following implications for solutions of the various three problems:

$$(BVP) \Rightarrow (PVW) \Leftrightarrow (MPE).$$

We will next show that the solutions of (PVW) are unique. From this and what was already proved, it will follow that all three problems have unique solutions.

Step 4: We prove that any two solutions u_1 and u_2, both belonging to $\mathscr{A}$, of the problem (PVW) must be identical. Thus we are assuming that $\langle u_i', v'\rangle = \langle f, v\rangle$ for all $v \in \mathscr{A}$(i = 1,2). Our task is to show $u_1 = u_2$. If we use $v = u_1 - u_2 \in \mathscr{A}$, we obtain that: $\left\langle u_1', [u_1-u_2]'\right\rangle = \langle f, u_1 - u_2\rangle$ and $\left\langle u_2', [u_1-u_2]'\right\rangle = \langle f, u_1-u_2\rangle$, Subtracting and using linearity gives us: $\left\langle [u_1-u_2]', [u_1-u_2]'\right\rangle = 0$, which translates to $\int_0^1 (u_1'(x) - u_2'(x))^2\,dx = 0$. Since the integrand is nonnegative and piecewise continuous, it follows that it must equal zero everywhere on $[0,1]$ except, possibly, at the endpoints of the intervals making up its pieces. We have used positive definiteness of the inner product here. The same is therefore true for $u_1' - u_2' = [u_1 - u_2]'$, so it follows that the antiderivative of this latter function must be a constant. Thus we can write $u_1 - u_2 = C$ or $u_1 = u_2 + C$. But the boundary conditions $u_i(0) = 0$ then force $C = 0$ and we can conclude $u_1 = u_2$, as desired.

Step 5: (Final Step) We show that (PVW) implies (BVP). At this point we invoke the fact, mentioned at the outset of this proof, that the (BVP) has a solution $u(x)$ (existence). From what was already proved, this function $u(x)$ is also a solution of (PVW), but from step 4 , the solution of (PVW) is unique. Consequently, any solution of (PVW) really must be the (unique) solution of (BVP), as required. QED

In order to solve the BVP (24), the above theorem allows us to focus our attention on either of the equivalent problems MPE (30) or PVW (31). The finite element method will use one of these two formulations but will replace the very large space $\mathscr{A}$ of admissible functions by a much smaller (finite-dimensional) space in each of the corresponding

governing conditions. We begin by partitioning the interval $(0,1)$ into subintervals: $\mathscr{P}: 0 = x_0 < x_1 < \cdots < x_{n+1} = 1$. We denote these intervals by $I_i = (x_i, x_{i+1})$ $(i = 0,1,2,\cdots,n)$ and their lengths by $h_i = x_{i+1} - x_i$. Unlike with finite difference methods, we do not require that these lengths be equal. We define the **mesh size** $\|\mathscr{P}\|$ of this partition as the maximum of the lengths $\max_{0\le i\le n} h_i$. Corresponding to such a partition $\mathscr{P}$ we define the following space of piecewise linear functions:

$$\mathscr{A}(\mathscr{P}) = \{v : [0,1] \to \mathbb{R} : v(x) \text{ is continuous on } [0,1], \text{linear on each } I_i \text{ and } v(0) = 0, v(1) = 0\}.$$

A typical function in this space is depicted in Figure 10.10.

EXERCISE FOR THE READER 1.5.

Show that the space $A(\mathscr{P})$ is closed under the operations of addition of functions and scalar multiplication. More precisely, if $v, w \in \mathscr{A}(\mathscr{P})$ and α is any real number, show that the functions $v+w$, and αv also belong to $\mathscr{A}(\mathscr{P})$.

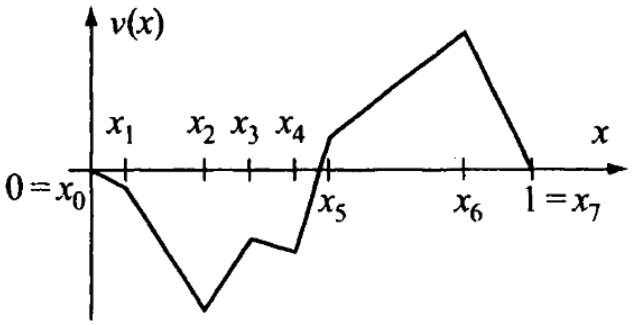


Figure 1.8: llustration of a typical function in the space $\mathscr{A}(\mathscr{P})$.

Notice that a function $v \in \mathscr{A}(\mathscr{P})$ is entirely determined by its values at the interior grid points: $v(x_1), v(x_2), \cdots, v(x_n)$. This follows from linearity and continuity. We need a set of basis functions that can be used to easily describe functions in $\mathscr{A}(\mathscr{P})$. These n functions are usually chosen so that each one equals zero on most of the interval $[0,1]$, so that it will have minimum interaction with other basis functions. [7] One simple set of basis functions meeting this criterion are the so-called **hat functions** $\phi_i(x)(1 \le i \le n)$. Each hat function $\phi_i(x)$ is that member of $\mathscr{A}(\mathscr{P})$ determined by the assignments: $\phi_i(x_i) = 1$ and $\phi_i(x_j) = 0$ for any index $j \ne i$. A typical hat function is shown in Figure 10.11.

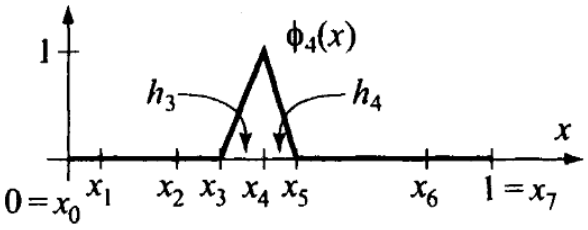


Figure 1.9: A typical hat function for a certain partition of $(0,1)$. Note the (possible) asymmetry.

We observe that any function $v \in \mathscr{A}(\mathscr{P})$ can be expressed in a unique way as a linear combination of the hat functions:

$$v(x) = \sum_{i=1}^{n} v(x_i)\,\phi_i(x). \tag{1.19}$$

(To prove this, just check that both functions agree at each partition point x_i, and then it will follow that they are always equal since both are piecewise linear.) In the language of linear algebra, we say that the n hat functions form a basis for the n-dimensional space $\mathscr{A}(\mathscr{P})$.

The equations of the hat functions are as follows:

$$\phi_i(x) = \begin{cases} 0, & \text{if } 0 \le x \le x_{i-1} \text{ or } x_{i+1} \le x \le 1, \\ \frac{x - x_{i-1}}{h_{i-1}}, & \text{if } x_{i-1} \le x \le x_i, \\ \frac{x_{i+1} - x}{h_i}, & \text{if } x_i \le x \le x_{i+1}. \end{cases} \tag{1.20}$$

[7] General Rayleigh-Ritz methods result from using any set of linearly independent functions which are continuous, piecewise differentiable and satisfy the required boundary conditions as a set of "basis functions."

The **Rayleigh-Ritz method** for approximating the BVP (24) is to solve the following finite-dimensional version (discretization) of it:

$$\text{Find } u \in \mathscr{A}(\mathscr{P}) \text{ satisfying } F(u) \leq F(v) \text{ for all } v \in \mathscr{A}(\mathscr{P}). \tag{1.21}$$

Note that the Rayleigh-Ritz problem (34) is obtained by the corresponding (MPE) problem (30) simply by replacing $\mathscr{A}$ by $\mathscr{A}(\mathscr{P})$. We will proceed now to discuss the special Rayleigh-Ritz method for our BVP (2) using the hat functions $\phi_i(x)$ of (33). Different basis functions and, more generally, different finite dimensional spaces give rise to different versions of the Rayleigh-Ritz method. Implementations using such hat functions are often referred to as the **(piecewise) linear Rayleigh-Ritz method**. Since, as in (32), any function in $\mathscr{A}(\mathscr{P})$ can be written as $\sum c_i\phi_i$, making use of bilinearity, we may write:

$$\begin{aligned} F\left(\sum c_i\phi_i\right) &= \frac{1}{2}\left\langle \left[\sum c_i\phi_i\right]', \left[\sum c_i\phi_i\right]'\right\rangle - \left\langle f, \sum c_i\phi_i\right\rangle \\ &= \frac{1}{2}\left\langle \sum c_i\phi_i', \sum c_j\phi_j'\right\rangle - \sum c_i\left\langle f,\phi_i\right\rangle \\ &= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} c_ic_j\left\langle \phi_i',\phi_j'\right\rangle - \sum c_i\left\langle f,\phi_i\right\rangle \end{aligned} \tag{1.22}$$

The above expression can be viewed as a (quadratic) function of the variable $(c_1, c_2, \cdots, c_n) \in \mathbb{R}^n$. We can locate its minimum by setting each of the partial derivatives equal to zero. Using the product rule, we can compute as follows:

$$\frac{\partial}{\partial c_k}F\left(\sum c_i\phi_i\right) = 0 \Rightarrow \frac{1}{2}\sum_{j=1}^{n} c_j\left\langle \phi_k',\phi_j'\right\rangle + \frac{1}{2}\sum_{i=1}^{n} c_i\left\langle \phi_i',\phi_k'\right\rangle = \left\langle f,\phi_k\right\rangle (1 \leq k \leq n).$$

Now using symmetry of the inner product, we can combine the two summations on the left into one:

$$\sum_{j=1}^{n}\left\langle \phi_k',\phi_j'\right\rangle c_j = \left\langle f,\phi_k\right\rangle (1 \leq k \leq n) \tag{1.23}$$

We abbreviate this linear system as

$$Ac = b, \tag{1.24}$$

where $A = [a_{ij}] = \left[\left\langle \phi_i',\phi_j'\right\rangle\right]$ is the so-called $(n \times n)$ **stiffness matrix**, and b is the so called $(n \times 1)$ **load vector**: $[b_j] = \left[\left\langle f,\phi_j\right\rangle\right]$. The terminology comes from the model of (24) for an elastic beam.

To compute the entries of the stiffness matrix: $\left\langle \phi_i',\phi_j'\right\rangle = \int_0^1 \phi_i'(x)\phi_j'(x)dx$, we first observe that, from the properties of the hat functions, $\phi_i'(x)\phi_j'(x) = 0$ unless i and j are equal or are adjacent indices. Thus, the stiffness matrix is both symmetric and tridiagonal. To compute the nonzero entries of A, there are just two cases. We use (33) for the computations:

$$\left\langle \phi_i',\phi_i'\right\rangle = \int_{x_{i-1}}^{x_{i+1}}\left[\phi_i'(x)\right]^2 dx = \int_{x_{i-1}}^{x_i}\left[1/h_{i-1}\right]^2 dx + \int_{x_i}^{x_{i+1}}\left[1/h_i\right]^2 dx = \frac{1}{h_{i-1}} + \frac{1}{h_i}, \tag{1.25}$$

$$\left\langle \phi_i',\phi_{i+1}'\right\rangle = \int_{x_i}^{x_{i+1}}\phi_i'(x)\phi_{i+1}'(x)dx = \int_{x_i}^{x_{i+1}}\left[\frac{-1}{h_i}\right]\left[\frac{1}{h_i}\right]dx = \frac{-1}{h_i}. \tag{1.26}$$

EXERCISE FOR THE READER 1.6.

(a) Show that the stiffness matrix A is positive definite (i.e., show that for any $n \times 1$ vector c, we have $c\prime Ac \geq 0$ with equality if and only if c is the zero vector). [8]

(b) Show that in case all grid spaces are equal, (i.e., $h_i = \|\|\mathscr{P}\|\|$ for all i), the stiffness matrix for linear system of the linear Rayleigh-Ritz (FEM) is a constant multiple of the coefficient matrix for the finite difference method introduced in Section 10.3. How do the linear systems compare?

As a general rule for Rayleigh-Ritz methods (and finite element methods for PDEs), it is usually a good idea to place more nodes where the (known) coefficient functions in the problem undergo the most activity. Adaptive methods can be developed in which successive refinements are used to see where to place additional nodes. We are ready to give a numerical example of the Rayleigh-Ritz method. In order to be able to get a check on errors, the following theorem will be useful:

THEOREM 1.3. *(Error Estimate for Rayleigh-Ritz Approximations)*

[8] Some general facts about positive definite matrices are that they are nonsingular and, if symmetric, their eigenvalues are all positive. The latter is, in fact, an equivalent definition (see, e.g., Section 8.4 of [HoKu-71] for proofs and more information on positive definite matrices). In particular, the stiffness matrix is nonsingular, so the Rayleigh-Ritz method leads to a unique solution.

Let $u_{\mathscr{P}}(x)$ denote the (piecewise) linear Rayleigh-Ritz approximation corresponding to a partition $\mathscr{P}$ of $[0,1]$ of the BVP (24) : $\begin{cases} -u''(x) = f(x), 0 < x < 1 \\ u(0) = 0, u(1) = 0 \end{cases}$, where $f(x)$ is a continuous function. The following error estimate holds for each $x, 0 \le x \le 1$:

$$|u_{\mathscr{P}}(x) - u(x)| \le \frac{\|\mathscr{P}\|^2}{2} \max_{0 \le x \le 1} |f(x)| \tag{1.27}$$

The proof of this theorem involves some nice ideas from analysis; an outline is left to Exercises 18-21 (see also the note preceding Exercise 17). In fact, in this setting it is even true that $u_{\mathscr{P}}(x_i) = u(x_i)$ at each grid point and thus $u_{\mathscr{P}}$ is really the piecewise linear interpolant of u with respect to the partition $\mathscr{T}$; see Exercise 21.

Example 1.4. Consider the (BVP) (24) $\begin{cases} -u''(x) = f(x), 0 < x < 1 \\ u(0) = 0, u(1) = 0 \end{cases}$ with [9]

$$f(x) = \sin\left[\operatorname{sign}(x-.5)\exp\left(\frac{1}{4|x-.5|^{1.05}+.3}\right)\right] \cdot \exp\left(\frac{1}{4|x-.5|^{1.2}+.2} - 100(x-.5)^2\right).$$

■

(a) Use the Rayleigh-Ritz method with $n = 50$ equally spaced interior grid values to solve this BVP and plot the resulting approximation.

(b) Solve the problem again with the Rayleigh-Ritz method and $n = 50$ interior grid values, but this time deploy a higher concentration of grid points where the inhomogeneity $f(x)$ is more oscillatory.

(c) Use Theorem 10.6 to find a (uniform) grid size that will guarantee that the Rayleigh-Ritz solution will be visually (without zooms) identical to the exact solution and compare both solutions of (a) and (b) with this more accurate solution.

SOLUTION: Since the BVP (24) is rather specialized, we will not bother writing here an M-file to perform the Rayleigh-Ritz method. Instead, we will go through each part directly, using MATLAB whenever convenient.

Part (a): Here we have $h_i = \|50\mathscr{P}\| = 1/51$ for each i, so that from the calculations above, the stiffness matrix is given by:

$$A = 51\begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & 0 & \\ & -1 & 2 & -1 & \\ & 0 & \ddots & \ddots & -1 \\ & & & -1 & 2 \end{bmatrix}$$

The entries of the load vector can be computed using MATLAB's integrator `quad`. The resulting system is then stored and solved using the Thomas algorithm. Note that in the case of equal grid spaces, the hat functions become symmetric and formula (33) for them can be abbreviated as (we set $h = \|\mathscr{P}\|$)

$$\phi_i(x) = \begin{cases} \frac{h-|x-x_i|}{h} = 1 - \frac{|x-x_i|}{h}, & \text{if } x_{i-1} \le x \le x_{i+1}, \\ 0, & \text{otherwise.} \end{cases}$$

(Verify this!) As the integrals required for the load vector entries are related, a loop will be used to compute them. The integrals will depend on a parameter. One way to compute such parameter-dependent integrals is to declare the parameter variables as global variables.

global var $\rightarrow$	Inside the definition of a function M-file having var as a variable, this command declares this variable to be a global variable. Recall that by default, all variables appearing in an M-file are local variables. Should also be used in the command window before invoking such an M-file.

The use of this strategy is demonstrated in the remainder of this example.

The coefficients of the load vector are given by: $(1 \le j \le 50)$

$$b_i = \langle f, \phi_i \rangle = \int_{x_{i-1}}^{x_{i+1}} f(x)\phi_i(x)dx = \int_{x_{i-1}}^{x_{i+1}} \left[1 - \frac{|x-x_i|}{h}\right] f(x)dx$$
$$= \int_{x_{i+1}} [1 - 51|x-x_i|] f(x)dx.$$

[9] We use the notation of the "sign function" (whose MATLAB counterpart has the same name): sign(x) $= 1$, if $x > 0, 0$, if $x = 0$, and -1, if $x < 0$.

The integrands depend on the parameter x_i, so we will first create an M-file for them using *xi*, which we declare as a global variable, to represent x_i [10].

```
function y = frayritzl0_7(x)
global xi;
y=(l-51*abs (x-xi)). *sin(sign(x-.5).*exp(l./(4*abs(x-.5).^1.05+.3))).*...
    .*exp(l./(4*abs(x-.5).^l.2+.2)-100*(x-.5).^2);
```

The load coefficients can now be created as follows: First declare our global variable and create the vector x of grid points. We remind the reader that vector indices must be positive integers so $x(1)$ represents x_0 and so on.

```
>> global xi;
>> for i=l:52
  x(i)=(i-l)/51;
end
```

With our M-file, the load coefficients are now easily created with the following loop. Notice that we have used `quadl` rather than `quad`. This former integrator works in the same syntax as `quad`, but uses a refined adaptive technique. It takes a bit more time to use but gives more accurate results.

```
>> for i=2:51
  xi=x(i);
  b(i)=quadl('frayritz10_7',x(i-1),x(i+1));
end
```

We have kept the indices consistent with those of the vector JC, but consequently we have created a vector with one extra component $b(1) = 0$. This component must be left out when we go on to solve the linear system. In order to solve the linear system, we will use the `thomas` M-file, which will solve our tridiagonal system quite efficiently. We must create the appropriate vectors to meet the syntax of this M-file:

```
>> d=2*ones(50,1)*51; %diagonal of stiffness matrix A
>> da=-1*ones(50,1)*51; da(51)=0; %superdiagonal (above)
>> db=-1*ones(50,1)*51; db(1)=0; %subdiagonal (below)
>> c=thomas(da,d,db,b(2:51));
```

As explained earlier, the values of the solution vector c are precisely the values of the numerical Rayleigh-Ritz solution at the interior grid points $x_1, x_2, \cdots, x_{50} = (x(24), x(25), \ldots, x(51))$. To plot the entire graph of c versus x, we need to augment the vector c to have first and last components which equal zero (from the boundary conditions). With this being done below, the resulting numerical plot is shown in Figure 10.12

```
>> c = [0 c 0];
>> plot(x,c,'b-o')
```

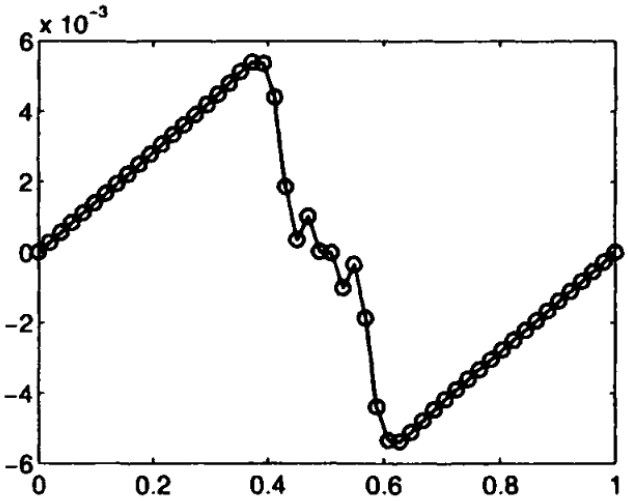


Figure 1.10: Rayleigh-Ritz solution of the BVP in Example 10.7 using 50 equally spaced interior grid points. The grid points/values are shown with (blue) circles.

Part (b): The right-hand side of the DE $-u''(x) = f(x)$ has the graph shown in Figure 10.13.

[10] A syntax note: If, after creating and storing this M-file we were to enter `frayritz10_7(24)`, the output would be "[]" (the empty vector), a reasonable answer since we have not yet defined `xi`. If we first entered a value for `xi`, say `xi=2` and reentered the above command, however, we would still get the empty vector as output. It is essential to first declare `xi` as a global variable in the command window (even though this was already done in the M-file). If this is done, and `xi=2` is reentered, then entering `frayritz10_7(24)` would finally produce an answer (ans =4.7321)).

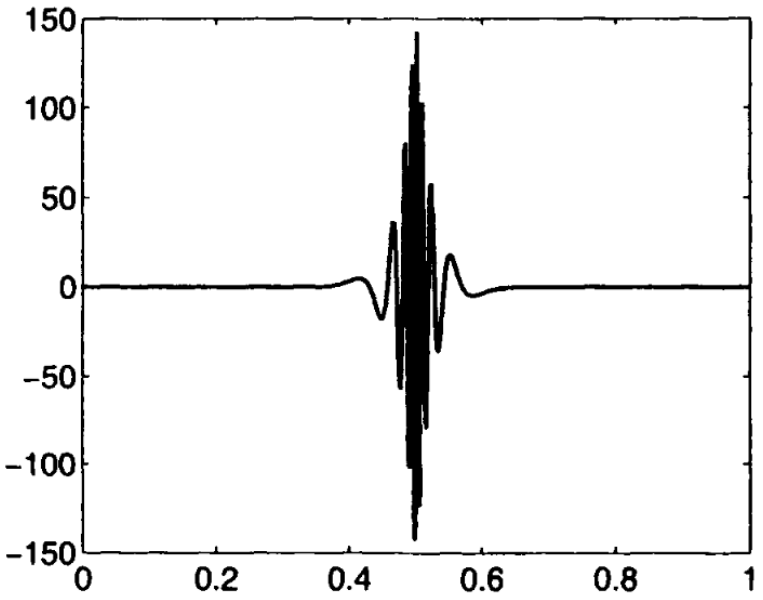


Figure 1.11: Graph of the right-hand side $f(x)$ of the DE $-u''(x) = f(x)$ of Example 10.7.

From Figure 10.13, we see that the inhomogeneity $f(x)$ is most oscillatory approximately on the interval $[0.35, 0.65]$ and elsewhere is rather tame. With this perspective, it would seem that any grid that is uniformly highly dense would give rise to much wasted computation on the long intervals of inactivity. Motivated by Figure 10.13, we propose the following deployment of the 50 interior grid points.

Put 6 in each of the intervals $[0, 0.35)$ and $(0.65, 1]$, and put the remaining 40 in $[0.35, 0.65]$. We stipulate that the grid points in each of these intervals be uniformly spaced but this is by no means necessary (the Rayleigh-Ritz method is totally flexible). The linspace command will make the construction of these grid values particularly straightforward:

```
>> x2(1:7)=linspace(0,0.35,7);
>> x2 (7:46)=linspace(0.35,0.65,40);
>> x2(46:52)=linspace(0.65,1,7);
```

Since the grid is no longer uniform, we need to construct a vector for the h_i:

```
>> for i=1:51, h(i)=x2(i+1)-x2(i); end
```

It is left to construct the load vector b. By (33) the coefficients are

$$b_i = \langle f, \phi_j \rangle = \int_{x_{i-1}}^{x_{i+1}} f(x)\phi_i(x)dx = \int_{x_{i-1}}^{x_i} \frac{x - x_{i-1}}{h_{i-1}} f(x)dx + \int_{x_i}^{x_{i+1}} \frac{x_{i+1} - x}{h_i} f(x)dx$$

Employing the strategy used in part (a), we need here a pair of M-files for the two respective integrands:

```
function y = frayritzl0_7a(x)
global xim;
global him;
y=(x-xim)./him.*sin(sign(x-.5).*exp(1./(4*abs(x-... .5).^1.05+.3))).*exp(
   l./(4*abs(x-.5).^1.2+.2)-100*(x-.5).^2);
function y = frayritzl0_7b(x)
global xip;
global hi;
y=(xip-x),/hi.*sin(sign(x-.5).*exp(l./(4*abs(x-... .5) .^1.05+.3))).*exp(
   l./(4*abs(x-.5).^l.2+.2)-100*(x-.5).^2);
```

The load vector is now easily constructed, and the linear tridiagonal system can be assembled and solved as before:

```
>> global xim him xip hi;
>> for i=2:51;
  xip=x2(i+l); xim=x2(i-l); hi=h(i); him=h(i-l);
  b2(i)=quadl('frayritzl8_la', x(i-l), x(i))+... quadl('frayritzl8_lb f ,
    x(i), x(i+l));
end
>> for i=1:51, h(i)=x(i+1)-x(i); end
>> for i=2:51, d2(i)=l/h(i-1)+l/h(i); end
>> %main diagonal will be d(Â£:51).
>> for i=2:50, da2(i)=-l/h(i); end
>> da2(51)=0; %superdiagonal will be da(2:51).
>> for i=2:50, db2(i)=-l h(i-1); end
```

```
>> %subdiagonal will be db(1:50)
>> c2=thomas(da2(2:51),d2(2:51),db2(1:50),b2(2:51));
```

The commands needed to plot this solution are just as in part (a), and those commands produce the plot shown in Figure 10.14(a). Part (c): From Figure 10.12, we see that the amplitude of the solution is roughly 6e-3. Theorem 10.6 gives maximum bound for the error to be $\frac{\|\mathscr{P}\|^2}{2}\max_{0\le x\le 1}|f(x)|$. Setting this expression to be $6e-3/100$ (so the maximum error will be less than about $1/100$ of the amplitude), using 150 for $\max_{0\le x\le 1}|f(x)|$ (from Figure 10.13), and solving for $\|\mathscr{P}\|$ gives roughly $1e-4$, so that if we use 10,000 interior grid points, the Rayleigh-Ritz solution should have the desired accuracy. The construction and plotting of this solution is done just as in part (a), except that instances of 50 or 51 , etc. should be changed to 10,000 or 10,001 , etc. The resulting graph is compared with the two obtained in parts (a) and (b) in Figure 10.14.

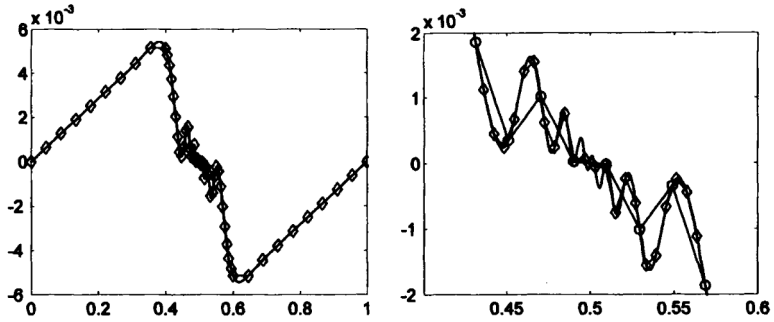


Figure 1.12: (a) (left) Rayleigh-Ritz solution obtained for Example 10.7(b) shown with diamonds, along with the exact solution in black. The grid used is nonuniform with more grid points (diamonds) deployed in the areas where the inhomogeneity is most active, (b) (right) Zoomed-in comparison of the Rayleigh-Ritz solutions in part (a) (circles) and part (b) (diamonds) with the exact solution (smooth curve) of Example 10.7. Note the surprising fact that the Rayleigh-Ritz solutions are exactly equal to the solution at the respective grid points, and hence the Rayleigh-Ritz solutions turn out simply to be the piecewise linear interpolants of the actual solution with respect to the associated grids (see Exercise 21 for a proof). This theorem will no longer hold in higher dimensions or even for more complicated single-variable BVPs.

We now turn to the **Galerkin method** [11] for approximating the solution of the BVP (24). In the piecewise linear setting with the finite-dimensional space $\mathscr{A}(\mathscr{P})$ in place of the space $\mathscr{A}$ of all admissible functions, this method solves the discrete analogue of the Principle of Virtual Work (31):

$$\text{Find} u\in\mathscr{A}(\mathscr{P})\text{satisfying}\left\langle u',v'\right\rangle=\langle f,v\rangle\text{for all}v\in\mathscr{A}(\mathscr{P}). \tag{1.28}$$

In light of the bilinearity of the inner product, it is enough to check (41) for the function v running through the n (basis) functions: $\{\phi_k\}_{k=1}^n$. The discrete problem is thus to determine the coefficients $(c_1,c_2,\cdots,c_n)\in\mathbb{R}^n$ of the function $u=\sum_{j=1}^n c_j\phi_j\in\mathscr{A}(\mathscr{P})$ such that $\left\langle\left(\sum c_i\phi_i\right)',\phi_k'\right\rangle=\langle f,\phi_k\rangle\,(1\le k\le n)$.

Using bilinearity, these equations become $\sum_{i=1}^n c_j\left\langle\phi_j',\phi_k'\right\rangle=\langle f,\phi_k\rangle$, which is precisely (36). Thus, for the (BVP) (24), the Rayleigh-Ritz and Galerkin methods coincide, and this is true for any choice of basis functions (not necessarily the piecewise linear basis functions). The next exercise for the reader will use a set of basis functions that does not depend on any particular partition, but rather comes from the so-called eigenfunctions of the BVP.[12]

EXERCISE FOR THE READER 1.7.

Apply the Galerkin method to re-solve the BVP of Example 10.7 using the following 50 basis functions: $\varphi_k(x)=\sin(k\pi x),k=1,2,\cdots,50$. Compare the accuracy with that obtained in part (a) of Example 10.7.

[11] Like the works of Rayleigh and Ritz, the work of Russian engineer/mathematician Boris Grigorievich Galerkin (1871-1945) was motivated by physical problems. It is fair to characterize Galerkin as an applied mathematician of the purest sense. He worked many years as an engineer before his first publication at the relatively late age of 38 on longitudinal curvature. The paper was a significant extension of work of Euler and it was applied in the construction of bridges and building frames. His continued interest in structural mechanics led him to the discovery in 1915 of his most notable contribution to mathematics, what is known today as the Galerkin method. He subsequently took on some academic posts in St. Petersburg, which was the de facto mathematical capital of Russia at the time. His interests in consulting with industry and in the relevant mathematics continued until his death. In 1937 he published a pivotal treatise on thin elastic plates.

[12] For the BVP $-u''=f(x),u(0)=u(1)=0$, the associated **eigenfunctions** are nontrivial solutions of the BVP $-u''=\lambda u,u(0)=u(1)=0$ for some $\lambda>0$. It can be shown that the totality of these eigenfunctions is as follows: $u_k(x)=\sin(k\pi x),k=1,2,\cdots$ (see Exercise 24). The eigenfunctions are pairwise orthogonal: $\langle u_k,u_\ell\rangle=\delta_{k\ell}/2$ (where $\delta_{k\ell}$ denotes the Kronecker delta equaling 1 if the indices are equal, otherwise equaling 0), as are their derivatives (Exercise 24). Moreover, the eigenfunctions have the remarkable property that any function u satisfying the same boundary conditions and satisfying reasonable regularity assumptions (say if $u\in\mathscr{A}$) can be expressed as an infinite series of these eigenfunctions: $u(x)=\sum_{k=1}^\infty c_k u_k(x)$. In particular, solutions of such inhomogeneous BVPs have such eigenfunction expansions. Such eigenfunction expansion theory of ODE BVPs falls under the name of Sturm-Liouville theory. The analog for PDE BVPs is the theory of Fourier series. Both of these analytical techniques are covered extensively in many theoretically or analytically oriented textbooks. For references we cite [Str-92] and [Sni-99]. All of these properties make finite subsets of these eigenfunctions seem like very reasonable candidates for Rayleigh-Ritz and Galerkin methods; these types of Rayleigh-Ritz methods are often referred to as spectral methods.

For general BVPs, the Rayleigh-Ritz and Galerkin methods often, but not always, coincide. For this reason the nomenclature sometimes refers to the "RayleighRitz-Galerkin method." Both methods have been developed for a great many BVPs. The formulation of the Rayleigh-Ritz method in general is a bit more involved since it entails the determination of the appropriate functional for the analogue of Theorem 10.5 to be valid. Such problems usually fall under the classical area of the *calculus of variations*. We now present a brief outline for the Rayleigh-Ritz method for solving the following more general BVP whose DE will be a prototype for the elliptic PDE problems we shall contemplate Chapter 13. All of what follows in this outline can backed up theoretically with techniques similar to those used earlier to deal with the simpler problem (24); some of the details will be left to the exercises.

$$\text{(BVP)} \begin{cases} -(p(x)u'(x))' + q(x)u(x) = f(x), \quad 0 < x < 1 \\ u(0) = 0, u(1) = 0 \end{cases} \tag{1.29}$$

Different boundary conditions and more general equations can be dealt with using modified functionals. The next exercise for the reader, however, shows how BVPs with general Dirichlet BC's can be reduced to (42) using a simple change of variables. The exercises will examine some further modifications.

EXERCISE FOR THE READER 1.8.

(a) Show that the following BVP,

$$\text{(BVP)} \begin{cases} -(p(x)w'(x))' + q(x)w(x) = f(x), \quad 0 < x < 1 \\ w(0) = \alpha, w(1) = \beta \end{cases} \tag{1.30a}$$

can be reduced to the form (42) by making the following change of variables/ function:

$$u(x) = w(x) - (1-x)\alpha - \beta x$$

(b) Show that the following BVP,

$$\text{(BVP)} \begin{cases} -(p(t)w'(t))' + q(t)w(t) = f(t), a < t < b \\ w(a) = \alpha, w(b) = \beta \end{cases} \tag{1.30b}$$

can be reduced to $(42a)$ by making the following change of variables/function:

$$x = (t-a)/(b-a).$$

The analogue for Theorem 10.5 (for the Rayleigh-Ritz formulation) is the following theorem whose complete proof can be found in Section 7.2 in [Sch-73].

THEOREM 1.4. *(Rayleigh-Ritz Principle for a One-Dimensional BVP*

In the BVP (42):

$$\begin{cases} -(p(x)u'(x))' + q(x)u(x) = f(x), 0 < x < 1 \\ u(0) = 0, u(1) = 0 \end{cases}$$

suppose that the (known) functions $p(x), q(x)$ and $f(x)$ are continuous and additionally that $p(x)$ is differentiable on the open interval $I = [0,1]$ [13]. Also assume that $p(x) > 0$ and $q(x) \geq 0$ throughout I. Under these assumptions, the BVP has a unique solution which coincides with the unique minimizer of the functional

$$F(v) = \int_0^1 \left[p(x) \left(v'(x)\right)^2 + q(x)(v(x))^2 - 2f(x)v(x) \right] dx, \tag{1.31}$$

over the set of admissible functions

$\mathscr{A} = \{v : [0,1] \to \mathbb{R} : v(x)$ is continuous, $v'(x)$ is piecewise continuous and bounded, and $v(0) = 0, v(1) = 0\}$. We remind the reader that without these hypotheses, the BVP (42), in general, may have no solutionâĂŤsee Exercise 12 of Section 10.2 or Exercise 24 of this section.

If we use spaces $\mathscr{A}(\mathscr{P})$ of admissible functions spanned by the hat-functions determined by a partition $\mathscr{P}$ of $[0,1]$, the Rayleigh-Ritz method seeks to minimize the functional F evaluated at a typical element $v(x) = \sum_{i=1}^{n} c_i \phi_i(x)$ of $\mathscr{A}(\mathscr{D})$ (see (32)). A similar computation to what was given above (Exercise 12) will show that if we substitute this function into (43) and set each of the partial derivatives (with respect to the parameters $c_i (1 \leq i \leq n)$) equal to zero, we obtain the $n \times n$ linear system

$$Ac = b,$$

[13] Actually, the theorem still works under weaker conditions stated in [Sch-73]. The most natural setting for the Rayleigh-Ritz method is in the context of Sobolev functions. This topic is rather advanced, however, so we fix our ideas on the classical formulation. The interested reader may also consult the references [StFi-73] and [AxBa-84] for more sophisticated treatments on the subject.

where the $n \times n$ **stiffness matrix** $A = [a_{ij}]$ has coefficients given by:

$$a_{ij} = \int_0^1 \left[p(x)\phi_i'(x)\phi_j'(x) + q(x)\phi_i(x)\phi_j(x) \right] dx \tag{1.32}$$

and the $n \times 1$ **load vector** b has entries given by:

$$b_j = \int_0^1 \left[f(x)\phi_j(x) \right] dx \tag{1.33}$$

As before, the stiffness matrix is clearly a tridiagonal symmetric matrix that can be shown to be positive definite. Thus there will be a unique solution of the linear system and so the method will always produce Rayleigh-Ritz approximations. There is also an error estimate analogous to Theorem 10.6 which states roughly that $\left| u_g(x) - u(x) \right| \leq C \left\| \mathscr{P} \right\|^2 \max_{0 \leq x \leq 1} |f(x)|$. Thus, we get the same type of error estimate (proportional to $\left\| \mathscr{P} \right\|^2$) as we had in the simpler introductory BVP. The proportionality constant C will of course depend on the data $p(x)$ and $q(x)$, but not on $u(x)$ or $f(x)$ (see [StFi-73] for details). The tridiagonal coefficients of the stiffness matrix in (44) can be simplified, using (33), as was done previously, to obtain (cf. with (38), (39)):

$$\begin{aligned} a_{ii} = & \frac{1}{h_{i-1}^2} \int_{x_{i-1}}^{x_i} p(x)dx + \frac{1}{h_i^2} \int_{x_i}^{x_{i+1}} p(x)dx \\ & + \frac{1}{h_{i-1}^2} \int_{x_{i-1}}^{x_i} (x - x_{i-1})^2 q(x)dx + \frac{1}{h_i^2} \int_{x_i}^{x_{i+1}} (x_{i+1} - x)^2 q(x)dx, \text{ for } 1 \leq i \leq n \end{aligned} \tag{1.34}$$

$$a_{i,i+1} = \frac{-1}{h_i^2} \int_{x_i}^{x_{i+1}} p(x)dx + \frac{1}{h_i^2} \int_{x_i}^{x_{i+1}} (x_{i+1} - x)(x - x_i) q(x)dx, \quad \text{for } 1 \leq i < n \tag{1.35}$$

Evaluation of the integrals in (44) through (47) can be a time-consuming process in cases of fine partitions. In such cases where the coefficient functions p, q, and f are not too wildly behaved, it is a good idea to replace each of these functions by their piecewise linear approximation (piecewise linear splines) in the integrals. By Exercise 13(b), the local errors of such approximations are $O\left(h_i^2\right)$ on each of the corresponding intervals, provided that the function is $\mathscr{C}^2$, and this in turn implies $O\left(h_i^3\right)$ estimates for each of the integrals. We do one such approximation for the fourth integral in (46); the rest are done in a similar fashion and are left as Exercise 13(a). The piecewise linear approximation $Q(x)$ to $q(x)$ relative to the partition $\mathscr{P}$ of $[0,1]$ can be expressed quite simply using the hat functions as follows:

$$Q(x) = q(x_i)\,\phi_i(x) + q(x_{i+1})\,\phi_{i+1}(x), \quad x \in [x_i, x_{i+1}].$$

Replacing this approximation for $q(x)$ in the last integral of (46) leads us to the following estimate:

$$\begin{aligned} & \frac{1}{h_i^2} \int_{x_i}^{x_{i+1}} (x_{i+1} - x)^2 q(x)dx \\ & \qquad \approx \frac{1}{h_i^2} \int_{x_i}^{x_{i+1}} (x_{i+1} - x)^2 \left[q(x_i)\,\phi_i(x) + q(x_{i+1})\,\phi_{i+1}(x) \right] dx \\ & \qquad = \frac{1}{h_i^2} \int_{x_i}^{x_{i+1}} (x_{i+1} - x)^2 \left[q(x_i)\,\frac{x_{i+1} - x}{h_i} + q(x_{i+1})\,\frac{x - x_i}{h_i} \right] dx \\ & \qquad = \frac{q(x_i)^{x_{i+1}}}{h_i^3} \int_{x_i} (x_{i+1} - x)^3 dx + \frac{q(x_{i+1})}{h_i^3} \int_{x_i}^{x_{i+1}} (x_{i+1} - x)^2 (x - x_i) dx. \end{aligned}$$

The two latter integrals are easily evaluated explicitly, for example:

$$\begin{aligned} \int_{x_i}^{x_{i+1}} (x_{i+1} - x)^2 (x - x_i) dx & \underset{u = x_{i+1} - x}{\overset{\text{Subst.}}{=}} \int_{h_i}^0 u^2 (h_i - u)(-du) = \int_{h_i}^0 \left[u^3 - h_i u^2 \right] du \\ & = \left[\frac{u^4}{4} - h_i \frac{u^3}{3} \right]_{h_i}^0 = \frac{h_i^4}{12}. \end{aligned}$$

In a similar fashion we find that $\int_{x_i}^{x_{i+1}} (x_{i+1} - x)^3 dx = \frac{h_i^4}{4}$. Putting these into the above estimate gives us that:

$$\frac{1}{h_i^2} \int_{x_i}^{x_{i+1}} (x_{i+1} - x)^2 q(x)dx \approx \frac{h_i}{12} \left[3q(x_i) + q(x_{i+1}) \right].$$

Similar treatments for the remaining integrals appearing in (46) through (47) (see Exercise 13(a)) result in the following estimates:

$$\begin{aligned} a_{ii} \approx & \frac{1}{2h_{i-1}} \left[p(x_{i-1}) + p(x_i) \right] + \frac{1}{2h_i} \left[p(x_i) + p(x_{i+1}) \right] \\ & + \frac{h_i}{12} \left[q(x_{i-1}) + 3q(x_i) \right] + \frac{h_i}{12} \left[3q(x_i) + q(x_{i+1}) \right] \end{aligned} \tag{1.36}$$

for $1 \leq i \leq n$, and

$$a_{i,i+1} \approx -\frac{1}{2h_i}\left[p\left(x_i\right)+p\left(x_{i+1}\right)\right]+\frac{h_i}{12}\left[q\left(x_i\right)+q\left(x_{i+1}\right)\right] \tag{1.37}$$

for $1 \leq i < n$. In the same fashion, the load vector coefficients are estimated as follows:

$$b_j \approx \frac{h_{j-1}}{6}\left[f\left(x_{j-1}\right)+2f\left(x_j\right)\right]+\frac{h_j}{6}\left[2f\left(x_j\right)+f\left(x_{j+1}\right)\right], \tag{1.38}$$

for $1 \leq j \leq n$.

Our next example will compare performance speed and accuracy of both of the above implementations of the Rayleigh-Ritz method for a specific BVP. It is possible to solve this BVP explicitly, so we will be able to make accurate estimates for the error. The explicit solution, however, is quite a mess. It can be derived using standard methods in differential equations (undetermined coefficients). To avoid having to even write it down, we use MATLAB's Symbolic Toolbox to compute the explicit solution but suppress its output.

Example 1.5. Consider the following problem:

■

$$\text{(BVP)}\begin{cases} -u''(x)+6u(x)=e^{10x}\cos(12x) & 0<x<1 \\ u(0)=0, u(1)=0 \end{cases}$$

(a) Use the Rayleigh-Ritz method with $n = 500$ equally spaced interior grid values to solve this BVP and plot the resulting approximation. Keep a record of the computing time it takes to determine the load vector and stiffness matrix coefficients.

(b) Repeat part (a), this time invoking the approximations (48) thru (50) for the integrals appearing in the Rayleigh-Ritz method.

(c) Compare both solutions of (a) and (b) with the exact solution as obtained using MATLAB's Symbolic Toolbox.

SOLUTION: The BVP given indeed fits the template of (42) with $p(x) = 1$, $q(x) = 6$, and $f(x) = e^{10x}\cos(12x)$.

Part (a): Here we have $h_i = \|\mathscr{P}\| = 1/501$ for each i, so we must compute the tridiagonal entries of the 501×501 stiffness matrix along with the 501 load coefficients using (45)-(47). The computations are done in a similar fashion to how the load vector coefficients were done in Example 10.7. Of the $1+4+2=7$ integrals appearing in (45) through (47), $1+2+1=4$ of them will need the "global variable" strategy in conjunction with MATLAB's numerical integrator `quadl`. The remaining three integrals have constant integrand $(p(x)=1)$ and so will be done directly. For (45) we use the fact that since the spacing is uniform, we have $\phi_i(x) = 1-|x-x_i|/\|\mathscr{P}\| = 1-501\,|x-x_i|$ for $x_{i-1} \leq x \leq x_{i+1}$. Actually, because $p(x)$ and $q(x)$ are constant functions for this problem, the approximations (48) and (49) are indeed exact. Nonetheless, we will proceed to use the quadl integrator for these integrals so as to give a good impression of the extra expense in bringing in a more sophisticated tool. For the global variables x_{i-1}, x_i, x_{i+1} we will use the MATLAB notation: `xim, xi, xip` (p for plus, *m* for minus). The four needed M-files are as follows.

```
function y = frayritzl0_81oad(x)
global xi;
y=(l-501*abs(xi-x)).*exp(10*x).*cos(12*x);
```

```
function y = frayritzl0_8stiff1(x)
global xim;
y=(x-xim).^2*6;
```

```
function y = frayritzl0_8stiff2(x)
global xip;
y=(x-xip).^2*6;
```

```
function y = frayritzl0_8stiff3(x)
global xi xip;
y=(xip-x).*(x-xi)*6
```

Note that all of the intervals of integration have length $h = \|\mathscr{P}\| = 1/501$, so that each of the integrals in (46) and (47) with integrand p equals (since $p(x) = 1$) $\|\mathscr{P}\| = 1/501$. With these M-files stored, the following loop will use (45) thru (47) to construct the needed coefficients:

```
>> x=linspace(0,1,502); h=l/501; global xi xim xip;
>> tic, for i=2:501
xi=x(i); xim=x(i-l); xip=x(i+l);
b(i)=quadl('frayritzl0_81oad',xim,xip);
d(i)=2/h+l/h^2*quadl('frayritzlO_8stiffl,/xim/xi)...
+l/h^2*quadl('frayritzl0_8stiff2', xi, xip)
```

```
%d(2:501) is diagonal of stiffness matrix
da(i)=-1/h+1/h"2*quadl('frayritzl0_8stiff3',xi,xip);
%da(2:501) is superdiagonal of stiffness matrix (above),
%once we set da(501)=0 (after loop).
end, toc
```

→ elapsed_time = 4.0360(seconds)

```
>> db(3:501)=da(2:500);
>> db(2)=0; da(501)=0;
>> %db is subdiagonal of stiffness matrix (below)
```

As usual, we needed to properly format the sub/superdiagonals for input into the Thomas algorithm, which we apply next.

```
>> cl=thomas(da(2:501),d(2:501),db(2:501),b(2:501));
cl=[0 cl 0] ;
plot(x,cl)
```

The resulting plot, which as we will see turns out to be visually indistinguishable from that of the exact solution, is shown in Figure 10.15.

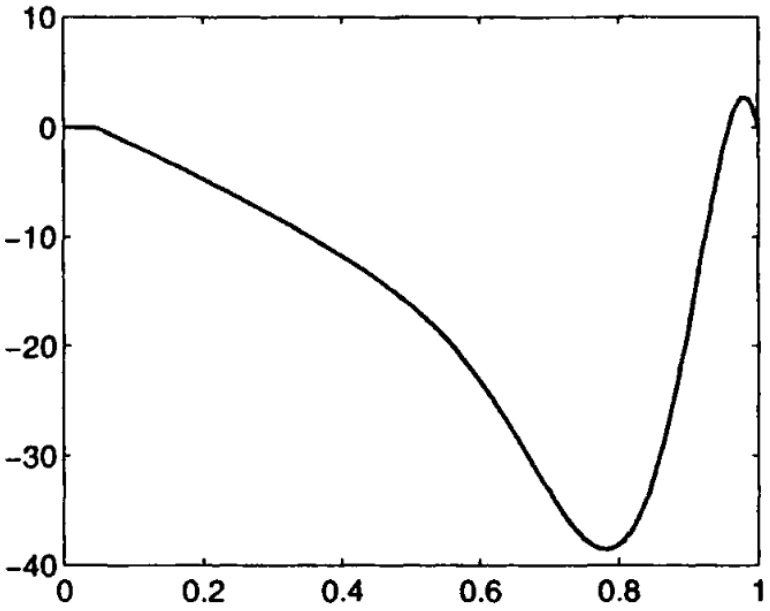


Figure 1.13: Plot of the solution of the BVP of Example 10.8.

Part (b): Using the estimates (48) through (50), it will be quite a simple (and quick) task to collect the needed coefficients. This can be accomplished with the following loop:

```
>> p=ones(502,1); q=6*p; x=linspace(0,1,502); f=exp(10*x).*cos(12*x);
>> h=1/501; %uniform step size
```

```
>> tic, for i=2:501
d(i)=1/(2*h)*(p(i-1)+2*p(i)+p(i+1))+h/12*(q(i-1)+6*q(i)+q(i+1) ;
%d(2:501) is diagonal of stiffness matrix
da(i)=-1/(2*h)*(p(i)+p(i+1))+h/12*(q(i)+q(i+1));
da(501)=0;
%da(2:501) is superdiagonal od stiffnes matrix (above)
db(i)=-1/(2*h)*(p(i-1)+p(i))+h/12*(q(i-1)+q(i));
db(2)=0;
%db(2:501) is subdiagonal of stiffness matrix (below)
b(i)=h/6*(f(i-1)+4*f(i)+f(i+1)) ;
% b(2:501) is load vector
end, toc
```

→ elapsed_time = 0.0500(seconds)

```
>> c2=thomas(da(2:501),d(2:501),db(2:501),b(2:501));
>> c2=[0 c2 0] ;
>> plot(x,c2)
```

The resulting plot is visually indistinguishable from the one in part (a), shown in Figure 10.15.

Part (c): The BVP is rather special in that an explicit solution can be written down. Labeling the symbolic solution as `yexact` and suppressing the long output, we can create it in a MATLAB session (provided the symbolic toolbox or student edition is being used) with the following command.

```
>> yexact=dsolve('-D2y+6*y=exp(10*t)*cos(12*t)', 'y(0)=0', 'y(1)=0');
```

We next create two vectors for the appropriate time values and corresponding values of the exact solution. We will need to use the `double` command along with the subs commands introduced earlier to convert the symbolic numbers to floating point format. The data is then plotted and the result is shown in Figure 10.15.

```
>> t=linspace(0,1,502);
>> Yexact=subs(yexact,t);
>> plot(t,Yexact)
```

With the variables from parts (a) and (b) still remaining in our workspace, we can easily obtain plots of the errors of the numerical solutions in those parts with the following commands. The two plots are shown in Figure 10.16.

```
>> plot(x,abs(c1-Yexact))
>> plot(x,abs(c2-Yexact))
```

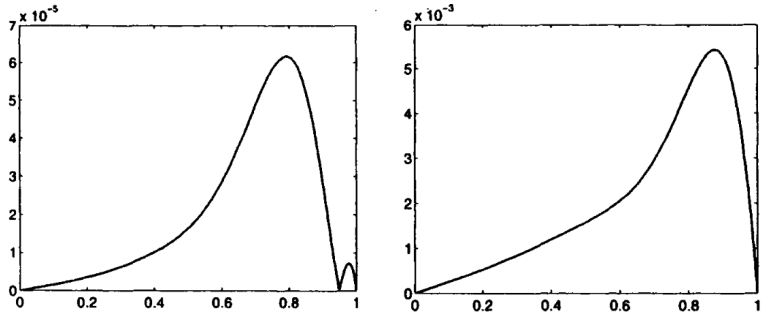


Figure 1.14: Plots for the errors of the two numerical solutions obtained in parts (a) (left) and (b) (right) of Example 10.8.

EXERCISE FOR THE READER 1.9.

(a) Write an M-file called `rayritz` having the following input/output variables: `[x,u]=rayritz(p,q,f,n)`. The program will implement the piecewise linear Rayleigh-Ritz method with (48) through (50) to solve the BVP (42) :

$$\begin{cases} -(p(x)u'(x))' + q(x)u(x) = f(x), 0 < x < 1 \\ u(0) = 0, u(1) = 0 \end{cases}$$

The first three inputs `p, q`, and `f` can represent the coefficient functions in the DE of (42), and the last input variable `n` denotes the number of interior grid points to use. A uniform grid is assumed. The output variables will be the domain and range vectors for the numerical solution.

(b) Starting with $n = 99$ interior grid points ($h = 1/100$), use this program to get a numerical solution $y1$ of the BVP in Example 10.8, then use 199 grid points ($h = 1/200$), getting a corresponding solution $y2$, and find the maximum absolute difference of the computed solutions on the common domain values. Now cut the gap in half again with $n = 399$, and get a corresponding solution $y3$ and look at the maximum absolute difference with the vector $y2$ at common domain points. Continue this process until the maximum absolute difference is less than 5×10^{-5}. Now (if you have access to the Symbolic Toolbox) compute the actual maximum error of this final solution compared to the exact solution in Example 10.8.

The Rayleigh-Ritz approximations we have obtained were all piecewise continuous but not differentiable. The versatility of the Rayleigh-Ritz method allows us, in fact, to use any sets of linearly independent functions as basis functions. The catch is that the resulting stiffness matrix should be reasonably well conditioned. Some different sets of basis functions will be examined in the exercises; see Exercise 6 for a problematic situation. The hat basis functions we used resulted in numerical approximations that were piecewise continuous but not differentiable. This lack of differentiability can be overcome by the use of more elaborate basis functions. One popular scheme is to use cubic splines for the basis functions; a typical one is shown in Figure 10.17.

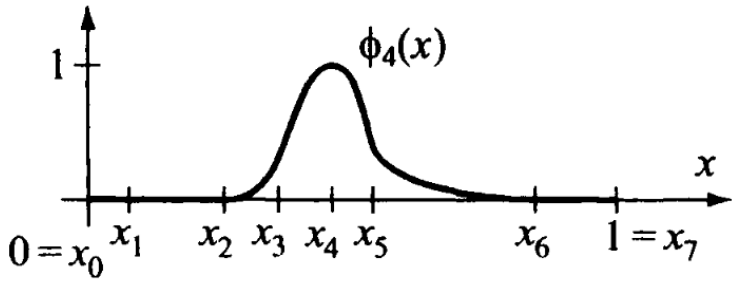


Figure 1.15: A cubic spline basis function. Unlike the piecewise linear hat functions of Figure 10.11, such basis functions are typically nonzero at three node points.

Each cubic spline basis function will have two continuous derivatives and thus so will the numerical approximations (since they are linear combinations of basis functions). The price we will need to pay for this extra smoothness in the numerical solutions is that the resulting stiffness matrix will typically have seven nonzero diagonals, rather than three, and the coefficients will be more complicated to compute. We proceed to outline an implementation of such cubic spline basis functions in the Rayleigh-Ritz method.

We restrict to the case of uniform grids and begin by defining the basic cubic spline function from which all other spline basis functions can be defined. This basic spline, which we denote by $BS(x)$, will be defined using the five nodes: $x_0 = -2, x_1 = -1, x_2 = 0, x_3 = 1$, and $x_4 = 2$ by the following requirements:

1. On each interval $[x_i, x_{i+1}]\,(i = 0,1,2,3), BS(x)$ will be a polynomial of degree at most three.
2. $BS(x), BS'(x)$, and $BS''(x)$ are each continuous at the node interfaces $x = x_1, x_2, x_3$.
3. $BS(\pm 2) = 0, BS(0) = 1$ (interpolation requirements).
4. $BS'(x)$, and $BS''(x)$ both equal zero at the endpoint nodes $x = x_0, x_4$.

EXERCISE FOR THE READER 1.10.

(a) Show that the conditions (i) through (iv) above uniquely determine the function $BS(x)$ to be an even function $(BS(-x) = BS(x))$ in $\mathscr{C}^2(\mathbb{R})$ and specified by the following formula:

$$BS(x) = \begin{cases} \left[(2-x)^3 - 4(1-x)^3\right]/4, & \text{if } x \in [0,1], \\ (2-x)^3/4, & \text{if } x \in (1,2], \\ 0, & \text{if } x > 2, \\ BS(-x), & \text{if } x < 0. \end{cases} \tag{1.39}$$

(b) Get MATLAB to plot this function. Using the basic spline function $BS(x)$, we can define our basis $\{\phi_i(x)\}_{i=1}^n$ functions for the BVP (42) on $[0,1]$ corresponding to a uniform grid $0 = x_0 < x_1 < \cdots < x_n < x_{n+1} = 1$ with mesh size $h = x_{i+1} - x_i = 1/(n+1)$. These functions are specified by the formulas below:

$$\phi_i(x) = \begin{cases} BS\left(\frac{x-h}{h}\right) - BS\left(\frac{x+h}{h}\right), & \text{if } i = 1 \\ BS\left(\frac{x-ih}{h}\right), & \text{if } i = 2,3,\cdots,n-1 \\ BS\left(\frac{x-nh}{h}\right) - BS\left(\frac{x-(n+2)h}{h}\right), & \text{if } i = n. \end{cases} \tag{1.40}$$

EXERCISE FOR THE READER 1.11.

(a) Show that the basis functions $\{\phi_i(x)\}_{i=1}^n$, as specified in (52), form a linearly independent set of functions. Also, show that on each interval $(x_i, x_{i+1}), \phi_i$ is a polynomial of degree at most three and that $\phi_i, \phi_i', \phi_i''$ are continuous at the endpoints x_i, x_{i+1}. Show that $\phi_i(x_i) = 1$, $\phi_i(x_j) = 0$ if $|i-j| \geq 2$ or $j = 0$ if $i = 1$, or $j = n+1$ if $i = n$, and $\phi_i(x) = 0$ if there is such an x_j that lies between x_i and x.
(b) Using the value $n = 5$, get MATLAB to plot each of the five corresponding hat functions.
In order to implement these basis functions in the method, we will need to compute their derivatives. By the chain rule, these can be easily computed in terms of BS'(x), which by simple computation is as specified below:

$$BS'(x) = \begin{cases} \frac{3}{4}\left[4(1-x)^2 - (2-x)^2\right], & \text{if } x \in [0,1], \\ -\frac{3}{4}(2-x)^2, & \text{if } x \in (1,2], \\ 0, & \text{if } x > 2, \\ -BS'(-x), & \text{if } x < 0. \end{cases} \tag{1.41}$$

Each of the "BS(• " expressions in (52) will have, by the chain rule, derivative equal to $BS'(\bullet)/h$. Note also that since $\phi_i(x)$ and $\phi_i'(x)$ equal zero outside the interval $[x_{i-2}, x_{i+2}]$, it follows that the stiffness matrix entries $a_{ij} = \int_0^1 \left[p(x)\phi_i'(x)\phi_j'(x) + q(x)\phi_i(x)\phi_j(x)\right] dx$ (from (44)) will be zero if $|i-j| > 3$, and from this it follows that the stiffness matrix will be a banded matrix with (at most) seven bands. With these observations, it is a simple matter to incorporate the cubic spline Rayleigh-Ritz method into a MATLAB program. Examples will be left to the exercises. We close this section with a result on errors of the cubic spline Rayleigh-Ritz method, which shows it is often worth the extra work required over the basic piecewise linear scheme.

THEOREM 1.5. *(Errors in Cubic Spline Rayleigh-Ritz Approximations)*

Suppose that the exact solution of the BVP (42)

$$\begin{cases} -(p(x)u'(x))' + q(x)u(x) = f(x), 0 < x < 1 \\ u(0) = 0, u(1) = 0 \end{cases}$$

is $\mathscr{C}^4([0,1])$ and the data $p(x), q(x)$, and $f(x)$ satisfy the assumptions of Theorem 10.7. If $u_{\mathscr{P}}$ is the cubic spline Rayleigh-Ritz approximation for this problem corresponding to a partition $\mathscr{P}$ of $[0,1]$, then we have the following error estimate:

$$|u_{\mathscr{P}}(x) - u(x)| \leq C\|\mathscr{P}\|^3 \max_{0 \leq x \leq 1} \left|u^{(4)}(x)\right| \text{ for each } x \text{ in } [0,1] \tag{1.42}$$

For a proof of this theorem we refer to Section 7.5 of [StBu-92]. The key point is that the error estimate is proportional to $\|\mathscr{P}\|^3$, which is superior to the $\|\mathscr{P}\|^2$ estimates for the piecewise linear Rayleigh-Ritz method and for the finite difference method.

EXERCISES 1.3.

1. For each of the following BVPs, perform the following tasks.

   (i) Use the piecewise linear Rayleigh-Ritz method with $n = 50$ equally spaced grid values to solve the BVP numerically and plot the results.

   (ii) Repeat part (i) with $n = 200$ equally spaced grid points.

   (iii) Repeat part (i) with $n = 500$ equally spaced grid points.

   In each part, first perform all integrals directly, and then repeat using the approximations (48)-(50) as needed. Compare performance times. When it is possible to compute the exact solution using the symbolic toolbox, or if one is given, plot the errors of each approximation obtained.

   (a) $(DE)u'' = e^{8x-[2(x-1)]^2}\cos\left(e^{8x}\right)(BC)u(0) = u(1) = 0$

   (b) $(DE)\left(e^{-3x}u'\right)' - e^{-3x}u = 3\pi\cos(\pi x), (BC)u(0) = u(1) = 0; u_{\text{exact}}(x) = e^{3x}\cos(\pi x)$

   (c) $(DE)(2u')' + 12u = x^3, (BC)u(0) = u(1) = 0; u_{\text{exact}}(x) = \left(x^3 - x\right)/12$

2. Repeat the instructions of Exercise 1 for each of the following BVPs:

   (a) $(DE)u'' = \cos(2x) + \sin(16x)/8, (BC)u(0) = u(1) = 0$

   (b) $(DE)\left(\left(1+x^2\right)u'\right)' = 2, (BC)u(0) = u(1) = 0; u_{\text{exact}}(x) = \ln\left(x^2+1\right)$

   (c) $\begin{cases} (DE) - u'' + 400u = -400\cos^2(\pi x) - 2\pi^2\cos(2\pi x) \\ (BC)u(0) = u(1) = 0 \end{cases}$

   $u_{\text{exact}}(x) = e^{20x}/\left(e^{20}+1\right) + e^{-20x}/\left(e^{-20}+1\right) - \cos^2(\pi x)$

3. Repeat each part of Exercise 1 for each of the BVPs given, but this time choose the indicated number of interior nodes randomly, using the `rand` function.

4. Repeat each part of Exercise 2 for each of the BVPs given, but this time choose the indicated number of nodes according to the properties of the coefficient and right-hand-side data.

5. For each of the BVPs given below, use the piecewise linear Rayleigh-Ritz method in conjunction with the method of Exercise for the Reader 10.15 to numerically solve the BVP according to each of the following node deployments:

   (i) Use $n = 50$ equally spaced interior nodes. Repeat with each of $n = 200$ and $n = 500$.

   (ii) Use $n = 250$ randomly chosen interior nodes. Repeat with each of $n = 200$ and $n = 500$.

   (iii) Use $n = 250$ nodes deployed (without equal spaces) in a way that seems reasonable from given data. Repeat with each of $n = 200$ and $n = 500$.

Whenever possible, graph the errors of each of these approximations.

(a) The beam-deflection problem of Example 10.3.

(b) The BVP of Exercise 3(a) of Section 10.2.

(c) The BVP of Exercise 3(c) of Section 10.2.

6. *(A Problematic Choice of Basis Functions)* Consider applying the Rayleigh-Ritz method to our model problem (24) $\begin{cases} -u''(x) = f(x), 0 < x < 1 \\ u(0) = 0, u(1) = 0 \end{cases}$ using the following bases: $\{\phi_i(x)\}_{i=1}^n$ where $\phi_i(x) = x^i(1-x)$.

   (a) Use the Rayleigh-Ritz method with this basis and $n = 50$ to re-solve the (BVP) (24) of Example 10.7. How does the solution compare with the "exact" solution found in part (c) ofthat example?

   (b) Try to repeat using $n = 500$ basis functions. What happens?

   (c) Show that $\langle \phi_i', \phi_j' \rangle = \frac{(i+1)(j+1)}{i+j+1} + \frac{(i+2)(j+2)}{i+j+3} - \frac{(i+1)(j+2)+(i+2)(j+1)}{i+j+2}$, for any $i, j > 0$.

   (d) Make a plot of the condition numbers (use cond (A)) of the $n \times n$ stiffness matrix A as a function of n as n ranges from 2 to 100 . Recall (Chapter 7) that large condition numbers make linear systems difficult to solve.

   (e) How would matters change if we instead used $\phi_i(x) = x^i$ as our basis functions?

7. Repeat each part of Exercise 2 for each of the BVPs given, but this time adapt the Rayleigh-Ritz method using the basis functions $\varphi_k(x) = \sin(k\pi x), k = 1, 2, \cdots, n$ of Exercise for the Reader 10.14.

8. Consider once again the (BVP) $\begin{cases} -u''(x) + 6u(x) = e^{10x}\cos(12x) \quad 0 < x < 1 \\ u(0) = 0, u(1) = 0 \end{cases}$ of the Example 10.8.

   (a) Use the cubic spline Rayleigh-Ritz method with $n = 500$ equally spaced interior grid values to solve this BVP and plot the resulting approximation. Keep a record of the computing time it takes to determine the load vector and stiffness matrix coefficients.

   (b) Graph the error of the numerical solution by using the exact solution as in the last example.

9. Repeat each part of Exercise 2 for each of the BVPs given, but this time use the cubic spline Rayleigh-Ritz method. Compare the results (and errors, when possible) with the numerical solutions obtained in Exercise 2.

10. *(Natural Boundary Conditions)* This exercise will show how to develop the Galerkin method for BVPs with non-Dirichlet boundary conditions on the following model problem:

$$\begin{cases} -u''(x) = f(x), 0 < x < 1 \\ u(0) = 0, u'(1) = 0 \end{cases}.$$

For this problem we use the following for our admissible functions:

$$\mathscr{A}_1 = \{v : [0,1] \to \mathbb{R} : v(x) \text{ is continuous,}$$

$v'(x)$ is piecewise continuous and bounded, and $v(0) = 0\}$. Notice the only difference with this and the class (28) of the model problem (24) considered in the text is that this class has one less requirement: the condition $v(l) = 0$ is no longer essential.

   (a) Use the DE and integrate by parts (as in step 3 of the proof of Theorem 10.5) to show that any solution of the BVP satisfies the corresponding PVW: $\langle u', v' \rangle = \langle f, v \rangle$ for all $v \in \mathscr{A}$ converse is also true and so the PVW is equivalent to the BVP just as in Theorem 10.5. This gives rise to a Galerkin method for numerically solving the BVP, given any basis of a finitedimensional subspace of A. The fact that no boundary condition is required at $x = 1$ for admissible functions in this method has motivated the terminology of a **natural boundary condition** at $x = 1$ as opposed to an **essential boundary condition** like the one at $x = 0$. It is quite surprising that even though the natural boundary conditions force no conditions on the admissible functions, the solution of the PVW will automatically satisfy them.

   (b) (Piecewise Linear Galerkin Method) Given a partition $\mathscr{P}$ of $[0,1]$, we let

$$\mathscr{A}_1(\mathscr{P}) = \{v : [0,1] \to \mathbb{R} : v(x) \text{ is continuous on } [0,1], \text{ linear on each } I_i \text{ and } v(0) = 0\}.$$

   The hat functions $\phi_i(x)(1 \le i \le n)$ need one more function to be added to form a basis of $\mathscr{A}_1(\mathscr{P})$. The function $\phi_{n+1}(x) \in \mathscr{A}_1(\mathscr{P})$ defined by $\phi_{n+1}(x_j) = 0, (j = 0, 1, \cdots, n)$ and $\phi_{n+1}(1) = 1$ will do the job. By substituting a linear combination of these $\sum_{i=1}^{n+1} c_i\phi_i(x)$ into the PVW, set up a linear system for the resulting Galerkin method.

   (c) Apply the method using $n = 50$ equally spaced interior nodes to the BVP in case $f(x) = e^{2x}\cos(\pi x)$. Compute the error by comparing with the exact solution (obtainable with the symbolic toolbox).

(d) Repeat part (c) with $n = 200$.

(e) What can be said in general about the stiffness matrix for this method (e.g., is it invertible, symmetric, positive definite)?

11. (Natural Boundary Conditions)Parts (a) through (c): Go through each part of Exercise 10 for the BVP $\begin{cases} -u''(x) = f(x), 0 < x < 1 \\ u'(0) = 0, u(1) = 0 \end{cases}$, making changes where needed.
(f) What happens if we try to develop a similar method when both boundary conditions are natural: $u'(0) = 0, u'(1) = 0$?

12. Complete the justification of the approximations (48) through (50).

13. Suppose that $b-a = h$ and that $p(x)$ is a function on $[a,b]$ whose second derivative is continuous on $[a,b]$ (i.e., $p(x)$ is in the space $\mathscr{C}^2([a,b])$). Let $p_\ell(x)$ be the linear function which agrees with $p(x)$ at $x = a$ and $x = b$. Show that for any x in $[a,b]$, we have $|p_\ell(x) - p(x)| = O\left(h^2\right)$. Next use this to show that $\left|\int_a^b p_\ell(x) - p(x)dx\right| = O\left(h^3\right)$
Suggestion: For Part (b), use the mean value theorem from calculus to find a number c in $[a,b]$ for which $p'(c) = p'_\ell(c)$. Next use Taylor's theorem to write $p(x) = p_\ell(x) + p''(z_x)(x-c)^2/2$ for any x in $[a,b]$, where z_x is an number between x and c. From this we get that $|p(x) - p_\ell(x)| \leq \max_{a\leq z\leq b} |p''(z)|(x-c)^2/2$ and the assertions readily follow.

14. Derive the Galerkin method for the BVP (41) : $\begin{cases} -(p(x)u'(x))' + q(x)u(x) = f(x), 0 < x < 1 \\ u(0) = 0, u(1) = 0 \end{cases}$ by mimicking step 3 of the proof of Theorem 10.5. Does the method agree with the Rayleigh-Ritz method?

15. Suppose that $g(x)$ is a continuous function on $[0,1]$ that satisfies $\int_0^1 g(x)v(x)dx = 0$ for every $v \in \mathscr{A}$. Prove that $g(x) \equiv 0$ by providing more details to the following outline.
Sketch of Proof: Suppose that $g(x_0) > 0$ for some $x_0 \in (0,1)$. Then by continuity, $g(x) > 0$ for all x in some interval $(x_0 - h, x_0 + h)$. Let $v(x)$ be a hat function with $v(x_0) = 1$, $v(x_0 \pm h) = 0$. Show that $\int_0^1 g(x)v(x)dx > 0$ and this hat function is admissible. This contradiction shows that we cannot have such an $x_0 \in (0,1)$. Conclude similarly that there is no $x_0 \in (0,1)$ for which $g(x_0) < 0$.

16. The proof of Theorem 10.5 made use of one external theorem stating the existence of a solution of the (BVP) (24). In this exercise, you are to follow an outline to prove that part (c) of this theorem implies part (a) by using only the assumption that $u''(x)$ exists and is piecewise continuous and bounded whenever $u(x)$ is a solution of (PVW).

Write (PVW) in the form $\int_0^1 u'(x)v'(x)dx = \int_0^1 f(x)v(x)dx \quad (v \in \mathscr{A})$. Fix a function $v \in \mathscr{A}$ integrate this by parts to obtain $\int_0^1 [u''(x) + f(x)]v(x)dx = 0$. Now use Exercise 16 to show the differential equation (24) must hold.

NOTE: Much error analysis for Rayleigh-Ritz methods depends on certain integral inequalities. A prototypical inequality of this sort is the so-called **Cauchy-Bunyakowski-Schwarz (CBS)** inequality, which reads as follows:

$$|\langle u, v\rangle| \leq |\langle u, u\rangle|^{1/2} |\langle v, v\rangle|^{1/2}, \tag{1.43}$$

valid for any functions u, v for which the inner product (25) is defined. In integral form (see (25)) the CBS inequality becomes:

$$\left|\int_0^1 u(x)v(x)dx\right| \leq \left(\int_0^1 u(x)^2 dx\right)^{1/2} \left(\int_0^1 v(x)^2 dx\right)^{1/2}.$$[14]

[14]The CBS inequality is a good example of an important mathematical result whose history is often subject to political bias. Cauchy was the first to discover a discrete version (for sums) of the inequality. Bunykowski was the first to discover, in 1859, the integral version of the CBS inequality as written above. Schwarz generalized Bunykowski's result some 25 years later to general inner products. Subsequent mathematical literature from each of the three countries usually attributes any version (from sums to general inner products) of the CBS inequality solely to their mathematical contributor. Thus, in French literature it is usually called Cauchy's inequality, etc. All three of these individuals were eminent mathematical figures in their respective countries. Cauchy began work in 1810 as a civil engineer, but his passion for mathematics kept him trying hard to land positions in mathematics. After numerous attempts he finally got one five years later. Cauchy's output was amazingâĂŤhis complete works spanned over practically all areas of mathematics and filled 27 volumes. His textbooks were used for many years in most French universities. Despite his keen mathematical abilities, however, his strong religious positions and often criticism for his contemporaries made it difficult for him to retain desirable positions. Bunyakowski actually earned his doctorate under Cauchy in Paris in 1825. He then went to St. Petersburg where he spent most of his mathematical career. Schwarz originally entered what is now known as the Technical University of Berlin with the intention of earning a degree in chemistry. This school had the top German mathematics department at the time and the lessons of his mathematics teachers (including the famous analyst Karl Weierstrass (1815-1897)) led him to switch his major and eventually earn a doctorate in mathematics. Schwarz had a remarkable potential in blending analytical and geometrical methods that led him to discover many important results. After he took over Weierstrass's professorial position in 1892, however, he had already begun shifting his focus away from research being his main priority and his output decreased to a less remarkable level. At about this time, the main mathematics institute in Germany shifted from Berlin to GÃűttingen.

Figure 1.16(a): Augustin Louis Cauchy (1789-1857), French mathematician

Figure 1.16(b): Viktor Yakovlevich Bunyakowsky (1804-1889), Russian mathematician.

Figure 1.16(c): Hermann Amandus Schwarz (1843-1921), German mathematician.

In manipulations with such integrals, it is often convenient to introduce the norm notation: $\|u\| = |\langle u,u\rangle|^{1/2} = \left(\int_0^1 u(x)^2 dx\right)^{1/2}$. Using this notation the Cauchy-Bunyakowski-Schwarz inequality takes on the more elegant form:

$$|\langle u,v\rangle| \le \|u\|\|v\|.$$

For a further discussion of such concepts and, in particular, a proof of the Cauchy-BunyakowskiSchwarz inequality, we refer the reader to any good book on analysis, for example [Ros-96] or [Rud64]. The next few exercises will give examples of such uses of the CBS inequality.

17. Show that the function norm defined above satisfies the three vector norm axioms (see Chapter 7, equations (36A-C)). For simplicity, assume, in your proofs, that all functions are continuous on $[0,1]$.

    (a) $\|u\| \ge 0, \|u\| = 0$ if and only if $u(x) = 0$ for all x in $[0,1]$.

    (b) $\|cu\| = |c|\|u\|$, for any scalar c.

    (c) $\|u+v\| \le \|u\| + \|v\|$ (triangle inequality).

    Suggestions: For an idea for part (a), see Exercise 15. For part (c) use the CBS inequality.
    Note: If we allow more general functions, such as piecewise continuous functions in some $\mathscr{A}(\mathscr{P})$, then we have to change the condition in part (a) to $u(x) = 0$ for all x in $[0,1]$ except for a possible finite set of exceptions.

18. *(Rayleigh-Ritz Approximations Have Errors of Minimum Internal Elastic Energy)* Recall that the internal elastic energy of an (admissible) function v was defined to be $(1/2)\langle v',v'\rangle = (1/2)\int_0^1 (v'(x))^2 dx$. Follow the outline below to prove the following useful and interesting error estimate which shows that among all admissible (piecewise linear) admissible functions, the Rayleigh-Ritz approximation to the solution of the BVP (24) $\begin{cases} -u''(x) = f(x), 0 < x < 1 \\ u(0) = 0, u(1) = 0 \end{cases}$ is the best possible approximation if errors are measured by internal elastic energies. That is, if $u(x)$ is the solution of (24) and $u_{\mathscr{P}}(x)$ is the (piecewise linear) Rayleigh-Ritz approximation, both corresponding to a partition $\mathscr{P}$ of $[0,1]$, then we have:

$$\left\|(u-u_{\mathscr{P}})'\right\| \le \|(u-v)' \text{ for all } v \in \mathscr{A}(\mathscr{P}). \tag{1.44}$$

    (a) Show that $\left\langle (u-u_g)', v\right\rangle = 0$ for any $v \in \mathscr{A}(\mathscr{P})$ by using the principle of virtual work..

    (b) For any $v \in \mathscr{A}(\mathscr{P})$, note that $w \equiv u_{\mathscr{P}} - v \in \mathscr{A}(\mathscr{P})$. Use (55) to write

$$\left\langle (u-u_{\mathscr{P}})', (u-u_{\mathscr{P}})'\right\rangle = \left\langle (u-u_{\mathscr{P}})', (u-v)'\right\rangle.$$

    (c) Next, use the CBS inequality to obtain (56).

19. *(Comparison of Solution with Linear Interpolant)*

    (a) Let $\bar{u}_{\mathscr{P}} \in \mathscr{A}(\mathscr{P})$ be the (piecewise) **linear interpolant** of the solution u of (24), i.e., $v(x_i) = u(x_i)$ at each partition point (and $\bar{u}_{\mathscr{P}}$ is linear between partition points). Let x be any number in $[0,1]$ that lies between two partition points of $\mathscr{P}: x_j < x < x_{j+1}$. Use the mean value theorem from calculus to show why we can write

$$\bar{u}_{\mathscr{P}}(x) = u(c) + (x-c)u'(x) \text{ for some fixed number } c, x_j < c < x_{j+1}.$$

    (So c depends only on j, but not on the particular value of x.)

(b) In the notation of part (a) use Taylor's theorem and only the assumption that u has a continuous second derivative to show that for any $x, x_j < x < x_{j+1}$, we have

$$|u(x) - \bar{u}_{\mathscr{P}}(x)| \leq \frac{h_j^2}{2} \max_{x_j<x<x_{j+1}} \left|u''(x)\right|.$$

Next use the differential equation of (24) to translate this estimate to the form

$$|u(x) - \bar{u}_{\mathscr{P}}(x)| \leq \frac{\|\mathscr{P}\|^2}{2} \max_{x_j<x<x_{j+1}} |f(x)|,$$

and that this is valid for all x in $[0,1]$.

(c) By applying a similar analysis as done in parts (a) and (b) except now on the derivatives of the above two functions, obtain the following estimate for all x in $[0,1]$ (except the partition points at which $\bar{u}'_{\mathscr{P}}(x)$ may not exist):

$$\left|u'(x) - \bar{u}'_{\mathscr{P}}(x)\right| \leq \|\mathscr{P}\| \max_{0\leq x\leq 1} |f(x)|.$$

20. *(Error Estimate for Rayleigh-Ritz Approximation)* This exercise will provide an outline for using the estimates of the preceding exercises to obtain an estimate for the error of the RayleighRitz approximation. We will show that if $u_{\mathscr{P}}(x)$ is the Rayleigh-Ritz approximation corresponding to a partition $\mathscr{P}$ of $[0,1]$ of the BVP (24): $\begin{cases} -u''(x) = f(x), 0 < x < 1 \\ u(0) = 0, u(1) = 0 \end{cases}$, where $f(x)$ is a continuous function, then we have the following error estimate valid for all $x, 0 \leq x \leq 1 : |u_{\mathscr{P}}(x) - u(x)| \leq \|\mathscr{P}\| \max_{0\leq x\leq 1} |f(x)|$.

(a) Use (54) with v taken to be the linear interpolant $\tilde{u}_{gp}$ of Exercise 19, and then use the results of Exercise 19 to justify the following string of inequalities:

$$\begin{aligned} \left\|(u-u_{\mathscr{P}})'\right\| \leq \left\|(u-\tilde{u}_{\mathscr{P}})'\right\| &= \left(\int_0^1 \left[(u-\tilde{u}_{\mathscr{P}})'(x)\right]^2 dx\right)^{1/2} \\ &\leq \left(\int_0^1 \left[\|\mathscr{P}\| \max_{0\leq x\leq 1} |f(x)|\right]^2 dx\right)^{1/2} \leq \|\mathscr{P}\| \max_{0\leq x\leq 1} |f(x)| \end{aligned}$$

(b) Since $u_{\mathscr{P}}(0) = u(0) = 0$, we can write $u(x) - u_{\mathscr{P}}(x) = \int_0^x (u-u_{\mathscr{P}})'(x)dx$. Use this and the CBS inequality to justify the following string of inequalities, thereby completing the proof of Theorem 10.6.

$$|u(x) - u_{\mathscr{P}}(x)| \leq \left\langle (u-\tilde{u}_{\mathscr{P}})', 1\right\rangle \leq \left\|(u-\tilde{u}_{\mathscr{P}})'\right\| \cdot \|1\| \leq \|\mathscr{P}\| \max_{0\leq x\leq 1} |f(x)|$$

21. *(Refined Error Estimate for Rayleigh-Ritz Approximation Using Green's Functions)* In this exercise we will show that when the Rayleigh-Ritz method is to solve the BVP (24): $\begin{cases} -u''(x) = f(x), 0 < x < 1 \\ u(0) = 0, u(1) = 0 \end{cases}$, where $f(x)$ is a continuous function for some partition 5 of $[0,1]$, then the Rayleigh-Ritz solution $u_{\mathscr{P}}$ actually coincides with the linear interpolant $\tilde{u}_{\mathscr{P}}$ of Exercise 19. From this it will follow from the estimate of part (b) of Exercise 19 that the estimate of Theorem 10.6 is valid. The key is the introduction of so-called **Green's functions** for the BVP. For each interior node x_i of $\mathscr{P}$ the following Green's function:

$$G_i(x) = \begin{cases} (1-x_i)x, & \text{for } 0 \leq x \leq x_i \\ x_i(1-x), & \text{for } x_i \leq x \leq 1 \end{cases}$$

(a) Show that $G_i(x) \in \mathscr{A}(\mathscr{P})$ and that for any $v \in \mathscr{A}$, we have: $\langle v', G_i' \rangle = v(x_i)$.

(b) Take $v = u - u_{\mathscr{P}}$ in part (a) and apply a result from one of the preceding exercises to show that $v(x_i) = 0$ and hence $u = u_{\mathscr{P}}$, as desired.

22. Show directly that the BVP (24) $\begin{cases} -u''(x) = f(x), 0 < x < 1 \\ u(0) = 0, u(1) = 0 \end{cases}$ has a unique solution.
Suggestion: Integrate the DE once to get $u'(x) = \int_0^x -f(t)dt$ and once more to get $u(x) = \int_0^x \int_0^t -f(s)dsdt + C$

23. Using the direct approach to solving the BVP (24) suggested in the preceding exercise, set up and execute a MATLAB code for solving the BVP in Example 10.7 once again. Arrange your parameters so that the total error of your numerical solution is no more than 10^4.
Suggestion: You may wish to try some different approaches using MATLAB's built-in integrator in conjunction with Simpson's rule or the trapezoidal rule (see any standard calculus textbook or [BuFa-01]) and perhaps even the Symbolic Toolbox if you have access to it.

24. (a) Verify that the general solution of the DE $-u'' = \lambda u$ for $\lambda > 0$ is given by $u = C\cos(\sqrt{\lambda}x) + D\sin(\sqrt{\lambda}x)$ for arbitrary constants C and D.

    (b) Show that if we also require the boundary conditions $u(0) = u(1) = 0$, then the resulting BVP will only have nontrivial solutions if $\lambda = (k\pi)^2, k = 1, 2, \cdots$ and these (eigenfunctions) are $u_k(x) = \sin(k\pi x)$.

    (c) Prove the orthogonality relations for the eigenfunctions: $\langle u_k, u_\ell \rangle = \delta_{k\ell}/2$, where $\delta_{k\ell}$ denotes the Kronecker delta.

    (d) Prove the following orthogonality relations for the derivatives of the eigenfunctions: $\left(u_k', u_i'\right) = k\ell\pi^2\delta_{kl}/2$.