Data structure issues (for example, those which arise when studying sparse matrix methods) are standardized by reliance on appropriate commands. MATLAB has facilities for audio and image file input and output. Differential equations simulations are simple to realize, due to the animation commands built into MATLAB. These goals can all be achieved in other ways. But it is helpful to have one package that will run on almost all operating systems and simplify the details so that students can focus on the real mathematical issues. Appendix B is a short MATLAB tutorial that can be used as an introduction to students or as a reference for those already familiar with the software.

The text comes with a CD that contains MATLAB programs taken directly from the text. The CD is available on dual platforms. These programs are also available on the Web site www.aw-bc.com/sauer, where new material and updates will be posted for users to download.

Unique to this text are solutions manuals for both instructors and students. The **Instructor's Solutions Manual** (ISBN: 0-321-28685-5) contains detailed solutions to the odd-numbered exercises, and answers to the even-numbered exercises. To provide help for students, the **Student's Solutions Manual** (ISBN: 0-321-28686-3) contains worked-out solutions to selected exercises. The manuals also show how to use MATLAB software as an aid to solving the types of problems that are presented in the exercises.

The Addison-Wesley Math Tutor Center is staffed by qualified mathematics and statistics instructors who provide students with tutoring on examples and odd-numbered exercises from the textbook. Tutoring is available via toll-free telephone, toll-free fax, e-mail, and the Internet. Interactive, web-based technology allows tutors and students to view and work through problems together in real time over the Internet. For more information, please visit our Web site at www.aw-bc.com/tutorcenter or call us at 1-888-777-0463.

Numerical Analysis is structured to move from foundational, elementary ideas at the outset to more sophisticated concepts later in the presentation. Chapter 0 provides fundamental building blocks for later use. Some instructors like to start at the beginning; others (including the author) prefer to start at Chapter 1 and fold in topics from Chapter 0 when required. Chapters 1 and 2 cover equation-solving in its various forms. Chapter 3 treats the fitting of data by interpolation, and Chapter 4 introduces fitting by least-squares methods. In the succeeding Chapters 5–8, we return to the classical numerical analysis areas of continuous mathematics—numerical differentiation and integration, and the solution of ordinary and partial differential equations with initial and boundary conditions.

Chapter 9 develops random numbers in order to provide complementary methods to Chapters 5–8: the Monte-Carlo alternative to the standard numerical integration schemes, and the counterpoint of stochastic differential equations, necessary when uncertainty is present in the model.

Compression is a core topic of numerical analysis, even though it often hides in plain sight in interpolation, least squares, and Fourier analysis. Modern compression techniques are featured in Chapters 10 and 11. In the former, the Fast Fourier Transform is treated as a device to carry out trigonometric interpolation, both in the exact and least squares sense. Links to audio compression are emphasized and fully carried out in Chapter 11 on the Discrete Cosine Transform and Huffman coding, the standard workhorse for modern audio and image compression. Chapter 12 on eigenvalues and singular values is also written to emphasize connections to data compression, which are growing in importance in contemporary applications. The final Chapter 13 provides a short introduction to optimization techniques.

3. Explain how to most accurately compute the two roots of the equation $x^2 + bx - 10^{-12} = 0$, where b is a number greater than 100.

4. Prove formula (0.14).

0.4 Computer Problems

1. Calculate the expressions that follow in double precision arithmetic (using MATLAB, for example) for $x = 10^{-1}, \ldots, 10^{-14}$. Then, using an alternative form of the expression that doesn't suffer from subtracting nearly equal numbers, repeat the calculation and make a table of results. Report the number of correct digits in the original expression for each x.

$$\text{(a)} \quad \frac{1 - \sec x}{\tan^2 x} \qquad \text{(b)} \quad \frac{1 - (1 - x)^3}{x}$$

2. Find the smallest value of p for which the expression calculated in double precision arithmetic at $x = 10^{-p}$ has no correct significant digits. (Hint: First find the limit of the expression as $x \to 0$.)

$$\text{(a)} \quad \frac{\tan x - x}{x^3} \qquad \text{(b)} \quad \frac{e^x + \cos x - \sin x - 2}{x^3}$$

3. Consider a right triangle whose legs are of length 3344556600 and 1.2222222. How much longer is the hypotenuse than the longer leg? Give your answer with at least four correct digits.

0.5 REVIEW OF CALCULUS

Some important basic facts from calculus will be necessary later. The Intermediate Value Theorem and the Mean Value Theorem are important for solving equations in Chapter 1. Taylor's Theorem is important for understanding interpolation in Chapter 3 and becomes of paramount importance for solving differential equations in Chapters 6, 7, and 8.

The graph of a continuous function has no gaps. For example, if the function is positive for one x-value and negative for another, it must pass through zero somewhere. This fact is basic for getting equation solvers to work in the next chapter. The first theorem formalizes this notion.

THEOREM 0.4 (Intermediate Value Theorem) Let f be a continuous function on the interval $[a, b]$. Then f realizes every value between $f(a)$ and $f(b)$. More precisely, if y is a number between $f(a)$ and $f(b)$, then there exists a number c with $a \le c \le b$ such that $f(c) = y$. ■

EXAMPLE 0.7 Show that $f(x) = x^2 - 3$ on the interval $[1, 3]$ must take on the values 0 and 1.

Because $f(1) = -2$ and $f(3) = 6$, all values between -2 and 6, including 0 and 1, must be taken on by f. For example, setting $c = \sqrt{3}$, note that $f(c) = f(\sqrt{3}) = 0$, and secondly, $f(2) = 1$.

◀

EXAMPLE 1.4 Use Fixed-Point Iteration to find a root of $\cos x = \sin x$.

The simplest way to convert the equation to a fixed point problem is to add x to each side of the equation. We can rewrite the problem as

$$x + \cos x - \sin x = x$$

and define

$$g(x) = x + \cos x - \sin x. \tag{1.12}$$

The result of applying the Fixed-Point Iteration method to this $g(x)$ is shown in the table.

i	x_i	$g(x_i)$	$e_i = \lvert x_i - r \rvert$	e_i/e_{i-1}
0	0.0000000	1.0000000	0.7853982	
1	1.0000000	0.6988313	0.2146018	0.273
2	0.6988313	0.8211025	0.0865669	0.403
3	0.8211025	0.7706197	0.0357043	0.412
4	0.7706197	0.7915189	0.0147785	0.414
5	0.7915189	0.7828629	0.0061207	0.414
6	0.7828629	0.7864483	0.0025353	0.414
7	0.7864483	0.7849632	0.0010501	0.414
8	0.7849632	0.7855783	0.0004350	0.414
9	0.7855783	0.7853235	0.0001801	0.414
10	0.7853235	0.7854291	0.0000747	0.415
11	0.7854291	0.7853854	0.0000309	0.414
12	0.7853854	0.7854035	0.0000128	0.414
13	0.7854035	0.7853960	0.0000053	0.414
14	0.7853960	0.7853991	0.0000022	0.415
15	0.7853991	0.7853978	0.0000009	0.409
16	0.7853978	0.7853983	0.0000004	0.444
17	0.7853983	0.7853981	0.0000001	0.250
18	0.7853981	0.7853982	0.0000001	1.000
19	0.7853982	0.7853982	0.0000000	

There are several interesting things to notice in the table. First, the iteration appears to converge to 0.7853982. Since $\cos \pi/4 = \sqrt{2}/2 = \sin \pi/4$, the true solution to the equation $\cos x - \sin x = 0$ is $r = \pi/4 \approx 0.7853982$. The fourth column is the "error column." It shows the absolute value of the difference between the best guess x_i at step i and the actual fixed point r. This difference becomes small near the bottom of the table, indicating convergence toward a fixed point.

Notice the pattern in the error column. The errors seem to decrease by a constant factor, each error being somewhat less than half the previous error. To be more precise, the ratio between successive errors is shown in the final column. In most of the table, we are seeing the ratio e_{i+1}/e_i of successive errors to approach a constant number, about 0.414. In other words, we are seeing the linear convergence relation

$$e_i \approx 0.414 e_{i-1}. \tag{1.13}$$

than x. Suggest a Fixed-Point Iteration on the basis of this fact, and use Theorem 1.6 to decide whether it will converge to the cube root of A.

10. Improve the cube root algorithm of Exercise 9 by reweighting the average. Setting $g(x) = wx + (1 - w)A/x^2$ for some fixed number $0 < w < 1$, what is the best choice for w?

11. Consider Fixed-Point Iteration applied to $g(x) = 1 - 5x + \frac{15}{2}x^2 - \frac{5}{2}x^3$. (a) Show that $1 - \sqrt{3/5}$, 1, and $1 + \sqrt{3/5}$ are fixed points. (b) Show that none of the three fixed points are locally convergent. (Computer Problem 7 investigates this example further.)

12. Show that the initial guesses 0, 1, and 2 lead to a fixed point in Exercise 11. What happens to other initial guesses close to those numbers?

13. Assume that $g(x)$ is continuously differentiable and that the Fixed-Point Iteration $g(x)$ has exactly three fixed points, $r_1 < r_2 < r_3$. Assume also that $|g'(r_1)| = 0.5$ and $|g'(r_3)| = 0.5$. List all values of $|g'(r_2)|$ that are possible under these conditions.

14. Assume that g is a continuously differentiable function and that the Fixed-Point Iteration $g(x)$ has exactly three fixed points, -3, 1, and 2. Assume that $g'(-3) = 2.4$ and that FPI started sufficiently near the fixed point 2 converges to 2. Find $g'(1)$.

15. Prove the variant of Theorem 1.6: If g is continuously differentiable and $|g'(x)| \leq B < 1$ on an interval $[a, b]$ containing the fixed point r, then FPI converges to r from any initial guess in $[a, b]$.

16. Prove that a continuously differentiable function $g(x)$ satisfying $|g'(x)| < 1$ on a closed interval cannot have two fixed points on that interval.

17. Consider Fixed-Point Iteration with $g(x) = x - x^3$. (a) Show that $x = 0$ is the only fixed point. (b) Show that if $0 < x_0 < 1$, then $x_0 > x_1 > x_2 \ldots > 0$. (c) Show that FPI converges to $r = 0$, while $g'(0) = 1$. (Hint: use the fact that every bounded monotonic sequence converges to a limit.)

18. Consider Fixed-Point Iteration with $g(x) = x + x^3$. (a) Show that $x = 0$ is the only fixed point. (b) Show that if $0 < x_0 < 1$, then $x_0 < x_1 < x_2 < \ldots$. (c) Show that FPI fails to converge to a fixed point, while $g'(0) = 1$. Together with Exercise 17, this shows that FPI may converge to a fixed point r or diverge from r when $|g'(r)| = 1$.

19. Consider the equation $x^3 + x - 2 = 0$, with root $r = 1$. Add the term cx to both sides and divide by c to obtain $g(x)$. (a) For what c is FPI locally convergent to $r = 1$?(b) For what c will FPI converge fastest?

20. Assume that Fixed-Point Iteration is applied to a twice continuously differentiable function $g(x)$ and that $g'(r) = 0$ for a fixed point r. Show that if FPI converges to r, then the error obeys $\lim_{i\to\infty}(e_{i+1})/e_i^2 = M$, where $M = |g''(r)|/2$.

21. Define Fixed-Point Iteration on the equation $x^2 + x = 5/16$ by isolating the x term. Find both fixed points, and determine which initial guesses lead to each fixed point under iteration. (Hint: Plot $g(x)$, and draw cobweb diagrams.)

22. Find the set of all initial guesses for which the Fixed-Point Iteration $x \to 4/9 - x^2$ converges to a fixed point.

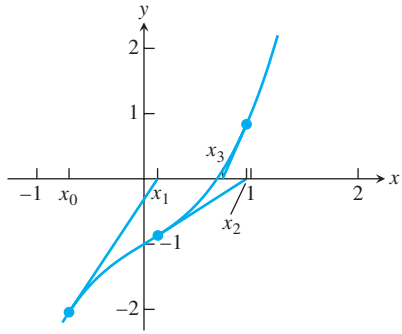

Figure 1.9 Three steps of Newton's Method. Illustration of Example 1.11. Starting with $x_0 = -0.7$, the Newton's Method iterates are plotted along with the tangent lines. The method appears to be converging to the root.

◀

1.4.1 Quadratic convergence of Newton's Method

The convergence in Example 1.11 is qualitatively faster than the linear convergence we have seen for the Bisection Method and Fixed-Point Iteration. A new definition is needed.

DEFINITION 1.10 Let e_i denote the error after step i of an iterative method. The iteration is **quadratically convergent** if

$$M = \lim_{i \to \infty} \frac{e_{i+1}}{e_i^2} < \infty.$$

■

THEOREM 1.11 Let f be twice continuously differentiable and $f(r) = 0$. If $f'(r) \neq 0$, then Newton's Method is locally and quadratically convergent to r. The error e_i at step i satisfies

$$\lim_{i \to \infty} \frac{e_{i+1}}{e_i^2} = M,$$

where

$$M = \left| \frac{f''(r)}{2f'(r)} \right|.$$

■

Proof. To prove local convergence, note that Newton's Method is a particular form of Fixed-Point Iteration, where

$$g(x) = x - \frac{f(x)}{f'(x)},$$

section ends with the description of Brent's Method, a hybrid method which combines the best features of iterative and bracketing methods.

1.5.1 Secant Method and variants

The Secant Method is similar to the Newton's Method, but replaces the derivative by a difference quotient. Geometrically, the tangent line is replaced with a line through the two last known guesses. The intersection point of the "secant line" is the new guess.

An approximation for the derivative at the current guess x_i is the difference quotient

$$\frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}}.$$

A straight replacement of this approximation for $f'(x_i)$ in Newton's Method yields the Secant Method.

Secant Method

$$x_0, x_1 = \text{initial guesses}$$
$$x_{i+1} = x_i - \frac{f(x_i)(x_i - x_{i-1})}{f(x_i) - f(x_{i-1})} \quad \textbf{for } i = 1, 2, 3, \ldots.$$

Unlike Fixed-Point Iteration and Newton's Method, two starting guesses are needed to begin the Secant Method.

It can be shown that, under the assumption that the Secant Method converges to r and $f'(r) \neq 0$, the approximate error relationship

$$e_{i+1} \approx \left| \frac{f''(r)}{2f'(r)} \right| e_i e_{i-1}$$

holds and that this implies that

$$e_{i+1} \approx \left| \frac{f''(r)}{2f'(r)} \right|^{\alpha - 1} e_i^{\alpha},$$

where $\alpha = (1 + \sqrt{5})/2 \approx 1.62$. (See Exercise 6.) The convergence of the Secant Method to simple roots is called **superlinear**, meaning that it lies between linearly and quadratically convergent methods.

EXAMPLE 1.16 Apply the Secant Method with starting guesses $x_0 = 0, x_1 = 1$ to find the root of $f(x) = x^3 + x - 1$.

The formula gives

$$x_{i+1} = x_i - \frac{(x_i^3 + x_i - 1)(x_i - x_{i-1})}{x_i^3 + x_i - (x_{i-1}^3 + x_{i-1})}. \tag{1.34}$$

Starting with $x_0 = 0$ and $x_1 = 1$, we compute

$$x_2 = 1 - \frac{(1)(1 - 0)}{1 + 1 - 0} = \frac{1}{2}$$

$$x_3 = \frac{1}{2} - \frac{-\frac{3}{8}(1/2 - 1)}{-\frac{3}{8} - 1} = \frac{7}{11},$$

```
     6         0.682225  -0.000246683        interpolation
     7         0.682328 -5.43508e-007        interpolation
     8         0.682328  1.50102e-013        interpolation
     9         0.682328             0        interpolation
Zero found in the interval: [0, 1].

ans=

   0.68232780382802
```

Alternatively, the command

```
>> fzero('x^3+x-1',1)
```

looks for a root of $f(x)$ near $x = 1$ by first locating a bracketing interval and then applying Brent's Method.

1.5 Exercises

1. Apply two steps of the Secant Method to the equation with initial guesses $x_0 = 1$ and $x_1 = 2$. (a) $x^3 = 2x + 2$ (b) $e^x + x = 7$ (c) $e^x + \sin x = 4$
2. Apply two steps of the Method of False Position with initial bracket $[1, 2]$ to the equations of Exercise 1.
3. Apply two steps of Inverse Quadratic Interpolation to the equations of Exercise 1. Use initial guesses $x_0 = 1, x_1 = 2$, and $x_2 = 0$, and update by retaining the three most recent iterates.
4. A commercial fisher wants to set the net at a water depth where the temperature is 40 degrees F. By dropping a line with a thermometer attached, she finds that the temperature is 38 degrees at a depth of 12 meters, and 46 at a depth of 5 meters. Use the Secant Method to determine a best estimate for the depth at which the temperature is 40.
5. Derive equation (1.36) by substituting $y = 0$ into (1.35).
6. If the Secant Method converges to r, $f'(r) \neq 0$, and $f''(r) \neq 0$, then the approximate error relationship $e_{i+1} \approx |f''(r)/(2f'(r))|e_i e_{i-1}$ can be shown to hold. Prove that if in addition $\lim_{i\to\infty} e_{i+1}/e_i^{\alpha}$ exists and is nonzero for some $\alpha > 0$, then $\alpha = (1 + \sqrt{5})/2$ and $e_{i+1} \approx |f''(r)/2f'(r)|^{\alpha-1} e_i^{\alpha}$.

1.5 Computer Problems

1. Use the Secant Method to find the (single) solution of each equation in Exercise 1.
2. Use the Method of False Position to find the solution of each equation in Exercise 1.
3. Use Inverse Quadratic Interpolation to find the solution of each equation in Exercise 1.
4. Set $f(x) = 54x^6 + 45x^5 - 102x^4 - 69x^3 + 35x^2 + 16x - 4$. Plot the function on the interval $[-2, 2]$, and use the Secant Method to find all five roots in the interval. To which of the roots is the convergence linear, and to which is it superlinear?

is $||A|| = 2.0001$, according to (2.20). The inverse of A is

$$A^{-1} = \begin{bmatrix} -10000 & 10000 \\ 10001 & -10000 \end{bmatrix},$$

which has norm $||A^{-1}|| = 20001$. The condition number of A is

$$\text{cond}(A) = (2.0001)(20001) = 40004.0001.$$

This is exactly the error magnification we found in Example 2.11, which evidently achieves the worst case, defining the condition number. The error magnification factor for any other b in this system will be less than or equal to 40004.0001. Exercise 3 asks for the computation of some of the other error magnification factors.

The significance of the condition number is the same as in Chapter 1. Error magnification factors of the magnitude $\text{cond}(A)$ are possible. In floating point arithmetic, the relative backward error cannot be expected to be less than ϵ_{mach}, since storing the entries of b already causes errors of that size. According to (2.19), relative forward errors of size $\epsilon_{\text{mach}} \cdot \text{cond}(A)$ are possible in solving $Ax = b$. In other words, if $\text{cond}(A) \approx 10^k$, we should prepare to lose k digits of accuracy in computing x.

In Example 2.11, $\text{cond}(A) \approx 4 \times 10^4$, so in double precision we should expect about $16 - 4 = 12$ correct digits in the solution x. We can test this by introducing MATLAB's best general-purpose linear equation solver: \.

In MATLAB, the backslash command `x = A\b` solves the linear system by using an advanced version of the LU factorization that we will explore in Section 2.4. For now, we will use it as an example of what we can expect from the best possible algorithm operating in floating point arithmetic. The following MATLAB commands deliver the computer solution x_c of Example 2.11:

```
>> A = [1 1;1.0001 1]; b=[2;2.0001];
>> xc = A\b
xc =
   1.00000000000222
   0.99999999999778
```

Compared with the correct solution $x = [1, 1]$, the computed solution has about 11 correct digits, close to the prediction from the condition number.

The Hilbert matrix H, with entries $H_{ij} = 1/(i + j - 1)$, is notorious for its large condition number.

EXAMPLE 2.12 Let H denote the $n \times n$ Hilbert matrix. Use MATLAB's \ to compute the solution of $Hx = b$, where $b = H \cdot [1, \ldots, 1]^T$, for $n = 6$ and 10.

The right-hand side b is chosen to make the correct solution the vector of n ones, for ease of checking the forward error. MATLAB finds the condition number (in the infinity norm) and computes the solution:

```
>> n=6;H=hilb(n);
>> cond(H,inf)
ans =
    2.907027900294064e+007
>> b=H*ones(n,1);
>> xc=H\b
```

EXAMPLE 3.9 Interpolate $f(x) = 1/(1 + 12x^2)$ at evenly-spaced points in $[-1, 1]$.

This is called the **Runge example**. The function has the same general shape as the triangular bump in Figure 3.5. Figure 3.6 shows the result of the interpolation, behavior that is characteristic of the Runge phenomenon: polynomial wiggle near the ends of the interpolation interval.

◀

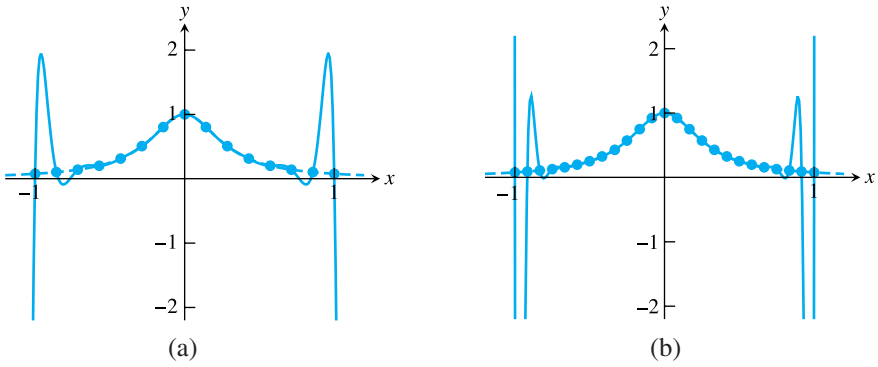

Figure 3.6 Runge example. Polynomial interpolation of the Runge function of Example 3.9 at evenly spaced base points causes extreme variation near the ends of the interval, similar to Figure 3.5. (a) 15 base points (b) 25 base points.

As we have seen, examples with the Runge phenomenon characteristically have large error near the outside of the interval of data points. The cure for this problem is intuitive: Move some of the interpolation points toward the outside of the interval, where the function producing the data can be better fit. We will see how to accomplish this in the next section on Chebyshev interpolation.

3.2 Exercises

1. (a) Find the degree 2 interpolating polynomial $P_2(x)$ through the points $(0, 0)$, $(\pi/2, 1)$, and $(\pi, 0)$. (b) Calculate $P_2(\pi/4)$, an approximation for $\sin(\pi/4)$. (c) Use Theorem 3.3 to give an error bound for the approximation in part (b). (d) Using a calculator or MATLAB, compare the actual error to your error bound.

2. (a) Given the data points $(1, 0)$, $(2, \ln 2)$, $(4, \ln 4)$, find the degree 2 interpolating polynomial. (b) Use the result of (a) to approximate $\ln 3$. (c) Use Theorem 3.3 to give an error bound for the approximation in part (b). (d) Compare the actual error to your error bound.

3. Assume that the polynomial $P_9(x)$ interpolates the function $f(x) = e^{-2x}$ at the 10 evenly-spaced points $x = 0, 1/9, 2/9, 3/9, \ldots, 8/9, 1$. (a) Find an upper bound for the error $|f(1/2) - P_9(1/2)|$. (b) How many decimal places can you guarantee to be correct if $P_9(1/2)$ is used to approximate e^{-1}?

4. Consider the interpolating polynomial for $f(x) = 1/(x + 5)$ with interpolation nodes $x = 0, 2, 4, 6, 8, 10$. Find an upper bound for the interpolation error at (a) $x = 1$ and (b) $x = 5$.

Gauss-Newton Method

To minimize

$$r_1(x)^2 + \cdots + r_m(x)^2.$$

Set $x^0 =$ initial vector,
for $k = 0, 1, 2, \ldots$

$$\begin{aligned} Dr(x^k)^T Dr(x^k) v^k &= -Dr(x^k)^T r(x^k) \\ x^{k+1} &= x^k + v^k \end{aligned} \tag{4.31}$$

end

Notice that each step of the Gauss-Newton Method is reminiscent of the normal equations, where the coefficient matrix has been replaced by Dr. The Gauss-Newton Method solves for a root of the gradient of the squared error. Although the gradient must be zero at the minimum, the converse is not true, so it is possible for the method to converge to a maximum or a neutral point. Caution must be used in interpreting the algorithm's result.

Two intersecting circles intersect in one or two points, unless the circles coincide. Three circles in the plane, however, typically have no points of common intersection. In such a case, we can ask for the point in the plane that comes closest to being an intersection point in the sense of least squares.

EXAMPLE 4.19 Consider the three circles in the plane with centers $(x_1, y_1) = (-1, 0)$, $(x_2, y_2) = (1, 1/2)$, $(x_3, y_3) = (1, -1/2)$ and radii $R_1 = 1$, $R_2 = 1/2$, $R_3 = 1/2$, respectively. Use the Gauss-Newton Method to find the point for which the sum of the squared distances to the three circles is minimized.

The circles are shown in Figure 4.11(a). The point (x, y) in question minimizes the sum of the squares of the residual errors:

$$\begin{aligned} r_1(x, y) &= \sqrt{(x - x_1)^2 + (y - y_1)^2} - R_1 \\ r_2(x, y) &= \sqrt{(x - x_2)^2 + (y - y_2)^2} - R_2 \\ r_3(x, y) &= \sqrt{(x - x_3)^2 + (y - y_3)^2} - R_3. \end{aligned}$$

This follows from the fact that the distance from a point (x, y) to a circle with center (x_1, y_1) and radius R_1 is $|\sqrt{(x - x_1)^2 + (y - y_1)^2} - R_1|$ (see Exercise 3). The Jacobian of $r(x, y)$ is

$$Dr(x, y) = \begin{bmatrix} \frac{x - x_1}{S_1} & \frac{y - y_1}{S_1} \\ \frac{x - x_2}{S_2} & \frac{y - y_2}{S_2} \\ \frac{x - x_3}{S_3} & \frac{y - y_3}{S_3} \end{bmatrix},$$

where $S_i = \sqrt{(x - x_i)^2 + (y - y_i)^2}$ for $i = 1, 2, 3$. The Gauss-Newton iteration with initial vector $(x^0, y^0) = (0, 0)$ converges to $(\overline{x}, \overline{y}) = (0.412891, 0)$ within six correct decimal places after seven steps.

13. Develop a first-order method for approximating $f''(x)$ that uses the data $f(x-h)$, $f(x)$, and $f(x+3h)$ only. Find the error term.

14. (a) Apply extrapolation to the formula developed in Exercise 13 to get a second-order formula for $f''(x)$. (b) Demonstrate the order of the new formula by approximating $f''(0)$, where $f(x)=\cos x$, with $h=0.1$ and $h=0.01$.

15. Develop a second-order method for approximating $f'(x)$ that uses the data $f(x-2h)$, $f(x)$, and $f(x+3h)$ only. Find the error term.

16. Find $E(h)$, an upper bound for the error of the machine approximation of the two-point forward-difference formula for the first derivative. Follow the reasoning preceding (5.11). Find the h corresponding to the minimum of $E(h)$.

17. Prove the second-order formula for the third derivative

$$f'''(x)=\frac{-f(x-2h)+2f(x-h)-2f(x+h)+f(x+2h)}{2h^3}+O(h^2).$$

18. Prove the second-order formula for the third derivative

$$f'''(x)=\frac{f(x-3h)-6f(x-2h)+12f(x-h)-10f(x)+3f(x+h)}{2h^3}+O(h^2).$$

19. Prove the second-order formula for the fourth derivative

$$f^{(iv)}(x)=\frac{f(x-2h)-4f(x-h)+6f(x)-4f(x+h)+f(x+2h)}{h^4}+O(h^2).$$

This formula is used in Reality Check 2.

20. This exercise justifies the beam equations (2.42) and (2.43) in Reality Check 2. Let $f(x)$ be a five-times continuously differentiable function.

(a) Prove that if $f(x)=f'(x)=0$, then

$$f^{(\mathrm{iv})}(x)=\frac{12f(x+h)-6f(x+2h)+\frac{4}{3}f(x+3h)}{h^4}-\frac{6}{5}f^{(\mathrm{v})}(c)h.$$

(b) Prove that if $f''(x)=f'''(x)=0$, then

$$f^{(\mathrm{iv})}(x)=\frac{12f(x-3h)-24f(x-2h)+12f(x-h)}{25h^4}+\frac{18}{25}f^{(\mathrm{v})}(c)h.$$

(c) Prove that if $f''(x)=f'''(x)=0$, then

$$f^{(\mathrm{iv})}(x)=\frac{25f(x-4h)-93f(x-3h)+111f(x-2h)-43f(x-h)}{25h^4}+\frac{217}{100}f^{(\mathrm{v})}(c)h.$$

21. Use Taylor expansions to prove that (5.16) is a fourth-order formula.

22. The error term in the two-point forward-difference formula for $f'(x)$ can be written in other ways. Prove the alternative result

$$f'(x) = \frac{f(x+h) - f(x)}{h} - \frac{h}{2}f''(x) - \frac{h^2}{6}f'''(c),$$

where c is between x and $x + h$. We will use this error form in the derivation of the Crank-Nicolson Method in Chapter 8.

23. Investigate the reason for the name extrapolation. Assume that $F(h)$ is an nth order formula for approximating a quantity Q, and consider the points $(Kh^n, F(h))$ and $(K(h/2)^n, F(h/2))$ in the xy-plane, where error is plotted on the x-axis and the formula output on the y-axis. Find the line through the two points (the best functional approximation for the relationship between error and F). The y-intercept of this line is the value of the formula when you extrapolate the error to zero. Show that this extrapolated value is given by formula (5.15).

5.1 Computer Problems

1. Make a table of the error of the three-point centered-difference formula for $f'(0)$, where $f(x) = \sin x - \cos x$, with $h = 10^{-1}, \ldots, 10^{-12}$, as in the table in Section 5.1.2. Draw a plot of the results. Does the minimum error correspond to the theoretical expectation?

2. Make a table and plot of the error of the three-point centered-difference formula for $f'(1)$, as in Computer Problem 1, where $f(x) = x^{-1}$.

3. Make a table and plot of the error of the two-point forward-difference formula for $f'(0)$, as in Computer Problem 1, where $f(x) = \sin x - \cos x$. Compare your answers with the theory developed in Exercise 16.

4. Make a table and plot as in Problem 3, but approximate $f'(1)$, where $f(x) = x^{-1}$. Compare your answers with the theory developed in Exercise 16.

5. Make a plot as in Problem 1 to approximate $f''(0)$ for $f(x) = \cos x$, using the 3-point centered difference formula. Where does the minimum error appear to occur, in terms of machine epsilon?

5.2 NEWTON-COTES FORMULAS FOR NUMERICAL INTEGRATION

The numerical calculation of definite integrals relies on many of the same tools we have already seen. In Chapters 3 and 4, methods were developed for finding function approximation to a set of data points, using interpolation and least squares modeling. We will discuss methods for **numerical integration**, or **quadrature**, based on both of these ideas.

For example, given a function f defined on an interval $[a, b]$, we can draw an interpolating polynomial through some of the points of $f(x)$. Since it is simple to evaluate the definite integral of a polynomial, this calculation can be used to approximate the integral of $f(x)$. This is the Newton-Cotes approach to approximating integrals. Alternatively, we could find a low-degree polynomial that approximates the function well in the sense of least

6.2 Exercises

1. Using initial condition $y(0) = 1$ and step size $h = 1/4$, calculate the Trapezoid Method approximation $w_0, \ldots, w_4$ on the interval $[0, 1]$. Find the error at $t = 1$ by comparing with the correct solution found in Exercise 6.1.3.

$$\text{(a)} \quad y' = t \quad \text{(b)} \quad y' = t^2 y \quad \text{(c)} \quad y' = 2(t+1)y$$

$$\text{(d)} \quad y' = 5t^4 y \quad \text{(e)} \quad y' = 1/y^2 \quad \text{(f)} \quad y' = t^3/y^2$$

2. Using initial condition $y(0) = 0$ and step size $h = 1/4$, calculate the Trapezoid Method approximation on the interval $[0, 1]$. Find the error at $t = 1$ by comparing with the correct solution found in Exercise 6.1.4.

$$\text{(a)} \quad y' = t + y \quad \text{(b)} \quad y' = t - y \quad \text{(c)} \quad y' = 4t - 2y$$

3. Find the formula for the second-order Taylor Method for the following differential equations: (a) $y' = ty$ (b) $y' = ty^2 + y^3$ (c) $y' = y \sin y$ (d) $y' = e^{yt^2}$

4. Apply the second-order Taylor Method to the initial value problems in Exercise 1. Using step size $h = 1/4$, calculate the second-order Taylor Method approximation on the interval $[0, 1]$. Compare with the correct solution found in Exercise 6.1.3, and find the error at $t = 1$.

5. (a) Prove (6.22). (b) Prove (6.23).

6.2 Computer Problems

1. Apply the explicit Trapezoid Method on a grid of step size $h = 0.1$ in $[0, 1]$ to the initial value problems in Exercise 1. Print a table of the t values, approximations, and global truncation error at each step.

2. Plot the approximate solutions for the IVPs in Exercise 1 on $[0, 1]$ for step sizes $h = 0.1, 0.05$, and 0.025, along with the true solution.

3. For the IVPs in Exercise 1, plot the global truncation error of the explicit Trapezoid Method at $t = 1$ as a function of $h = 0.1 \times 2^{-k}$ for $0 \le k \le 5$. Use a loglog plot as in Figure 6.4.

4. For the IVPs in Exercise 1, plot the global truncation error of the second-order Taylor Method at $t = 1$ as a function of $h = 0.1 \times 2^{-k}$ for $0 \le k \le 5$.

6.3 SYSTEMS OF ORDINARY DIFFERENTIAL EQUATIONS

Approximation of systems of differential equations can be done as a simple extension of the methodology for a single differential equation. Treating systems of equations greatly extends our ability to model interesting dynamical behavior.

The ability to solve systems of ordinary differential equations lies at the core of the art and science of computer simulation. In this section, we introduce two physical systems

6. Adapt `pend.m` to build a damped pendulum with oscillating pivot. The goal is to investigate the phenomenon of parametric resonance, by which the inverted pendulum becomes stable! The equation is

$$y'' + dy' + \left(\frac{g}{l} + A\cos 2\pi t\right)\sin y = 0,$$

where A is the forcing strength. Set $d = 0.1$ and the length of the pendulum to be 2.5 meters. In the absence of forcing $A = 0$, the downward pendulum $y = 0$ is a stable equilibrium, and the inverted pendulum $y = \pi$ is an unstable equilibrium. Find as accurately as possible the range of parameter A for which the inverted pendulum becomes stable. (Of course, $A = 0$ is too small; it turns out that $A = 30$ is too large.) Use the initial condition $y = 3.1$ for your test, and call the inverted position "stable" if the pendulum does not pass through the downward position.

7. Use the parameter settings of Computer Problem 6 to demonstrate the other effect of parametric resonance: The stable equilibrium can become unstable with an oscillating pivot. Find the smallest (positive) value of the forcing strength A for which this happens. Classify the downward position as unstable if the pendulum eventually travels to the inverted position.

8. Adapt `pend.m` to build the double pendulum. A new pair of `rod` and `bob` must be defined for the second pendulum. Note that the pivot end of the second rod is equal to the formerly free end of the first rod: The (x, y) position of the free end of the second rod can be calculated by using simple trigonometry.

9. Adapt `orbit.m` to solve the two-body problem. Set the masses to $m_2 = 0.3$, $m_1 = 0.03$, and plot the trajectories with initial conditions $(x_1, y_1) = (2, 2)$, $(x_1', y_1') = (0.2, -0.2)$ and $(x_2, y_2) = (0, 0)$, $(x_2', y_2') = (-0.01, 0.01)$.

10. Adapt `orbit.m` to solve the three-body problem. Set the masses to $m_2 = 0.3$, $m_1 = m_3 = 0.03$. (a) Plot the trajectories with initial conditions $(x_1, y_1) = (2, 2)$, $(x_1', y_1') = (0.2, -0.2)$, $(x_2, y_2) = (0, 0)$, $(x_2', y_2') = (0, 0)$ and $(x_3, y_3) = (-2, -2)$, $(x_3', y_3') = (-0.2, 0.2)$. (b) Change the initial condition of x_1' to 0.20001, and compare the resulting trajectories. This is a striking visual example of sensitive dependence.

6.4 RUNGE-KUTTA METHODS AND APPLICATIONS

The Runge-Kutta Methods are a family of ODE solvers that include the Euler and Trapezoid Methods, and also more sophisticated methods of higher order. In this section, we introduce a variety of one-step methods and apply them to simulate trajectories of some key applications.

6.4.1 The Runge-Kutta family

We have seen that the Euler Method has order one and the Trapezoid Method has order two. In addition to the Trapezoid Method, there are other second-order methods of the Runge-Kutta type. One important example is the Midpoint Method.

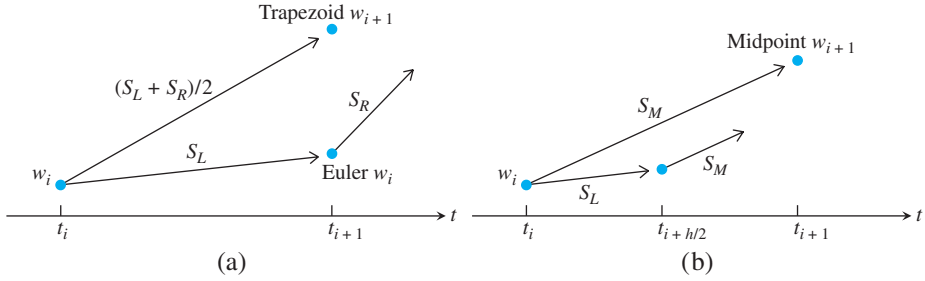

Figure 6.14 Schematic view of two members of the RK2 family. (a) The Trapezoid Method uses an average from the left and right endpoints to traverse the interval. (b) The Midpoint Method uses a slope from the interval midpoint.

Midpoint Method

$$\begin{aligned} w_0 &= y_0 \\ w_{i+1} &= w_i + hf\left(t_i + \frac{h}{2}, w_i + \frac{h}{2} f(t_i, w_i)\right). \end{aligned} \tag{6.46}$$

To verify the order of the Midpoint Method, we must compute its local truncation error. When we did this for the Trapezoid Method, we found the expression (6.31) useful:

$$y_{i+1} = y_i + hf(t_i, y_i) + \frac{h^2}{2}\left(\frac{\partial f}{\partial t}(t_i, y_i) + \frac{\partial f}{\partial y}(t_i, y_i) f(t_i, y_i)\right) + \frac{h^3}{6} y'''(c). \tag{6.47}$$

To compute the local truncation error at step i, we assume that $w_i = y_i$ and calculate $y_{i+1} - w_{i+1}$. Repeating the use of the Taylor series expansion as for the Trapezoid Method, we can write

$$\begin{aligned} w_{i+1} &= y_i + hf\left(t_i + \frac{h}{2}, y_i + \frac{h}{2} f(t_i, y_i)\right) \\ &= y_i + h\left(f(t_i, y_i) + \frac{h}{2}\frac{\partial f}{\partial t}(t_i, y_i) + \frac{h}{2} f(t_i, y_i)\frac{\partial f}{\partial y}(t_i, y_i) + O(h^2)\right). \end{aligned} \tag{6.48}$$

Comparing (6.47) and (6.48) yields

$$y_{i+1} - w_{i+1} = O(h^3),$$

so the Midpoint Method is of order two by Theorem 6.4.

Each function evaluation of the right-hand side of the differential equation is called a **stage** of the method. The Trapezoid and Midpoint Methods are members of the family of two-stage, second-order Runge-Kutta Methods, having form

$$w_{i+1} = w_i + h\left(1 - \frac{1}{2\alpha}\right) f(t_i, w_i) + \frac{h}{2\alpha} f(t_i + \alpha h, w_i + \alpha h f(t_i, w_i)) \tag{6.49}$$

for some $\alpha \neq 0$. Setting $\alpha = 1$ corresponds to the explicit Trapezoid Method, and $\alpha = 1/2$ to the Midpoint Method. Exercise 5 asks you to verify the order of methods in this family.

the parameters is $s = 10, r = 28$, and $b = 8/3$. These settings were used for the trajectory shown in Figure 6.17, computed by order four Runge-Kutta, using the following code to describe the differential equation.

```
function z=ydot(t,y)
%Lorenz equations
s=10; r=28; b=8/3;
z(1)=-s*y(1)+s*y(2);
z(2)=-y(1)*y(3)+r*y(1)-y(2);
z(3)=y(1)*y(2)-b*y(3);
```

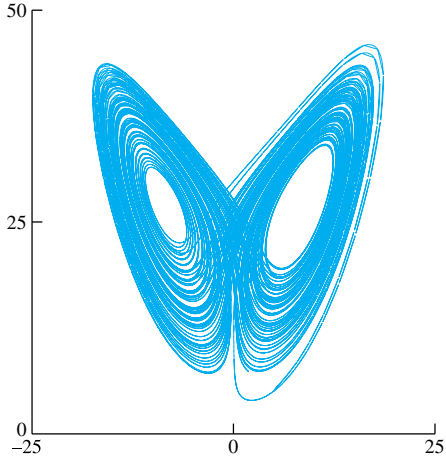

Figure 6.17 One trajectory of the Lorenz equations (6.53), projected to the *xz*-plane. Parameters are set to $s = 10, r = 28$, and $b = 8/3$.

The Lorenz equations are an important example because the trajectories show great complexity, despite the fact that the equations are deterministic and fairly simple (almost linear). The explanation for the complexity is similar to that of the double pendulum or three-body problem: sensitive dependence on initial conditions. Computer Problems 8 and 9 explore the sensitive dependence of this so-called chaotic attractor.

6.4 Exercises

1. Apply the Midpoint Method for the IVPs

$$\text{(a)}\quad y' = t \quad \text{(b)}\quad y' = t^2 y \quad \text{(c)}\quad y' = 2(t+1)y$$

$$\text{(d)}\quad y' = 5t^4 y \quad \text{(e)}\quad y' = 1/y^2 \quad \text{(f)}\quad y' = t^3/y^2$$

with initial condition $y(0) = 1$. Using step size $h = 1/4$, calculate the Midpoint Method approximation on the interval $[0, 1]$. Compare with the correct solution found in Exercise 6.1.3, and find the global truncation error at $t = 1$.

multistep methods can achieve the same order with less computational effort—usually just one function evaluation per step.

Since multistep methods use more than one previous w value, they need help getting started. The start-up phase for an s-step method typically consists of a one-step method that uses w_0 to produce $s - 1$ values $w_1, w_2, \ldots, w_{s-1}$, before the multistep method can be used. The Adams-Bashforth Two-Step Method (6.72) needs w_1, along with the given initial condition w_0, in order to begin. The following MATLAB code uses the Trapezoid Method to provide the start-up value w_1. The command `plot(t,y)` is used to plot the output.

```
% Program 6.7 Multistep method
% Inputs: [inter(1),inter(2)] time interval,
%  ic=[y0]  initial condition,
%  h=stepsize, s=number of (multi)steps, e.g. 2 for 2-step method
% Output:time steps t, solution y
% Calls a multistep method such as ab2step.m
% Example usage: exmultistep ([0,1],1,0.05,2)
function [t,y]=exmultistep(inter,ic,h,s)
n=round((inter(2)-inter(1))/h);
% Start-up phase
y(1,:)=ic;t(1)=int(1);
for i=1:s-1                % start-up phase, using one-step method
  t(i+1)=t(i)+h;
  y(i+1,:)=trapstep(t(i),y(i,:),h);
  f(i,:)=ydot(t(i),y(i,:));
end
for i=s:n                  % multistep method loop
  t(i+1)=t(i)+h;
  f(i,:)=ydot(t(i),y(i,:));
  y(i+1,:)=ab2step(t(i),i,y,f,h);
end
function y=trapstep(t,x,h)
%one step of the Trapezoid Method from section 6.2
z1=ydot(t,x);
g=x+h*z1;
z2=ydot(t+h,g);
y=x+h*(z1+z2)/2;

function z=ab2step(t,i,y,f,h)
%one step of the Adams-Bashforth 2-step method
z=y(i,:)+h*(3*f(i,:)/2-f(i-1,:)/2);

function z=unstable2step(t,i,y,f,h)
%one step of an unstable 2-step method
z=-y(i,:)+2*y(i-1,:)+h*(5*f(i,:)/2+f(i-1,:)/2);

function z=weaklystable2step(t,i,y,f,h)
%one step of a  weakly-stable 2-step method
z=y(i-1,:)+h*2*f(i,:);

function z=ydot(t,y)  % IVP from section 6.1
z=t*y+t^3;
```

Adams-Moulton Four-Step Method (fifth-order)

$$w_{i+1} = w_i + \frac{h}{720}[251 f_{i+1} + 646 f_i - 264 f_{i-1} + 106 f_{i-2} - 19 f_{i-3}]. \tag{6.96}$$

These methods are heavily used in predictor–corrector methods, along with an Adams-Bashforth predictor of the same order. Computer Problems 5 and 6 ask for MATLAB code to implement this idea.

6.7 Exercises

1. Apply the Adams-Bashforth Two-Step Method to the IVPs

   (a) $y' = t$ (b) $y' = t^2 y$ (c) $y' = 2(t+1)y$
   (d) $y' = 5t^4 y$ (e) $y' = 1/y^2$ (f) $y' = t^3/y^2$

   with initial condition $y(0) = 1$. Use step size $h = 1/4$ on the interval $[0, 1]$. Use the Explicit Trapezoid Method to create w_1. Using the correct solution in Exercise 6.1.3, find the global truncation error at $t = 1$.

2. Carry out the steps of Exercise 1 on the IVPs

   (a) $y' = t + y$ (b) $y' = t - y$ (c) $y' = 4t - 2y$

   with initial condition $y(0) = 0$. Use the correct solution from Exercise 6.1.4 to find the global truncation error at $t = 1$.

3. Find a two-step, third-order explicit method. Is the method stable?

4. Find a second-order, two-step explicit method whose characteristic polynomial has a double root at 1.

5. Show that the Implicit Trapezoid Method (6.89) is a second-order method.

6. Explain why the characteristic polynomial of an explicit or implicit s-step method, for $s \geq 2$, must have a root at 1 if its order is at least one.

7. (a) For which a_1 does there exist a strongly stable second-order, two-step explicit method?
   (b) Answer the same question for weakly stable such method.

8. Show that the coefficients of the Adams-Moulton Two-Step Implicit Method satisfy (6.92) and that the method is strongly stable.

9. Find the order and stability type for the following two-step implicit methods:

   (a) $w_{i+1} = 3w_i - 2w_{i-1} + \frac{h}{12}[13 f_{i+1} - 20 f_i - 5 f_{i-1}]$
   (b) $w_{i+1} = \frac{4}{3}w_i - \frac{1}{3}w_{i-1} + \frac{2}{3}h f_{i+1}$
   (c) $w_{i+1} = \frac{4}{3}w_i - \frac{1}{3}w_{i-1} + \frac{h}{9}[4 f_{i+1} + 4 f_i - 2 f_{i-1}]$
   (d) $w_{i+1} = 3w_i - 2w_{i-1} + \frac{h}{12}[7 f_{i+1} - 8 f_i - 11 f_{i-1}]$
   (e) $w_{i+1} = 2w_i - w_{i-1} + \frac{h}{2}[f_{i+1} - f_{i-1}]$

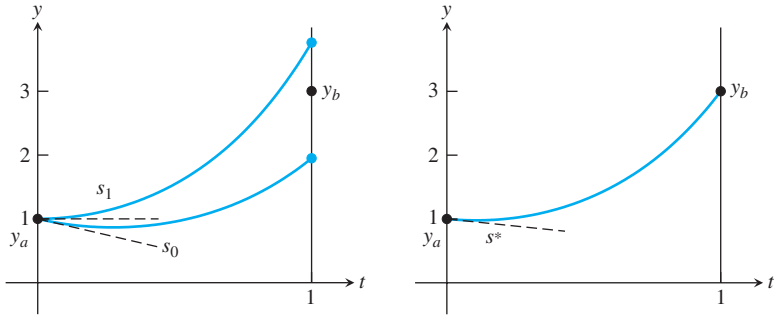

Figure 7.3 The Shooting Method. (a) To solve the BVP, the IVP with initial conditions $y(a)=y_a, y'(a)=s_0$ is solved with initial guess s_0. The value of $F(s_0)$ is $y(b)-y_b$. Then a new s_1 is chosen, and the process is repeated with the goal of solving $F(s)=0$ for s. (b) MATLAB's `ode45` is used with root s^* to plot the solution of the BVP (7.7).

the solution can be found (by an IVP solver as in Chapter 6, for example) as the solution to the initial value problem

$$\begin{cases} y'' = f(t, y, y') \\ y(a) = y_a \\ y'(a) = s^* \end{cases}. \tag{7.6}$$

We show a MATLAB implementation of the shooting method in the next example.

EXAMPLE 7.6 Apply the shooting method to the boundary value problem

$$\begin{cases} y'' = 4y \\ y(0) = 1. \\ y(1) = 3 \end{cases} \tag{7.7}$$

Write the differential equation as a first-order system in order to use MATLAB's `ode45` IVP solver:

$$\begin{aligned} y' &= v \\ v' &= 4y. \end{aligned} \tag{7.8}$$

Write a function file `de.m` as input to `ode45`:

```
function ydot=de(t,y)
ydot=[0;0];
ydot(1)=y(2);
ydot(2)=4*y(1);
```

Write a function file `F.m` as input to `bisect.m` from Chapter 1:

```
function z=F(s)
a=0; b=1; yb=3;
[t,y]=ode45('de',[a,b],[0,s])
z=y(end,1)-yb; % end means last entry of solution y
```

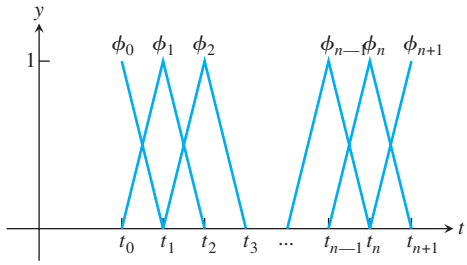

Figure 7.10 Piecewise-linear B-splines used as finite elements. Each $\phi_i(t)$, for $1 \le i \le n$, has support on the interval from t_{i-1} to t_{i+1}.

For a set of data points (t_i, c_i), define the **piecewise-linear B-spline**

$$S(t) = \sum_{i=0}^{n+1} c_i \phi_i(t).$$

It follows immediately from (7.22) that $S(t_j) = \sum_{i=0}^{n+1} c_i \phi_i(t_j) = c_j$. Therefore, $S(t)$ is a piecewise-linear function that interpolates the data points (t_i, c_i). In other words, the y-coordinates are the coefficients! This will simplify the interpretation of the solution (7.21). The c_i are not only the coefficients, but also the solution values at the grid points t_i.

EXAMPLE 7.12 Apply the Finite Element Method to the BVP

$$\begin{cases} y'' = 4y \\ y(0) = 1. \\ y(1) = 3 \end{cases}$$

SPOTLIGHT ON

Orthogonality

We saw in Chapter 4 that the distance from a point to a plane is minimized by drawing the perpendicular segment from the point to the plane. The plane represents candidates to approximate the point; the distance between them is approximation error. This simple fact about orthogonality permeates numerical analysis. It is the core of least squares approximation and is fundamental to the Galerkin approach to boundary value problems and partial differential equations, as well as Gaussian quadrature (Chapter 5), compression (see Chapters 10 and 11), and the solutions of eigenvalue problems (Chapter 12).

Let $\phi_0, \ldots, \phi_{n+1}$ be piecewise-linear B-splines on a grid on $[a, b]$, as shown in Figure 7.10. They will serve as the basis functions for the Galerkin method.

The first and last of the c_i are found from collocation:

$$1 = y(0) = \sum_{i=0}^{n+1} c_i \phi_i(0) = c_0 \phi_0(0) = c_0$$
$$3 = y(1) = \sum_{i=0}^{n+1} c_i \phi_i(1) = c_{n+1} \phi_{n+1}(1) = c_{n+1}.$$

For $i = 1, \ldots, n$, use the finite element equations (7.20):

$$\int_0^1 f(t, y, y') \phi_i(t)\, dt + \int_0^1 y'(t) \phi_i'(t)\, dt = 0.$$

Note that the boundary terms of (7.20) are zero for $i = 1, \ldots, n$.

Now substitute the functional form $y(t) = \sum c_i \phi_i(t)$ and use the differential equation $f(t, y, y') = 4y$ to get

$$\begin{aligned} 0 &= \int_0^1 \left(4\phi_i(t) \sum_{j=0}^{n+1} c_j \phi_j(t) + \sum_{j=0}^{n+1} c_j \phi_j'(t) \phi_i'(t) \right) dt \\ &= \sum_{j=0}^{n+1} c_j \left[4 \int_0^1 \phi_i(t) \phi_j(t)\, dt + \int_0^1 \phi_j'(t) \phi_i'(t)\, dt \right]. \end{aligned}$$

Assume that the grid is evenly-spaced with step size h. We will need the following integrals, for $i = 1, \ldots, n$:

$$\begin{aligned} \int_a^b \phi_i(t) \phi_{i+1}(t)\, dt &= \int_0^h \frac{t}{h} \left(1 - \frac{t}{h}\right) dt = \int_0^h \left(\frac{t}{h} - \frac{t^2}{h^2} \right) dt \\ &= \frac{t^2}{2h} - \frac{t^3}{3h^2} \Big|_0^h = \frac{h}{6} \end{aligned} \tag{7.23}$$

$$\int_a^b (\phi_i(t))^2\, dt = 2 \int_0^h \left(\frac{t}{h} \right)^2 dt = \frac{2}{3} h \tag{7.24}$$

$$\int_a^b \phi_i'(t) \phi_{i+1}'(t)\, dt = \int_0^h \frac{1}{h} \left(-\frac{1}{h} \right) dt = -\frac{1}{h} \tag{7.25}$$

$$\int_a^b (\phi_i'(t))^2\, dt = 2 \int_0^h \left(\frac{1}{h} \right)^2 dt = \frac{2}{h}. \tag{7.26}$$

8.2 Exercises

1. Prove that the functions (a) $u(x,t) = \sin \pi x \cos 4\pi t$, (b) $u(x,t) = e^{-x-2t}$, (c) $u(x,t) = \ln(1 + x + t)$ are solutions of the wave equation with the specified initial-boundary conditions:

(a)
$$\begin{cases} u_{tt} = 16u_{xx} \\ u(x,0) = \sin \pi x \text{ for } 0 \le x \le 1 \\ u_t(x,0) = 0 \text{ for } 0 \le x \le 1 \\ u(0,t) = 0 \text{ for } 0 \le t \le 1 \\ u(1,t) = 0 \text{ for } 0 \le t \le 1 \end{cases}$$

(b)
$$\begin{cases} u_{tt} = 4u_{xx} \\ u(x,0) = e^{-x} \text{ for } 0 \le x \le 1 \\ u_t(x,0) = -2e^{-x} \text{ for } 0 \le x \le 1 \\ u(0,t) = e^{-2t} \text{ for } 0 \le t \le 1 \\ u(1,t) = e^{-1-2t} \text{ for } 0 \le t \le 1 \end{cases}$$

(c)
$$\begin{cases} u_{tt} = u_{xx} \\ u(x,0) = \ln(1 + x) \text{ for } 0 \le x \le 1 \\ u_t(x,0) = 1/(1 + x) \text{ for } 0 \le x \le 1 \\ u(0,t) = \ln(1 + t) \text{ for } 0 \le t \le 1 \\ u(1,t) = \ln(2 + t) \text{ for } 0 \le t \le 1 \end{cases}$$

2. Prove that the functions (a) $u(x,t) = \sin \pi x \sin 2\pi t$, (b) $u(x,t) = (x + 2t)^5$, (c) $u(x,t) = \sinh x \cosh 2t$ are solutions of the wave equation with the specified initial-boundary conditions:

(a)
$$\begin{cases} u_{tt} = 4u_{xx} \\ u(x,0) = 0 \text{ for } 0 \le x \le 1 \\ u_t(x,0) = 2\pi \sin \pi x \text{ for } 0 \le x \le 1 \\ u(0,t) = 0 \text{ for } 0 \le t \le 1 \\ u(1,t) = 0 \text{ for } 0 \le t \le 1 \end{cases}$$

(b)
$$\begin{cases} u_{tt} = 4u_{xx} \\ u(x,0) = x^5 \text{ for } 0 \le x \le 1 \\ u_t(x,0) = 10x^4 \text{ for } 0 \le x \le 1 \\ u(0,t) = 32t^5 \text{ for } 0 \le t \le 1 \\ u(1,t) = (1 + 2t)^5 \text{ for } 0 \le t \le 1 \end{cases}$$

(c)
$$\begin{cases} u_{tt} = 4u_{xx} \\ u(x,0) = \sinh x \text{ for } 0 \le x \le 1 \\ u_t(x,0) = 0 \text{ for } 0 \le x \le 1 \\ u(0,t) = 0 \text{ for } 0 \le t \le 1 \\ u(1,t) = \frac{1}{2}(e - \frac{1}{e}) \cosh 2t \text{ for } 0 \le t \le 1 \end{cases}$$

3. Prove that $u_1(x,t) = \sin \alpha x \cos c\alpha t$ and $u_2(x,t) = e^{x+ct}$ are solutions of the wave equation (8.25).

4. Prove that if $s(x)$ is twice differentiable, then $u(x,t) = s(\alpha x + c\alpha t)$ is a solution of the wave equation (8.25).

5. Prove that the eigenvalues of A in (8.30) lie between $2 - 4\sigma^2$ and 2.

6. Let λ be a complex number. (a) Prove that if $\lambda + 1/\lambda$ is a real number, then $|\lambda| = 1$ or λ is real. (b) Prove that if λ is real and $|\lambda + 1/\lambda| \le 2$, then $|\lambda| = 1$.

8.2 Computer Problems

1. Solve the initial-boundary value problems in Exercise 1 on $0 \le x \le 1, 0 \le t \le 1$ by the Finite Difference Method with $h = 0.05, k = h/c$. Use MATLAB's `mesh` command to plot the solution.

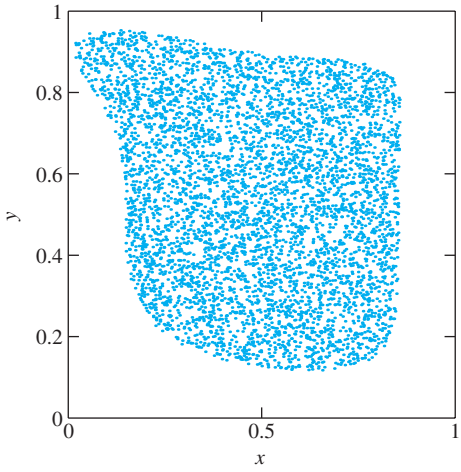

Figure 9.2 Monte Carlo calculation of area. From 10,000 random pairs in $[0,1] \times [0,1]$, the ones that satisfy the inequality in Example 9.2 are plotted. The proportion of plotted random pairs is an approximation to the area.

The random seed $x_0 \neq 0$ is chosen arbitrarily. The nonprime modulus was originally selected to make the modulus operation as fast as possible, and the multiplier was selected primarily because its binary representation was simple. The serious problem with this generator is that it flagrantly disobeys the independence postulate for random numbers. Notice that

$$\begin{aligned} a^2 - 6a &= (2^{16} + 3)^2 - 6(2^{16} + 3) \\ &= 2^{32} + 6 \cdot 2^{16} + 9 - 6 \cdot 2^{16} - 18 \\ &= 2^{32} - 9. \end{aligned}$$

Therefore, $a^2 - 6a + 9 = 0 \pmod{m}$, so

$$\begin{aligned} x_{i+2} - 6x_{i+1} + 9x_i &= a^2 x_i - 6a x_i + 9x_i \pmod{m} \\ &= 0 \pmod{m}. \end{aligned}$$

Dividing by m yields

$$u_{i+2} = 6u_{i+1} - 9u_i \pmod{1}. \tag{9.5}$$

The problem is not that u_{i+2} is predictable from the two previous numbers generated. Of course, it will be predictable even from one previous number, because the generator is deterministic. The problem lies with the small coefficients in the relation (9.5), which make the correlation between the random numbers very noticeable. Figure 9.3(a) shows a plot of 10,000 random numbers generated by `randu` and plotted in triples (u_i, u_{i+1}, u_{i+2}).

of Marsaglia and Tsang [15], essentially a very efficient way of inverting the cumulative distribution function.

9.1 Exercises

1. Find the period of the linear congruential generator defined by (a) $a = 2, b = 0, m = 5$ (b) $a = 4, b = 1, m = 9$.
2. Find the period of the LCG defined by $a = 4, b = 0, m = 9$. Does the period depend on the seed?
3. Approximate the area under the curve $y = x^2$ for $0 \le x \le 1$, using the LCG with (a) $a = 2, b = 0, m = 5$ (b) $a = 4, b = 1, m = 9$.
4. Approximate the area under the curve $y = 1 - x$ for $0 \le x \le 1$, using the LCG with (a) $a = 2, b = 0, m = 5$ (b) $a = 4, b = 1, m = 9$.
5. Consider the RANDNUM-CRAY random number generator, used on the Cray X-MP, one of the first supercomputers. This LCG used $m = 2^{48}, a = 2^{24} + 3$, and $b = 0$. Prove that $u_{i+2} = 6u_{i+1} - 9u_i \pmod 1$. Is this worrisome? See Computer Problems 9 and 10.

9.1 Computer Problems

1. Implement the Minimal Standard random number generator, and find the Monte Carlo approximation of the volume in Example 9.3. Use 10^6 three-dimensional points with seed $x_0 = 1$. How close is your approximation to the correct answer?
2. Implement `randu` and find the Monte Carlo approximation of the volume in Example 9.3, as in Computer Problem 1. Verify that no point (u_i, u_{i+1}, u_{i+2}) enters the given ball.
3. (a) Using calculus, find the area bounded by the two parabolas $P_1(x) = x^2 - x + 1/2$ and $P_2(x) = -x^2 + x + 1/2$. (b) Estimate the area as a Type 1 Monte Carlo simulation, by finding the average value of $P_2(x) - P_1(x)$ on $[0, 1]$. Find estimates for $n = 10^i$ for $2 \le i \le 6$. (c) Same as (b), but estimate as a Type 2 Monte Carlo problem: Find the proportion of points in the square $[0, 1] \times [0, 1]$ that lie between the parabolas. Compare the efficiency of the two Monte Carlo approaches.
4. Carry out the steps of Computer Problem 3 for the subset of the first quadrant bounded by the polynomials $P_1(x) = x^3$ and $P_2(x) = 2x - x^2$.
5. Use $n = 10^4$ pseudo-random points to estimate the interior area of the ellipses (a) $13x^2 + 34xy + 25y^2 \le 1$ in $-1 \le x, y \le 1$ and (b) $40x^2 + 25y^2 + y + 9/4 \le 52xy + 14x$ in $0 \le x, y \le 1$. Compare your estimate with the correct areas (a) $\pi/6$ and (b) $\pi/18$, and report the error of the estimate. Repeat with $n = 10^6$ and compare results.
6. Use $n = 10^4$ pseudo-random points to estimate the interior volume of the ellipsoid defined by $2 + 4x^2 + 4z^2 + y^2 \le 4x + 4z + y$, contained in the unit cube $0 \le x, y, z \le 1$. Compare your estimate with the correct volume $\pi/24$, and report the error. Repeat with $n = 10^6$ points.

7. (a) Use calculus to evaluate the integral $\int_0^1 \int_{x^2}^{\sqrt{x}} xy \, dy \, dx$. (b) Use $n = 10^6$ pairs in the unit square $[0, 1] \times [0, 1]$ to estimate the integral as a Type 1 Monte Carlo problem. (Average the function that is equal to xy if (x, y) is in the integration domain and 0 if not.)

8. Use 10^6 random pairs in the unit square to estimate $\int_A xy \, dx \, dy$, where A is the area described by Example 9.2.

9. Implement the questionable random number generator from Exercise 5, and draw the plot analogous to Figure 9.3.

10. Devise a Monte Carlo approximation problem that completely foils the RANDNUM-CRAY generator of Exercise 5, following the ideas of Example 9.3.

9.2 MONTE CARLO SIMULATION

We have already seen examples of two types of Monte Carlo simulation. In this section, we explore the range of problems that are suited for this technique and discuss some of the refinements that make it work better, including quasi-random numbers. We will need to use the language of random variables and expected values in this section.

9.2.1 Power laws for Monte Carlo estimation

We would like to understand the convergence rate of Monte Carlo simulation. At what rate does the estimation error decrease as the number of points n used in the estimate grows? This is similar to the convergence questions in Chapter 5 for the quadrature methods and in Chapters 6, 7, and 8 for differential equation solvers. In the previous cases, they were posed as questions about error versus step size. Cutting the step size is analogous to adding more random numbers in Monte Carlo simulations.

Think of Type 1 Monte Carlo as the calculation of a function mean using random samples, then multiplying by the volume of the integration region. Calculating a function mean can be viewed as calculating the mean of a probability distribution given by that function. We will use the notation $E(X)$ for the expected value of the random variable X. The **variance** of a random variable X is $E[(X - E(X))^2]$, and the **standard deviation** of X is the square root of its variance. The error expected in estimating the mean will decrease with the number n of random points, in the following way:

Type 1 or Type 2 Monte Carlo with pseudo-random numbers.

$$\text{Error} \propto n^{-\frac{1}{2}} \tag{9.9}$$

To understand this formula, view the integral as the volume of the domain times the mean value A of the function over the domain. Consider the identical random variables X_i corresponding to a function evaluation at a random point. Then the mean value is the expected value of the random variable $Y = (X_1 + \cdots + X_n)/n$, or

$$E\left[\frac{X_1 + \cdots + X_n}{n}\right] = nA/n = A,$$

the probability is $2/\pi$ that the needle will straddle both colors. (a) Prove this result analytically. Consider the distance d of the needle's midpoint to the nearest edge, and its angle θ with the stripes. Express the probability as a simple integral. (b) Design a Monte Carlo Type 2 simulation that approximates the probability, and carry it out with $n = 10^6$ pseudo-random pairs (d, θ).

7. (a) What proportion of 2×2 matrices with entries in the interval $[0, 1]$ have positive determinant? Find the exact value, and approximate with a Monte Carlo simulation. (b) What proportion of symmetric 2×2 matrices with entries in $[0, 1]$ have positive determinant? Find the exact value and approximate with a Monte Carlo simulation.

8. Run a Monte Carlo simulation to approximate the proportion of 2×2 matrices with entries in $[-1, 1]$ whose eigenvalues are both real.

9. What proportion of 4×4 matrices with entries in $[0, 1]$ undergo no row exchanges under partial pivoting? Use a Monte Carlo simulation involving MATLAB's `lu` command to estimate this probability.

9.3 DISCRETE AND CONTINUOUS BROWNIAN MOTION

Although previous chapters of this book have focused largely on principles that are important for the mathematics of deterministic models, these models are only a part of the arsenal of modern techniques. One of the most important applications of random numbers is to make stochastic modeling possible.

We will begin with one of the simplest stochastic models, the random walk, also called discrete Brownian motion. The basic principles that underlie this discrete model are essentially the same for the more sophisticated models that follow, based on continuous Brownian motion.

9.3.1 Random walks

A **random walk** W_t is defined on the real line by starting at $W_0 = 0$ and moving a step of length s_i at each integer time i, where the s_i are independent and identically distributed random variables. Here we will assume each s_i is $+1$ or -1 with equal probability $1/2$. **Discrete Brownian motion** is defined to be the random walk given by the sequence of accumulated steps

$$W_t = W_0 + s_1 + s_2 + \cdots + s_t,$$

for $t = 0, 1, 2, \ldots$ Figure 9.8 illustrates a single realization of discrete Brownian motion.

The following MATLAB code carries out a random walk of 10 steps:

```
t=10;
w=0;
for i=1:t
  if rand>1/2
    w=w+1;
  else
    w=w-1;
  end
end
```

The next one is different:

$$1 + \omega^n + \omega^{2n} + \omega^{3n} + \cdots + \omega^{n(n-1)} = 1 + 1 + 1 + 1 + \cdots + 1 = n. \tag{10.6}$$

This information is collected into the following lemma.

LEMMA 10.1 **Primitive roots of unity**. Let ω be a primitive nth root of unity and k be an integer. Then

$$\sum_{j=0}^{n-1} \omega^{jk} = \begin{cases} n & \text{if } k/n \text{ is an integer} \\ 0 & \text{otherwise} \end{cases}.$$

■

Exercise 6 asks the reader to fill in the details of the proof.

10.1.2 Discrete Fourier Transform

Let $x = [x_0, \ldots, x_{n-1}]^T$ be a (real-valued) n-dimensional vector, and denote $\omega = e^{-i2\pi/n}$. Here is the fundamental definition of this chapter.

DEFINITION 10.2 The **Discrete Fourier Transform** (DFT) of $x = [x_0, \ldots, x_{n-1}]^T$ is the n-dimensional vector $y = [y_0, \ldots, y_{n-1}]$, where $\omega = e^{-i2\pi/n}$ and

$$y_k = \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} x_j \omega^{jk}. \tag{10.7}$$

■

For example, Lemma 10.1 shows that the DFT of $x = [1, 1, \ldots, 1]$ is $y = [\sqrt{n}, 0, \ldots, 0]$. In matrix terms, this definition says

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_{n-1} \end{bmatrix} = \begin{bmatrix} a_0 + ib_0 \\ a_1 + ib_1 \\ a_2 + ib_2 \\ \vdots \\ a_{n-1} + ib_{n-1} \end{bmatrix} = \frac{1}{\sqrt{n}} \begin{bmatrix} \omega^0 & \omega^0 & \omega^0 & \cdots & \omega^0 \\ \omega^0 & \omega^1 & \omega^2 & \cdots & \omega^{n-1} \\ \omega^0 & \omega^2 & \omega^4 & \cdots & \omega^{2(n-1)} \\ \omega^0 & \omega^3 & \omega^6 & \cdots & \omega^{3(n-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ \omega^0 & \omega^{n-1} & \omega^{2(n-1)} & \cdots & \omega^{(n-1)^2} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{n-1} \end{bmatrix}. \tag{10.8}$$

Each $y_k = a_k + ib_k$ is a complex number. The $n \times n$ matrix in (10.8) is called the **Fourier matrix**

$$F_n = \frac{1}{\sqrt{n}} \begin{bmatrix} \omega^0 & \omega^0 & \omega^0 & \cdots & \omega^0 \\ \omega^0 & \omega^1 & \omega^2 & \cdots & \omega^{n-1} \\ \omega^0 & \omega^2 & \omega^4 & \cdots & \omega^{2(n-1)} \\ \omega^0 & \omega^3 & \omega^6 & \cdots & \omega^{3(n-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ \omega^0 & \omega^{n-1} & \omega^{2(n-1)} & \cdots & \omega^{(n-1)^2} \end{bmatrix}. \tag{10.9}$$

Putting the pieces together, this corresponds to the following operations:

$$\sqrt{p}\cdot \texttt{ifft}_{[p]}\sqrt{\frac{p}{n}}\frac{1}{\sqrt{n}}\cdot \texttt{fft}_{[n]} = \frac{p}{n}\cdot \texttt{ifft}_{[p]}\cdot \texttt{fft}_{[n]}. \tag{10.22}$$

Of course, F_p^{-1} can only be applied to a length p vector, so we need to place the degree n Fourier coefficients into a length p vector before inverting. The short program `dftinterp.m` carries out these steps.

```
%Program 10.1 Fourier interpolation
%Interpolate n data points on [c,d] with trig function P(t)
%  and plot interpolant at p (>=n) evenly spaced points.
%Input: interval [c,d], data points x, even number of data
%  points n, even number p>=n
%Output: data points of interpolant xp
function xp=dftinterp(inter,x,n,p)
c=inter(1);d=inter(2);t=c+(d-c)*(0:n-1)/n; tp=c+(d-c)*(0:p-1)/p;
y=fft(x);                       % apply DFT
yp=zeros(p,1);                  % yp will hold coefficients for ifft
yp(1:n/2+1)=y(1:n/2+1);         % move n frequencies from n to p
yp(p-n/2+2:p)=y(n/2+2:n);       % same for upper tier
xp=real(ifft(yp))*(p/n);        % invert fft to recover data
plot(t,x,'o',tp,xp)             % plot data points and interpolant
```

Running the function `dftinterp([0, 1], [−2.2 −2.8 −6.1 −3.9 0.0 1.1 −0.6 −1.1],8,100)`, for example, produces the $p = 100$ plotted points in Figure 10.6 without explicitly using sines or cosines. A few comments on the code are in order. The goal is to apply $\texttt{fft}_{[n]}$, followed by $\texttt{ifft}_{[p]}$, and then multiply by p/n. After applying `fft` to the n values in x, the coefficients in the vector `y` are moved from the n frequencies in $P_n(t)$ to a vector `yp` holding p frequencies, where $p \geq n$. There are many higher frequencies among the p frequencies that are not used by P_n, which leads to zero coefficients in those high frequencies, in positions $n/2 + 2$ to $p/2 + 1$. The upper half of the entries in `yp` gives a recapitulation of the lower half, with complex conjugates and in reverse order, following (10.13). After the DFT is inverted with the `ifft` command, although theoretically the result is real, computationally there may be a small imaginary part due to rounding. This is removed by applying the `real` command.

A particularly simple and useful case is $c = 0, d = n$. The data points x_j are collected at the integer interpolation nodes $s_j = j$ for $j = 0, \ldots, n - 1$. The points (j, x_j) are interpolated by the trigonometric function

$$P_n(s) = \frac{a_0}{\sqrt{n}} + \frac{2}{\sqrt{n}}\sum_{k=1}^{n/2-1}\left(a_k \cos\frac{2k\pi}{n}s - b_k \sin\frac{2k\pi}{n}s\right) + \frac{a_{n/2}}{\sqrt{n}}\cos \pi s. \tag{10.23}$$

In Chapter 11, we will use integer interpolation nodes exclusively, for compatibility with the usual conventions for audio and image data compression algorithms.

10.2 Exercises

1. Use the DFT and Corollary 10.8 to find the trigonometric interpolating function for the following data:

(a)

t	x
0	0
$\frac{1}{4}$	1
$\frac{1}{2}$	0
$\frac{3}{4}$	-1

(b)

t	x
0	1
$\frac{1}{4}$	1
$\frac{1}{2}$	-1
$\frac{3}{4}$	-1

(c)

t	x
0	-1
$\frac{1}{4}$	1
$\frac{1}{2}$	-1
$\frac{3}{4}$	1

(d)

t	x
0	1
$\frac{1}{4}$	1
$\frac{1}{2}$	1
$\frac{3}{4}$	1

2. Use (10.23) to find the trigonometric interpolating function for the following data:

(a)

t	x
0	0
1	1
2	0
3	-1

(b)

t	x
0	1
1	1
2	-1
3	-1

(c)

t	x
0	1
1	2
2	4
3	1

(d)

t	x
0	1
1	0
2	1
3	0

3. Find the trigonometric interpolating function for the following data:

(a)

t	x
0	0
$\frac{1}{8}$	1
$\frac{1}{4}$	0
$\frac{3}{8}$	-1
$\frac{1}{2}$	0
$\frac{5}{8}$	1
$\frac{3}{4}$	0
$\frac{7}{8}$	-1

(b)

t	x
0	1
$\frac{1}{8}$	2
$\frac{1}{4}$	1
$\frac{3}{8}$	0
$\frac{1}{2}$	1
$\frac{5}{8}$	2
$\frac{3}{4}$	1
$\frac{7}{8}$	0

(c)

t	x
0	1
$\frac{1}{8}$	1
$\frac{1}{4}$	1
$\frac{3}{8}$	1
$\frac{1}{2}$	0
$\frac{5}{8}$	0
$\frac{3}{4}$	0
$\frac{7}{8}$	0

(d)

t	x
0	1
$\frac{1}{8}$	-1
$\frac{1}{4}$	1
$\frac{3}{8}$	-1
$\frac{1}{2}$	1
$\frac{5}{8}$	-1
$\frac{3}{4}$	1
$\frac{7}{8}$	-1

4. Find the trigonometric interpolating function for the following data:

(a)

t	x
0	0
1	1
2	0
3	-1
4	0
5	1
6	0
7	-1

(b)

t	x
0	1
1	2
2	1
3	0
4	1
5	2
6	1
7	0

(c)

t	x
0	1
1	0
2	1
3	0
4	1
5	0
6	1
7	0

(d)

t	x
0	-1
1	0
2	0
3	0
4	1
5	0
6	0
7	0

5. Find a version of (10.19) for the interpolating function in the case where n is odd.

10.2 Computer Problems

1. Find the order 8 trigonometric interpolating function $P_8(t)$ for the following data:

(a)

t	x
0	0
$\frac{1}{8}$	1
$\frac{1}{4}$	2
$\frac{3}{8}$	3
$\frac{1}{2}$	4
$\frac{5}{8}$	5
$\frac{3}{4}$	6
$\frac{7}{8}$	7

(b)

t	x
0	2
$\frac{1}{8}$	-1
$\frac{1}{4}$	0
$\frac{3}{8}$	1
$\frac{1}{2}$	1
$\frac{5}{8}$	3
$\frac{3}{4}$	-1
$\frac{7}{8}$	-1

(c)

t	x
0	3
1	1
2	4
3	2
4	3
5	1
6	4
7	2

(d)

t	x
1	1
2	-2
3	5
4	3
5	-2
6	-3
7	1
8	2

Plot the data points and $P_8(t)$.

2. Find the order 8 trigonometric interpolating function $P_8(t)$ for the following data:

(a)

t	x
0	6
$\frac{1}{8}$	5
$\frac{1}{4}$	4
$\frac{3}{8}$	3
$\frac{1}{2}$	2
$\frac{5}{8}$	1
$\frac{3}{4}$	0
$\frac{7}{8}$	-1

(b)

t	x
0	3
$\frac{1}{8}$	1
$\frac{1}{4}$	2
$\frac{3}{8}$	-1
$\frac{1}{2}$	-1
$\frac{5}{8}$	-2
$\frac{3}{4}$	3
$\frac{7}{8}$	0

(c)

t	x
0	1
2	2
4	4
6	-1
8	0
10	1
12	0
14	2

(d)

t	x
-7	2
-5	1
-3	0
-1	5
1	7
3	2
5	1
7	-4

Plot the data points and $P_8(t)$.

3. Find the order $n = 8$ trigonometric interpolating function for $f(t) = e^t$ at the evenly spaced points $(j/8, f(j/8))$ for $j = 0, \ldots, 7$. Plot $f(t)$, the data points, and the interpolating function.

4. Plot the interpolating function $P_n(t)$ on $[0, 1]$ in Computer Problem 3, along with the data points and $f(t) = e^t$ for (a) $n = 16$ (b) $n = 32$.

5. Find the order 8 trigonometric interpolating function for $f(t) = \ln t$ at the evenly spaced points $(1 + j/8, f(1 + j/8))$ for $j = 0, \ldots, 7$. Plot $f(t)$, the data points, and the interpolating function.

6. Plot the interpolating function $P_n(t)$ on $[0, 1]$ in Computer Problem 5, along with the data points and $f(t) = \ln t$ for (a) $n = 16$ (b) $n = 32$.

10.3 THE FFT AND SIGNAL PROCESSING

The DFT Interpolation Theorem 10.6 is just one application of the Fourier transform. In this section, we look at interpolation from a more general point of view, which will show how

EXAMPLE 10.4 Let $[c, d]$ be an interval and let n be an even positive integer. Show that the assumptions of Theorem 10.9 are satisfied for $t_j = c + j(d-c)/n$, $j = 0, \ldots, n-1$, and

$$\begin{aligned} f_0(t) &= \sqrt{\frac{1}{n}} \\ f_1(t) &= \sqrt{\frac{2}{n}} \cos \frac{2\pi(t-c)}{d-c} \\ f_2(t) &= \sqrt{\frac{2}{n}} \sin \frac{2\pi(t-c)}{d-c} \\ f_3(t) &= \sqrt{\frac{2}{n}} \cos \frac{4\pi(t-c)}{d-c} \\ f_4(t) &= \sqrt{\frac{2}{n}} \sin \frac{4\pi(t-c)}{d-c} \\ &\vdots \\ f_{n-1}(t) &= \frac{1}{\sqrt{n}} \cos \frac{n\pi(t-c)}{d-c}. \end{aligned}$$

The matrix is

$$A = \sqrt{\frac{2}{n}} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \cdots & \frac{1}{\sqrt{2}} \\ 1 & \cos\frac{2\pi}{n} & \cdots & \cos\frac{2\pi(n-1)}{n} \\ 0 & \sin\frac{2\pi}{n} & \cdots & \sin\frac{2\pi(n-1)}{n} \\ \vdots & \vdots & & \vdots \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}}\cos\pi & \cdots & \frac{1}{\sqrt{2}}\cos(n-1)\pi \end{bmatrix}. \tag{10.25}$$

Lemma 10.10 shows that the rows of A are pairwise orthogonal.

◀

LEMMA 10.10 Let $n \geq 1$ and k, l be integers. Then

$$\sum_{j=0}^{n-1} \cos\frac{2\pi jk}{n} \cos\frac{2\pi jl}{n} = \begin{cases} n & \text{if both } (k-l)/n \text{ and } (k+l)/n \text{ are integers} \\ \frac{n}{2} & \text{if exactly one of } (k-l)/n \text{ and } (k+l)/n \text{ is an integer} \\ 0 & \text{if neither is an integer} \end{cases}$$

$$\sum_{j=0}^{n-1} \cos\frac{2\pi jk}{n} \sin\frac{2\pi jl}{n} = 0$$

$$\sum_{j=0}^{n-1} \sin\frac{2\pi jk}{n} \sin\frac{2\pi jl}{n} = \begin{cases} 0 & \text{if both } (k-l)/n \text{ and } (k+l)/n \text{ are integers} \\ \frac{n}{2} & \text{if } (k-l)/n \text{ is an integer and } (k+l)/n \text{ is not} \\ -\frac{n}{2} & \text{if } (k+l)/n \text{ is an integer and } (k-l)/n \text{ is not} \\ 0 & \text{if neither is an integer} \end{cases}$$

■

is part of a vast literature on signal processing, and the reader is referred to [9] for further study. In Reality Check 10, we investigate a filter of widespread application called the Wiener filter.

10.3 Exercises

1. Find the best order 2 least squares approximation to the data in Exercise 10.2.1, using the basis functions 1 and $\cos 2\pi t$.
2. Find the best order 3 least squares approximation to the data in Exercise 10.2.1, using the basis functions 1, $\cos 2\pi t$, and $\sin 2\pi t$.
3. Find the best order 4 least squares approximation to the data in Exercise 10.2.3, using the basis functions 1, $\cos 2\pi t$, $\sin 2\pi t$, and $\cos 4\pi t$.
4. Find the best order 4 least squares approximation to the data in Exercise 10.2.4, using the basis functions 1, $\cos \frac{\pi}{4} t$, $\sin \frac{\pi}{4} t$, and $\cos \frac{\pi}{2} t$.
5. Prove Lemma 10.10. (*Hint:* Express $\cos 2\pi\, jk/n$ as $(e^{i2\pi\, jk/n} + e^{-i2\pi\, jk/n})/2$, and write everything in terms of $\omega = e^{-i2\pi/n}$, so that Lemma 10.1 can be applied.)

10.3 Computer Problems

1. Find the least squares trigonometric approximating functions of orders $m = 2$ and 4 for the following data points:

(a)

t	y
0	3
$\frac{1}{4}$	0
$\frac{1}{2}$	-3
$\frac{3}{4}$	0

(b)

t	y
0	2
$\frac{1}{4}$	0
$\frac{1}{2}$	5
$\frac{3}{4}$	1

(c)

t	y
0	5
1	2
2	6
3	1

(d)

t	y
1	-1
2	1
3	4
4	3
5	3
6	2

Using `dftfilter.m`, plot the data points and the approximating functions, as in Figure 10.7.

2. Find the least squares trigonometric approximating functions of orders 4, 6, and 8 for the following data points:

(a)

t	y
0	3
$\frac{1}{8}$	0
$\frac{1}{4}$	-3
$\frac{3}{8}$	0
$\frac{1}{2}$	3
$\frac{5}{8}$	0
$\frac{3}{4}$	-6
$\frac{7}{8}$	0

(b)

t	y
0	1
$\frac{1}{8}$	0
$\frac{1}{4}$	-2
$\frac{3}{8}$	1
$\frac{1}{2}$	3
$\frac{5}{8}$	0
$\frac{3}{4}$	-2
$\frac{7}{8}$	1

(c)

t	y
0	1
$\frac{1}{8}$	2
$\frac{1}{4}$	3
$\frac{3}{8}$	1
$\frac{1}{2}$	-1
$\frac{5}{8}$	-1
$\frac{3}{4}$	-3
$\frac{7}{8}$	0

(d)

t	y
0	4.2
$\frac{1}{8}$	5.0
$\frac{1}{4}$	3.8
$\frac{3}{8}$	1.6
$\frac{1}{2}$	-2.0
$\frac{5}{8}$	-1.4
$\frac{3}{4}$	0.0
$\frac{7}{8}$	1.0

Plot the data points and the approximating functions, as in Figure 10.7.

The rows of an orthogonal matrix are pairwise orthogonal unit vectors. The orthogonality of C follows from the fact that the columns of C^T are the unit eigenvectors of the real symmetric $n \times n$ matrix

$$\begin{bmatrix} 1 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & -1 & 2 & -1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & -1 & 2 & -1 \\ & & & & -1 & 1 \end{bmatrix}. \tag{11.5}$$

Exercise 6 asks the reader to verify this fact.

The fact that C is a real orthogonal matrix is what makes the DCT useful. The Orthogonal Function Interpolation Theorem 10.9 applied to the matrix C implies Theorem 11.2.

THEOREM 11.2

DCT Interpolation Theorem. Let $x = [x_0, \ldots, x_{n-1}]^T$ be a vector of n real numbers. Define $y = [y_0, \ldots, y_{n-1}]^T = Cx$, where C is the Discrete Cosine Transform. Then the real function

$$P_n(t) = \frac{1}{\sqrt{n}} y_0 + \frac{\sqrt{2}}{\sqrt{n}} \sum_{k=1}^{n-1} y_k \cos \frac{k(2t+1)\pi}{2n}$$

satisfies $P_n(j) = x_j$ for $j = 0, \ldots, n-1$. ■

Proof. Follows directly from Theorem 10.9. □

Theorem 11.2 shows that the $n \times n$ matrix C transforms n data points into n interpolation coefficients. Like the Discrete Fourier Transform, the Discrete Cosine Transform gives coefficients for a trigonometric interpolation function. Unlike the DFT, the DCT uses cosine terms only and is defined entirely in terms of real arithmetic.

EXAMPLE 11.1 Use the DCT to interpolate the points $(0, 1), (1, 0), (2, -1), (3, 0)$.

It is helpful to notice, using elementary trigonometry, that the 4×4 DCT matrix can be viewed as

$$C = \frac{1}{\sqrt{2}} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \cos\frac{\pi}{8} & \cos\frac{3\pi}{8} & \cos\frac{5\pi}{8} & \cos\frac{7\pi}{8} \\ \cos\frac{2\pi}{8} & \cos\frac{6\pi}{8} & \cos\frac{10\pi}{8} & \cos\frac{14\pi}{8} \\ \cos\frac{3\pi}{8} & \cos\frac{9\pi}{8} & \cos\frac{15\pi}{8} & \cos\frac{21\pi}{8} \end{bmatrix} = a \begin{bmatrix} a & a & a & a \\ b & c & -c & -b \\ a & -a & -a & a \\ c & -b & b & -c \end{bmatrix}, \tag{11.6}$$

where

$$a = \cos\frac{\pi}{4} = \frac{1}{\sqrt{2}}, b = \cos\frac{\pi}{8} = \frac{\sqrt{2+\sqrt{2}}}{2}, c = \cos\frac{3\pi}{8} = \frac{\sqrt{2-\sqrt{2}}}{2}. \tag{11.7}$$

3. Find the DCT of the following data vectors x, and find the corresponding interpolating function $P_n(t)$ for the data points $(i, x_i), i = 0, \ldots, n-1$ (you may state your answers in terms of the b and c defined in (11.7)):

(a)

t	x
0	1
1	0
2	1
3	0

(b)

t	x
0	1
1	1
2	1
3	1

(c)

t	x
0	1
1	0
2	0
3	0

(d)

t	x
0	1
1	2
2	3
3	4

4. Find the DCT least squares approximation with $m = 2$ terms for the data in Exercise 3.

5. Carry out the trigonometry needed to establish equations (11.6) and (11.7).

6. (a) Prove the trigonometric formula $\cos(x + y) + \cos(x - y) = 2\cos x \cos y$ for any x, y. (b) Show that the columns of C^T are eigenvectors of the matrix T in (11.5), and identify the eigenvalues. (c) Show that the columns of C^T are unit vectors.

7. Extend the DCT Interpolation Theorem 11.2 to the interval $[c, d]$ as follows. Let n be a positive integer and set $\Delta_t = (d - c)/n$. Use the DCT to produce a polynomial $P_n(t)$ that satisfies $P_n(c + j\Delta_t) = x_j$ for $j = 0, \ldots, n-1$.

11.1 Computer Problems

1. Plot the data from Exercise 3, along with the DCT interpolant and the DCT least squares approximation with $m = 2$ terms.

2. Plot the data along with the $m = 4, 6$, and 8 DCT least squares approximations.

(a)

t	x
0	3
1	5
2	−1
3	3
4	1
5	3
6	−2
7	4

(b)

t	x
0	4
1	1
2	−3
3	0
4	0
5	2
6	−4
7	0

(c)

t	x
0	3
1	−1
2	−1
3	3
4	3
5	−1
6	−1
7	3

(d)

t	x
0	4
1	2
2	−4
3	2
4	4
5	2
6	−4
7	2

3. Plot the function $f(t)$, the data points $(j, f(j))$, $j = 0, \ldots, 7$, and the DCT interpolating function. (a) $f(t) = e^{-t/4}$ (b) $f(t) = \cos \frac{\pi}{2} t$.

11.2 TWO-DIMENSIONAL DCT AND IMAGE COMPRESSION

The two-dimensional Discrete Cosine Transform is often used to compress small blocks of an image, as small as 8×8 pixels. The compression is lossy, meaning that some information from the block is ignored. The key feature of the DCT is that it helps organize the information

LEMMA 11.10 Denote by c_j the jth column of the (extended) DCT4 matrix (11.27). Then (a) $c_j = c_{-1-j}$ for all integers j (the columns are symmetric around $j = -\frac{1}{2}$), and (b) $c_j = -c_{2n-1-j}$ for all integers j (the columns are antisymmetric around $j = n - \frac{1}{2}$). ■

Proof. To prove part (a) of the lemma, write $j = -\frac{1}{2} + (j + \frac{1}{2})$ and $-1 - j = -\frac{1}{2} - (j + \frac{1}{2})$. Using the definition (11.27) yields

$$\begin{aligned} c_j &= c_{-\frac{1}{2}+(j+\frac{1}{2})} = \sqrt{\frac{2}{n}} \cos \frac{(i+\frac{1}{2})(j+\frac{1}{2})\pi}{n} = \sqrt{\frac{2}{n}} \cos \frac{(i+\frac{1}{2})(-j-\frac{1}{2})\pi}{n} \\ &= c_{-\frac{1}{2}-(j+\frac{1}{2})} = c_{-1-j} \end{aligned}$$

for $i = 0, \ldots, n-1$.

For the proof of (b), set $r = n - \frac{1}{2} - j$. Then $j = n - \frac{1}{2} - r$ and $2n - 1 - j = n - \frac{1}{2} + r$, and we must show that $c_{n-\frac{1}{2}-r} + c_{n-\frac{1}{2}+r} = 0$. By the cosine addition formula,

$$\begin{aligned} c_{n-\frac{1}{2}-r} &= \sqrt{\frac{2}{n}} \cos \frac{(2i+1)(n-r)\pi}{2n} = \sqrt{\frac{2}{n}} \cos \frac{2i+1}{2}\pi \cos \frac{(2i+1)r\pi}{2n} \\ &\quad + \sqrt{\frac{2}{n}} \sin \frac{2i+1}{2}\pi \sin \frac{(2i+1)r\pi}{2n} \\ c_{n-\frac{1}{2}+r} &= \sqrt{\frac{2}{n}} \cos \frac{(2i+1)(n+r)\pi}{2n} = \sqrt{\frac{2}{n}} \cos \frac{2i+1}{2}\pi \cos \frac{(2i+1)r\pi}{2n} \\ &\quad - \sqrt{\frac{2}{n}} \sin \frac{2i+1}{2}\pi \sin \frac{(2i+1)r\pi}{2n} \end{aligned}$$

for $i = 0, \ldots, n-1$. Since $\cos \frac{1}{2}(2i+1)\pi = 0$ for all integers i, the sum $c_{n-\frac{1}{2}-r} + c_{n-\frac{1}{2}+r} = 0$, as claimed. □

We will use the DCT4 matrix E to build the Modified Discrete Cosine Transform. Assume that n is even. We are going to create a new matrix, using the columns $c_{\frac{n}{2}}, \ldots, c_{\frac{5}{2}n-1}$. Lemma 11.10 shows that, for any integer j, the column c_j can be expressed as one of the columns of DCT4—that is, one of the c_i for $0 \le i \le n-1$, as shown in Figure 11.10, up to a possible sign change.

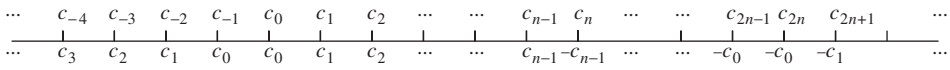

Figure 11.10 Illustration of Lemma 11.10. The columns $c_0, \ldots, c_{n-1}$ make up the $n \times n$ DCT4 matrix. For integers j outside that range, the column defined by c_j in equation (11.27) still corresponds to one of the n columns of DCT4, according to Lemma 11.10.

DEFINITION 11.11 Let n be an even positive integer. The **Modified Discrete Cosine Transform** (MDCT) of $x = (x_0, \ldots, x_{2n-1})^T$ is the n-dimensional vector

$$y = Mx, \tag{11.29}$$

3. Find the characteristic polynomial and the eigenvalues and eigenvectors of the following matrices:

$$\text{(a)}\ \begin{bmatrix} 1 & 0 & 1 \\ 0 & 3 & -2 \\ 0 & 0 & 2 \end{bmatrix} \quad \text{(b)}\ \begin{bmatrix} 1 & 0 & -\frac{1}{3} \\ 0 & 1 & \frac{2}{3} \\ -1 & 1 & 1 \end{bmatrix} \quad \text{(c)}\ \begin{bmatrix} -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{6} \\ -1 & 0 & \frac{1}{3} \\ -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}$$

4. Prove that a square matrix and its transpose have the same characteristic polynomial, and therefore the same set of eigenvalues.

5. Assume that A is a 3×3 matrix with the given eigenvalues. Decide to which eigenvalue Power Iteration will converge, and determine the convergence rate constant S. (a) $\{3, 1, 4\}$ (b) $\{3, 1, -4\}$ (c) $\{-1, 2, 4\}$ (d) $\{1, 9, 10\}$

6. Assume that A is a 3×3 matrix with the given eigenvalues. Decide to which eigenvalue Power Iteration will converge, and determine the convergence rate constant S. (a) $\{1, 2, 7\}$ (b) $\{1, 1, -4\}$ (c) $\{0, -2, 5\}$ (d) $\{8, -9, 10\}$

7. Assume that A is a 3×3 matrix with the given eigenvalues. Decide to which eigenvalue Inverse Power Iteration with the given shift s will converge, and determine the convergence rate constant S. (a) $\{3, 1, 4\}, s = 0$ (b) $\{3, 1, -4\}, s = 0$ (c) $\{-1, 2, 4\}, s = 0$ (d) $\{1, 9, 10\}, s = 6$

8. Assume that A is a 3×3 matrix with the given eigenvalues. Decide to which eigenvalue Inverse Power Iteration with the given shift s will converge, and determine the convergence rate constant S. (a) $\{3, 1, 4\}, s = 5$ (b) $\{3, 1, -4\}, s = 4$ (c) $\{-1, 2, 4\}, s = 1$ (d) $\{1, 9, 10\}, s = 8$

9. Let $A = \begin{bmatrix} 1 & 2 \\ 4 & 3 \end{bmatrix}$. (a) Find all eigenvalues and eigenvectors of A. (b) Apply three steps of Power Iteration with initial vector $x_0 = [1, 0]$. At each step, approximate the eigenvalue by the current Rayleigh quotient. (c) Predict the result of applying Inverse Power Iteration with shift $s = 0$ (d) with shift $s = 3$.

10. Let $A = \begin{bmatrix} -2 & 1 \\ 3 & 0 \end{bmatrix}$. Carry out the steps of Exercise 9 for this matrix.

11. If A is a 6×6 matrix with eigenvalues $-6, -3, 1, 2, 5, 7$, which eigenvalue of A will the following algorithms find? (a) Power Iteration (b) Inverse Power Iteration with shift $s = 4$ (c) Find the linear convergence rates of the two computations. Which converges faster?

12.1 Computer Problems

1. Using the supplied code (or code of your own) for the Power Iteration method, find the dominant eigenvector of A, and estimate the dominant eigenvalue by calculating a Rayleigh quotient. Compare your conclusions with the corresponding part of Exercise 5.

$$\text{(a)}\ \begin{bmatrix} 10 & -12 & -6 \\ 5 & -5 & -4 \\ -1 & 0 & 3 \end{bmatrix} \quad \text{(b)}\ \begin{bmatrix} -14 & 20 & 10 \\ -19 & 27 & 12 \\ 23 & -32 & -13 \end{bmatrix}$$

2. Put the matrix $\begin{bmatrix} 1 & 0 & 2 & 3 \\ -1 & 0 & 5 & 2 \\ 2 & -2 & 0 & 0 \\ 2 & -1 & 2 & 0 \end{bmatrix}$ into upper Hessenberg form.

3. Show that a symmetric matrix in Hessenberg form is tridiagonal.

4. Call a square matrix of nonnegative numbers **stochastic** if the entries of each column add to one. Prove that a stochastic matrix (a) has an eigenvalue equal to one, and (b) all eigenvalues are, at most, one in absolute value.

5. Carry out Normalized Simultaneous Iteration with the following matrices, and explain how it fails:

$$\text{(a)} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad \text{(b)} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

6. (a) Show that the determinant of a matrix in real Schur form is the product of the determinants of the 1×1 and 2×2 blocks on the main diagonal. (b) Show that the eigenvalues of a matrix in real Schur form are the eigenvalues of the 1×1 and 2×2 blocks on the main diagonal.

7. Decide whether the preliminary version of the QR algorithm finds the correct eigenvalues, both before and after changing to Hessenberg form.

$$\text{(a)} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \text{(b)} \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

8. Decide whether the general version of the QR algorithm finds the correct eigenvalues, both before and after changing to Hessenberg form, for the matrices in Exercise 7.

12.2 Computer Problems

1. Apply the shifted QR algorithm (preliminary version `shiftedqr0`) with tolerance 10^{-14} directly to the following matrices:

$$\text{(a)} \begin{bmatrix} -3 & 3 & 5 \\ 1 & -5 & -5 \\ 6 & 6 & 4 \end{bmatrix} \quad \text{(b)} \begin{bmatrix} 3 & 1 & 2 \\ 1 & 3 & -2 \\ 2 & 2 & 6 \end{bmatrix} \quad \text{(c)} \begin{bmatrix} 17 & 1 & 2 \\ 1 & 17 & -2 \\ 2 & 2 & 20 \end{bmatrix} \quad \text{(d)} \begin{bmatrix} -7 & -8 & 1 \\ 17 & 18 & -1 \\ -8 & -8 & 2 \end{bmatrix}$$

2. Apply the shifted QR algorithm directly to find all eigenvalues of the following matrices:

$$\text{(a)} \begin{bmatrix} 3 & 1 & -2 \\ 4 & 1 & 1 \\ -3 & 0 & 3 \end{bmatrix} \quad \text{(b)} \begin{bmatrix} 1 & 5 & 4 \\ 2 & -4 & -3 \\ 0 & -2 & 4 \end{bmatrix} \quad \text{(c)} \begin{bmatrix} 1 & 1 & -2 \\ 4 & 2 & -3 \\ 0 & -2 & 2 \end{bmatrix} \quad \text{(d)} \begin{bmatrix} 5 & -1 & 3 \\ 0 & 6 & 1 \\ 3 & 3 & -3 \end{bmatrix}$$

3. Apply the shifted QR algorithm directly to find all eigenvalues of the following matrices

(a) $\begin{bmatrix} -1 & 1 & 3 \\ 3 & 3 & -2 \\ -5 & 2 & 7 \end{bmatrix}$ (b) $\begin{bmatrix} 7 & -33 & -15 \\ 2 & 26 & 7 \\ -4 & -50 & -13 \end{bmatrix}$ (c) $\begin{bmatrix} 8 & 0 & 5 \\ -5 & 3 & -5 \\ 10 & 0 & 13 \end{bmatrix}$ (d) $\begin{bmatrix} -3 & -1 & 1 \\ 5 & 3 & -1 \\ -2 & -2 & 0 \end{bmatrix}$

4. Repeat Computer Problem 3, but precede the application of the QR iteration with reduction to upper Hessenberg form. Print the Hessenberg form and the eigenvalues.

5. Apply the QR algorithm directly to find all real and complex eigenvalues of the following matrices:

(a) $\begin{bmatrix} 4 & 3 & 1 \\ -5 & -3 & 0 \\ 3 & 2 & 1 \end{bmatrix}$ (b) $\begin{bmatrix} 3 & 2 & 0 \\ -4 & -2 & 1 \\ 2 & 1 & 0 \end{bmatrix}$ (c) $\begin{bmatrix} 7 & 2 & -4 \\ -8 & 0 & 7 \\ 2 & -1 & -2 \end{bmatrix}$ (d) $\begin{bmatrix} 11 & 4 & -2 \\ -10 & 0 & 5 \\ 4 & 1 & 2 \end{bmatrix}$

6. Use the QR algorithm to find the eigenvalues. In each matrix, all eigenvalues have equal magnitude, so Hessenberg may be needed. Compare the results of QR algorithm before and after reduction to Hessenberg form.

(a) $\begin{bmatrix} -5 & -10 & -10 & 5 \\ 4 & 16 & 11 & -8 \\ 12 & 13 & 8 & -4 \\ 22 & 48 & 28 & -19 \end{bmatrix}$ (b) $\begin{bmatrix} 7 & 6 & 6 & -3 \\ -26 & -20 & -19 & 10 \\ 0 & -1 & 0 & 0 \\ -36 & -28 & -24 & 13 \end{bmatrix}$ (c) $\begin{bmatrix} 13 & 10 & 10 & -5 \\ -20 & -16 & -15 & 8 \\ -12 & -9 & -8 & 4 \\ -30 & -24 & -20 & 11 \end{bmatrix}$

Reality Check 12 HOW SEARCH ENGINES RATE PAGE QUALITY

Web search engines such as `Google.com` distinguish themselves by the quality of their returns to search queries. We will discuss a rough approximation of Google's method for judging the quality of web pages by using knowledge of the network of links that exists on the web.

When a web search is initiated, there is a rather complex series of tasks that are carried out by the search engine. One obvious task is word-matching, to find pages that contain the query words, in the title or body of the page. Another key task is to rate the pages that are identified by the first task, to help the user wade through the possibly large set of choices. For very specific queries, there may be only a few text matches, all of which can be returned to the user. (In the early days of the web, there was a game to try to discover search queries that resulted in exactly one hit.) In the case of very specific queries, the quality of the returned pages is not so important, since no sorting may be necessary. The need for a quality ranking becomes apparent for more general queries. For example, the Google query "new automobile" returns several million pages, beginning with automobile buying services, a reasonably useful outcome. How is the ranking determined?

The answer to this question is that `Google.com` assigns a nonnegative real number, called the **page rank**, to each web page that it indexes. The page rank is computed by Google in what is one of the world's largest ongoing Power Iterations for determining eigenvectors.

```
      else              % shrink simplex toward best point
        for j=2:n+1
          x(:,j) = 0.5*x(:,1)+0.5*x(:,j); y(j) = f(x(:,j));
        end
      end
    end
  end
  [y,r] = sort(y);     % resort the obj function values
  x=x(:,r);            %    and rank the vertices the same way
end
```

The code implements the flowchart in Figure 13.5(b). The number of iteration steps is required as an input. Computer Problem 8 asks the reader to rewrite the code with a stopping criterion based on a user-given error tolerance. A common stopping criterion is to require both that the simplex has reduced in size to within a small distance tolerance and that the maximum spread of the function values at the vertices is within a small tolerance. MATLAB implements the Nelder-Mead Method in its `fminsearch` command.

13.1 Exercises

1. Prove that the functions are unimodal on some interval and find the absolute minimum and where it occurs. (a) $f(x) = e^x + e^{-x}$ (b) $f(x) = x^6$ (c) $f(x) = 2x^4 + x$ (d) $f(x) = x - \ln x$
2. Find the absolute minimum in the given interval and at which x it occurs.
   (a) $f(x) = \cos x, [3, 4]$ (b) $f(x) = 2x^3 + 3x^2 - 12x + 3, [0, 2]$
   (c) $f(x) = x^3 + 6x^2 + 5, [-5, 5]$ (d) $f(x) = 2x + e^{-x}, [-5, 5]$

13.1 Computer Problems

1. Plot the function $y = f(x)$, and find a length-one starting interval on which f is unimodal around each relative minimum. Then apply Golden Section Search to locate each of the function's relative minima to within five correct digits.
   (a) $f(x) = 2x^4 + 3x^2 - 4x + 5$ (b) $f(x) = 3x^4 + 4x^3 - 12x^2 + 5$
   (c) $f(x) = x^6 + 3x^4 - 2x^3 + x^2 - x - 7$ (d) $f(x) = x^6 + 3x^4 - 12x^3 + x^2 - x - 7$
2. Apply Successive Parabolic Interpolation to the functions in Computer Problem 1. Locate the minima to within five correct digits.
3. Find the point on the hyperbola $y = 1/x$ closest to the point $(2, 3)$ in two different ways: (a) by Newton's Method applied to find a critical point (b) by Golden Section Search on the square of the distance between a point on the conic and $(2, 3)$.
4. Find the point on the ellipse $4x^2 + 9y^2 = 4$ farthest from $(1, 5)$, using methods (a) and (b) of Computer Problem 3.
5. Use the Nelder-Mead Method to find the minimum of $f(x, y) = e^{-x^2y^2} + (x - 1)^2 + (y - 1)^2$. Try various initial conditions, and compare answers. How many correct digits can you obtain by using this method?

7. (a) $P(x) = (x-1) - (x-1)^2/2 + (x-1)^3/3 - (x-1)^4/4$ (b) $P(0.9) = -0.1053583$, $P(1.1) = 0.0953083$ (c) error bound $= 0.000003387$ for $x = 0.9$, 0.000002 for $x = 1.1$ (d) Actual error ≈ 0.00000218 at $x = 0.9$, 0.00000185 at $x = 1.1$
9. $\sqrt{1+x} = 1 + x/2 \pm x^2/8$. For $x = 1.02$, $\sqrt{1.02} \approx 1.01 \pm 0.00005$. Actual value is $\sqrt{1.02} = 1.0099505$, error $= 0.0000495$

CHAPTER 1

1.1 Exercises

1. (a) $[2, 3]$ (b) $[1, 2]$ (c) $[6, 7]$
3. (a) 2.125 (b) 1.125 (c) 6.875
5. (a) $[2, 3]$ (b) 33 steps

1.1 Computer Problems

1. (a) 2.080083 (b) 1.169726 (c) 6.776092
3. (a) Intervals $[-2, -1], [-1, 0], [1, 2]$, roots $-1.641783, -0.168254, 1.810038$ (b) Intervals $[-2, -1], [-0.5, 0.5], [0.5, 1.5]$, roots $-1.023482, 0.163823, 0.788942$ (c) Intervals $[-1.7, -0.7], [-0.7, 0.3], [0.3, 1.3]$, roots $-0.818094, 0, 0.506308$
5. (a) $[1, 2]$, 27 steps, 1.25992105 (b) $[1, 2]$, 27 steps, 1.44224957 (c) $[1, 2]$, 27 steps, 1.70997595
7. first root -17.188498, determinant correct to 2 places; second root 9.708299, determinant correct to 3 places.
9. $H = 635.5$mm

1.2 Exercises

1. (a) loc. convergent (b) divergent (c) divergent
3. (a) 0 is locally convergent, 1 is divergent (b) $1/2$ is locally convergent, $3/4$ is divergent
5. (a) For example, $x = x^3 + e^x$, $x = (x - e^x)^{1/3}$, and $x = \ln(x - x^3)$; (b) For example, $x = 9x^2 + 3/x^3$, $x = 1/9 - 1/3x^4$, and $x = (x^5 - 9x^6)/3$
7. $g(x) = \sqrt{(1-x)/2}$ is locally convergent to $1/2$, and $g(x) = -\sqrt{(1-x)/2}$ is locally convergent to -1.
9. $g(x) = (x + A/x^2)/2$ converges to $A^{1/3}$.
11. (a) Substitute and check (b) $|g'(r)| > 1$ for all three fixed points r
13. $g'(r_2) > 1$
17. (a) $x = x - x^3$ implies $x = 0$ (b) If $0 < x_i < 1$, then $x_{i+1} = x_i - x_i^3 = x_i(1 - x_i^2) < x_i$, and $0 < x_{i+1} < x_i < 1$. (c) The bounded monotonic sequence x_i converges to a limit L, which must be a fixed point. Therefore $L = 0$.
19. (a) $c < -2$ (b) $c = -4$
21. The open interval $(-5/4, 5/4)$ of initial guesses converge to the fixed point $1/4$; the two initial guesses $-5/4, 5/4$ lead to $-5/4$.

1.2 Computer Problems

1. (a) 1.76929235 (b) 1.67282170 (c) 1.12998050
3. (a) 1.73205081 (b) 2.23606798

5. fixed point is $r = 0.641714$ and $S = |g'(r)| \approx 0.959$
7. (a) $0 < x_0 < 1$ (b) $1 < x_0 < 2$ (c) $x_0 > 2.2$, for example

1.3 Exercises

1. (a) FE $= 0.01$, BE $= 0.04$ (b) FE $= 0.01$, BE $= 0.0016$ (c) FE $= 0.01$, BE $= 0.000064$ (d) FE $= 0.01$, BE $= 0.342$
3. (a) 2 (b) FE $= 0.0001$, BE $= 5 \times 10^{-9}$
5. BE $= |a|$ FE
7. (b) $(-1)^j (j-1)!(20-j)!$

1.3 Computer Problems

1. (a) $m = 3$ (b) $x_c = \text{FE} = 2.0735 \times 10^{-8}$, BE $= 0$
3. (a) $x_c = \text{FE} = 0.000169$, BE $= 0$ (b) Terminates after 13 steps, $x_c = -0.00006103$
5. Predicted root $= r + \Delta r = 4 + 4^6 10^{-6}/6 = 4.0006826\overline{6}$, actual root $= 4.0006825$

1.4 Exercises

1. (a) $x_1 = 2, x_2 = 18/13$ (b) $x_1 = 1, x_2 = 1$ (c) $x_1 = -1, x_2 = -2/3$
3. (a) $r = -1, e_{i+1} = \frac{5}{2}e_i^2$; $r = 0, e_{i+1} = 2e_i^2$; $r = 1, e_{i+1} = \frac{2}{3}e_i$ (b) $r = -1/2, e_{i+1} = 2e_i^2$; $r = 1, e_{i+1} = \frac{2}{3}e_i$
5. $r = 0$, Newton's Method; $r = 1/2$, Bisection Method
7. No, $2/3$
9. $x_{i+1} = (x_i + A/x_i)/2$
11. $x_{i+1} = (n-1)x_i/n + A/(nx_i^{n-1})$
13. (a) 0.75×10^{-12} (b) 0.5×10^{-18}

1.4 Computer Problems

1. (a) 1.76929235 (b) 1.67282170 (c) 1.12998050
3. (a) $r = -2/3, m = 3$ (b) $r = 1/6, m = 2$
5. $r = 3.2362$ m
7. -1.197624, quadratic conv.; 0, linear conv., $m = 4$; 1.530134, quadratic conv.
9. 0.857143, quadratic conv., $M = 2.414$; 2, linear conv., $m = 3$, $S = 2/3$
11. initial guess $= 1.75$, solution $V = 1.70$ L
13. $3/4$

1.5 Exercises

1. (a) $x_2 = 8/5, x_3 = 1.742268$ (b) $x_2 = 1.578707, x_3 = 1.66016$ (c) $x_2 = 1.092907, x_3 = 1.119357$
3. (a) $x_3 = -1/5, x_4 = -0.11996018$ (b) $x_3 = 1.757713, x_4 = 1.662531$ (c) $x_3 = 1.139481, x_4 = 1.129272$

3. (a) One, $P(x) = 3 + (x + 1)(x - 2)$ (b) None (c) Infinitely many, for example $P(x) = 3 + (x + 1)(x - 2) + (x + 1)(x - 1)(x - 2)(x - 3)$
5. (a) $P(x) = 4 - 2x$ (b) $P(x) = 4 - 2x + A(x + 2)x(x - 1)(x - 3)$ for $A \neq 0$
7. 4
9. (a) $P(x) = 10(x - 1)\cdots(x - 6)/6!$ (b) Same as (a)
11. None
13. (a) 316 (b) 465
15. (a) $\frac{1}{2}n^2 + \frac{3}{2}n - 1$ additions and $n(2n - 2)$ multiplications (b) $2n - 2$ additions and $n - 1$ multiplications

3.1 Computer Problems

1. (a) 4494564854 (b) 4454831984 (c) 4472888288

3.2 Exercises

1. (a) $P_2(x) = \frac{2}{\pi}x - \frac{4}{\pi^2}x(x - \pi/2)$ (b) $P_2(\pi/4) = 3/4$ (c) $\pi^3/128 \approx 0.242$ (d) $|\sqrt{2}/2 - 3/4| \approx 0.043$
3. (a) 7.06×10^{-11} (b) at least 9 decimal places, since $7.06 \times 10^{-11} < 0.5 \times 10^{-9}$
5. Expect errors at $x = 0.35$ to be smaller; approximately $5/21$ the size of the error at $x = 0.55$.

3.2 Computer Problems

1. (a) $P_4(x) = 1.433329 + (x - 0.6)(1.98987 + (x - 0.7)(3.2589 + (x - 0.8)(3.680667 + (x - 0.9)(4.000417))))$ (b) $P_4(0.82) = 1.95891$, $P_4(0.98) = 2.612848$ (c) Upper bound for error at $x = 0.82$ is 0.0000537, actual error is 0.0000234. Upper bound for error at $x = 0.98$ is 0.000217, actual error is 0.000107.
3. -1.952×10^{12} bbl/day. The estimate is nonsensical, due to the Runge phenomenon.

3.3 Exercises

1. (a) $\cos\pi/12, \cos\pi/4, \cos 5\pi/12, \cos 7\pi/12, \cos 3\pi/4, \cos 11\pi/12$
   (b) $2\cos\pi/8, 2\cos 3\pi/8, 2\cos 5\pi/8, 2\cos 7\pi/8$
   (c) $8 + 4\cos\pi/12, 8 + 4\cos\pi/4, 8 + 4\cos 5\pi/12, 8 + 4\cos 7\pi/12, 8 + 4\cos 3\pi/4, 8 + 4\cos 11\pi/12$
   (d) $1/5 + 1/2\cos\pi/10, 1/5 + 1/2\cos 3\pi/10, 1/5, 1/5 + 1/2\cos 7\pi/10, 1/5 + 1/2\cos 9\pi/10$
3. 0.000118, 3 correct digits
5. 0.00521
7. $d = 14$
9. (a) -1 (b) 1 (c) 0 (d) 1 (e) 1 (f) $-1/2$

3.4 Exercises

1. (a) not a cubic spline (b) cubic spline
3. (a) $c = 9/4$, natural (b) $c = 4$, parabolically-terminated and not-a-knot (c) $c = 5/2$, not-a-knot
5. One, $S_1(x) = S_2(x) = x$

(d)

t_i	w_i	error
0.0	1.0000	0.0000
0.1	1.0000	0.0000
0.2	1.0003	0.0001
0.3	1.0022	0.0002
0.4	1.0097	0.0005
0.5	1.0306	0.0012
0.6	1.0785	0.0024
0.7	1.1778	0.0052
0.8	1.3754	0.0124
0.9	1.7711	0.0338
1.0	2.6107	0.1076

(e)

t_i	w_i	error
0.0	1.0000	0.0000
0.1	1.0907	0.0007
0.2	1.1686	0.0010
0.3	1.2375	0.0011
0.4	1.2995	0.0011
0.5	1.3561	0.0011
0.6	1.4083	0.0011
0.7	1.4570	0.0011
0.8	1.5026	0.0011
0.9	1.5456	0.0010
1.0	1.5864	0.0010

(f)

t_i	w_i	error
0.0	1.0000	0.0000
0.1	1.0000	0.0000
0.2	1.0003	0.0000
0.3	1.0019	0.0001
0.4	1.0062	0.0002
0.5	1.0151	0.0003
0.6	1.0311	0.0003
0.7	1.0564	0.0003
0.8	1.0931	0.0003
0.9	1.1426	0.0001
1.0	1.2051	0.0001

6.6 Exercises

1. (a) $w = [0, 0.0833, 0.2778, 0.6204, 1.1605]$, error $= 0.4422$
   (b) $w = [0, 0.0500, 0.1400, 0.2620, 0.4096]$, error $= 0.0417$
   (c) $w = [0, 0.1667, 0.4444, 0.7963, 1.1975]$, error $= 0.0622$

6.6 Computer Problems

1. (a) $y = 1$, Euler step size ≤ 1.8 (b) $y = 1$, Euler step size $\leq 1/3$

6.7 Exercises

1. (a) $w = [1.0000, 1.0313, 1.1250, 1.2813, 1.5000]$, error $= 0$
   (b) $w = [1.0000, 1.0078, 1.0314, 1.1203, 1.3243]$, error $= 0.0713$
   (c) $w = [1.0000, 1.7188, 3.0801, 6.0081, 12.7386]$, error $= 7.3469$
   (d) $w = [1.0000, 1.0024, 1.0098, 1.1257, 1.7540]$, error $= 0.9642$
   (e) $w = [1.0000, 1.2050, 1.3383, 1.4616, 1.5673]$, error $= 0.0201$
   (f) $w = [1.0000, 1.0020, 1.0078, 1.0520, 1.1796]$, error $= 0.0255$
3. $w_{i+1} = -4w_i + 5w_{i-1} + h[4f_i + 2f_{i-1}]$; No.
7. (a) $0 < a_1 < 2$ (b) $a_1 = 0$
9. (a) second order unstable (b) second order strongly stable (c) third order strongly stable (d) third order unstable (e) third order unstable
11. For example, $a_1 = 0, a_2 = 1, b_0 = 1, b_1 = -1, b_2 = 2$.
13. (a) $a_1 + a_2 + a_3 = 1, -a_2 - 2a_3 + b_1 + b_2 + b_3 = 1, a_2 + 4a_3 - 2b_2 - 4b_3 = 1, -a_2 - 8a_3 + 3b_2 + 12b_3 = 1$ (c) $P(x) = x^3 - x^2$ has double root at 0, simple root at 1. (d) $w_{i+1} = w_{i-1} + h[\frac{7}{3}f_i - \frac{2}{3}f_{i-1} + \frac{1}{3}f_{i-2}]$
15. (a) $a_1 + a_2 + a_3 = 1, -a_2 - 2a_3 + b_0 + b_1 + b_2 + b_3 = 1, a_2 + 4a_3 + 2b_0 - 2b_2 - 4b_3 = 1, -a_2 - 8a_3 + 3b_0 + 3b_2 + 12b_3 = 1, a_2 + 16a_3 + 4b_0 - 4b_2 - 32b_3 = 1$ (c) $P(x) = x^3 - x^2 = x^2(x - 1)$ has simple root at 1.

3. (a) 34 bits needed, $34/11 = 3.09$ bits/symbol $> 3.03 =$ Shannon inf. (b) 73 bits needed, $73/21 = 3.48$ bits/symbol $> 3.42 =$ Shannon inf. (c) 108 bits needed, $108/35 = 3.09$ bits/symbol $> 3.04 =$ Shannon inf.

11.4 Exercises

1. (a) $[-12b - 2c, 2b - 12c]$ (b) $[-3b - c, b - 3c]$ (c) $[-8b + 5c, -5b - 8c]$
3. (a) $+101.$, error $= 0$ (b) $+101.$, error $= 1/15$ (c) $+011.$, error $= 1/35$
5. (a) $+110000.$, error $= 1/170$ (b) $-101101.$, error $= 1/85$ (c) $+1011100.$, error $= 7/510$ (d) $+1100100.$, error ≈ 0.0043
7. (a) $\frac{1}{2}(w_2 + w_3) = [-1.2246, 0.9184] \approx [-1, 1]$ (b) $\frac{1}{2}(w_2 + w_3) = [2.1539, -0.9293] \approx [2, -1]$ (c) $\frac{1}{2}(w_2 + w_3) = [-1.7844, -3.0832] \approx [-2, -3]$
9. $c_{5n} = -c_{n-1}, c_{6n} = -c_0$

CHAPTER 12

12.1 Exercises

1. (a) $P(\lambda) = (\lambda - 5)(\lambda - 2)$, 2 and $[1, 1]$, 5 and $[1, -1]$ (b) $P(\lambda) = (\lambda + 2)(\lambda - 2)$, -2 and $[1, -1]$, 2 and $[1, 1]$ (c) $P(\lambda) = (\lambda - 3)(\lambda + 2)$, 3 and $[-3, 4]$, -2 and $[4, 3]$ (d) $P(\lambda) = (\lambda - 100)(\lambda - 200)$, 200 and $[-3, 4]$, 100 and $[4, 3]$
3. (a) $P(\lambda) = -(\lambda - 1)(\lambda - 2)(\lambda - 3)$, 3 and $[0, 1, 0]$, 2 and $[1, 2, 1]$, 1 and $[1, 0, 0]$
   (b) $P(\lambda) = -\lambda(\lambda - 1)(\lambda - 2)$, 2 and $[-1, 2, 3]$, 1 and $[1, 1, 0]$, 0 and $[1, -2, 3]$
   (c) $P(\lambda) = -\lambda(\lambda - 1)(\lambda + 1)$, 1 and $[1, -2, -3]$, 0 and $[1, -2, 3]$, -1 and $[1, 1, 0]$
5. (a) $\lambda = 4, S = 3/4$ (b) $\lambda = -4, S = 3/4$ (c) $\lambda = 4, S = 1/2$ (d) $\lambda = 10, S = 9/10$
7. (a) $\lambda = 1, S = 1/3$ (b) $\lambda = 1, S = 1/3$ (c) $\lambda = -1, S = 1/2$ (d) $\lambda = 9, S = 3/4$
9. (a) 5 and $[1, 2]$, -1 and $[-1, 1]$ (b) $x_1 = [1, 4]$, RQ = 1; $x_2 = [9/\sqrt{17}, 16/\sqrt{17}]$, RQ = 4.29; $x_3 = [2.2334, 4.5758]$, RQ = 5.08 (c) IPI converges to $\lambda = -1$. (d) IPI converges to $\lambda = 5$.
11. (a) 7 (b) 5 (c) $S = 6/7, S = 1/2$; IPI with $s = 4$ is faster.

12.1 Computer Problems

1. (a) converges to 4 and $[1, 1, -1]$ (b) converges to -4 and $[1, 1, -1]$ (c) converges to 4 and $[1, 1, -1]$ (d) converges to 10 and $[1, 1, -1]$
3. (a) $\lambda = 4$ (b) $\lambda = 3$ (c) $\lambda = 2$ (d) $\lambda = 9$

12.2 Exercises

1. (a) $\begin{bmatrix} 1 & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\sqrt{2} & \frac{1}{2} & \frac{1}{2} \\ 0 & \frac{1}{2} & \frac{1}{2} \end{bmatrix}$ (b) $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & -1 & 0 \end{bmatrix}$ (c) $\begin{bmatrix} 2 & -\frac{4}{5} & -\frac{3}{5} \\ -5 & \frac{37}{25} & -\frac{16}{25} \\ 0 & \frac{9}{25} & \frac{13}{25} \end{bmatrix}$

   (d) $\begin{bmatrix} 1 & -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ -\sqrt{8} & \frac{5}{2} & \frac{3}{2} \\ 0 & \frac{3}{2} & \frac{1}{2} \end{bmatrix}$

3. (a) Best line $y = 3.3028x$; projections are $\begin{bmatrix} 1.1934 \\ 3.9415 \end{bmatrix}, \begin{bmatrix} 1.4707 \\ 4.8575 \end{bmatrix}, \begin{bmatrix} 1.2774 \\ 4.2188 \end{bmatrix}$.

   (b) Best line $y = 0.3620x$; projections are $\begin{bmatrix} 1.7682 \\ 0.6402 \end{bmatrix}, \begin{bmatrix} 3.8565 \\ 1.3963 \end{bmatrix}, \begin{bmatrix} 3.2925 \\ 1.1921 \end{bmatrix}$.

   (c) Best line $(x(t), y(t), z(t)) = [0.3105, 0.3416, 0.8902]t$; projections are $\begin{bmatrix} 1.3702 \\ 1.5527 \\ 4.0463 \end{bmatrix}, \begin{bmatrix} 1.8325 \\ 2.0764 \\ 5.4111 \end{bmatrix}, \begin{bmatrix} 1.8949 \\ 2.1471 \\ 5.5954 \end{bmatrix}$, $\begin{bmatrix} 0.9989 \\ 1.1319 \\ 2.9498 \end{bmatrix}$.

5. (a) $\begin{bmatrix} 3 & 0 \\ 4 & 0 \end{bmatrix} = \begin{bmatrix} -0.6 & -0.8 \\ -0.8 & 0.6 \end{bmatrix} \begin{bmatrix} 5 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$

   (b) $\begin{bmatrix} 6 & -2 \\ 8 & \frac{3}{2} \end{bmatrix} = \begin{bmatrix} 0.6 & -0.8 \\ 0.8 & 0.6 \end{bmatrix} \begin{bmatrix} 10 & 0 \\ 0 & \frac{5}{2} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

   (c) $\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$

   (d) $\begin{bmatrix} -4 & -12 \\ 12 & 11 \end{bmatrix} = \begin{bmatrix} -0.6 & -0.8 \\ 0.8 & -0.6 \end{bmatrix} \begin{bmatrix} 20 & 0 \\ 0 & 5 \end{bmatrix} \begin{bmatrix} 0.6 & -0.8 \\ 0.8 & 0.6 \end{bmatrix}$

   (e) $\begin{bmatrix} 0 & -2 \\ -1 & 0 \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$

CHAPTER 13

13.1 Exercises

1. (a) $(0, 2)$ (b) $(0, 0)$ (c) $(-1/2, -3/8)$ (d) $(1, 1)$

13.1 Computer Problems

1. (a) $1/2$ (b) $-2, 1$ (c) 0.47033 (d) 1.43791
3. (a), (b): $(0.358555, 2.788973)$
5. $(1.20881759, 1.20881759)$, about 8 correct places
7. $(1, 1)$

13.2 Computer Problems

1. Minimum is $(1.2088176, 1.2088176)$. Different initial conditions will yield answers that differ by about $\epsilon^{1/2}$.
3. $(1, 1)$. Newton's Method will be accurate to machine precision, since it is finding a simple root. Steepest Descent will have error of size $\approx \epsilon^{1/2}$.
5. (a) $(1.132638, -0.465972), (-0.465972, 1.132638)$ (b) $\pm(0.6763, 0.6763)$