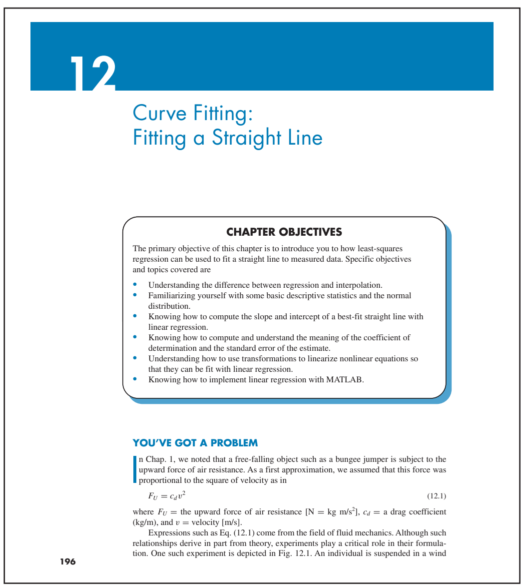
GUIDED TOUR

Chapter Objectives *Chapter Objectives* begin each chapter. The objectives provide students with the function of each chapter as well as the specific topics covered in each chapter. The objectives enable students to set tangible goals before they begin each chapter.


12

Curve Fitting:
Fitting a Straight Line

CHAPTER OBJECTIVES

The primary objective of this chapter is to introduce you to how least-squares regression can be used to fit a straight line to measured data. Specific objectives and topics covered are

- Understanding the difference between regression and interpolation.
- Familiarizing yourself with some basic descriptive statistics and the normal distribution.
- Knowing how to compute the slope and intercept of a best-fit straight line with linear regression.
- Knowing how to compute and understand the meaning of the coefficient of determination and the standard error of the estimate.
- Understanding how to use transformations to linearize nonlinear equations so that they can be fit with linear regression.
- Knowing how to implement linear regression with MATLAB.

YOU'VE GOT A PROBLEM

In Chap. 1, we noted that a free-falling object such as a bungee jumper is subject to the upward force of air resistance. As a first approximation, we assumed that this force was proportional to the square of velocity as in

$$F_U = c_d v^2 \qquad (12.1)$$

where F_U = the upward force of air resistance [N = kg m/s²], c_d = a drag coefficient (kg/m), and v = velocity [m/s].

Expressions such as Eq. (12.1) come from the field of fluid mechanics. Although such relationships derive in part from theory, experiments play a critical role in their formulation. One such experiment is depicted in Fig. 12.1. An individual is suspended in a wind

You've Got a Problem A section entitled *You've Got a Problem* can be found on the first page of most chapters. Here Chapra poses a real-life problem that requires the type of numerical solution technique that is the subject of the chapter. The intent is to introduce the student to the topic via a tangible example rather than through abstract mathematics. After an exposition of the numerical methods, the problem is then revisited in order to demonstrate how the learned material provides the means to solve the problem.

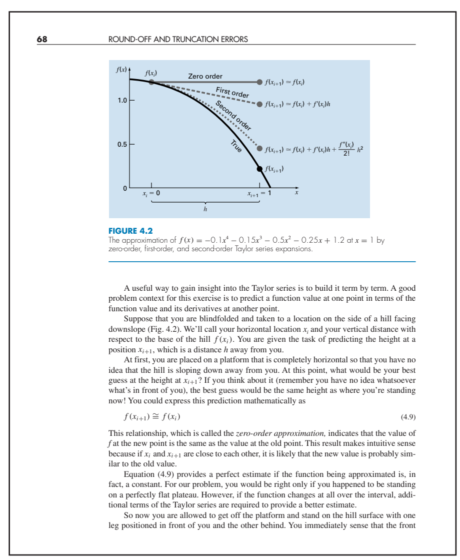


FIGURE 4.2
The approximation of $f(x) = -0.1x^4 - 0.15x^3 - 0.5x^2 - 0.25x + 1.2$ at $x = 1$ by zero-order, first-order, and second-order Taylor series expansions.

A useful way to gain insight into the Taylor series is to build it term by term. A good problem context for this exercise is to predict a function value at one point in terms of the function value and its derivatives at another point.

Suppose that you are blindfolded and taken to a location on the side of a hill facing downslope (Fig. 4.2). We'll call your horizontal location x_i and your vertical distance with respect to the base of the hill $f(x_i)$. You are given the task of predicting the height at a position x_{i+1}, which is a distance h away from you.

At first, you are placed on a platform that is completely horizontal so that you have no idea that the hill is sloping down away from you. At this point, what would be your best guess at the height at x_{i+1}? If you think about it (remember you have no idea whatsoever what's in front of you), the best guess would be the same height as where you're standing now! You could express this prediction mathematically as

$$f(x_{i+1}) \cong f(x_i) \qquad (4.9)$$

This relationship, which is called the *zero-order approximation,* indicates that the value of f at the new point is the same as the value at the old point. This result makes intuitive sense because if x_i and x_{i+1} are close to each other, it is likely that the new value is probably similar to the old value.

Equation (4.9) provides a perfect estimate if the function being approximated is, in fact, a constant. For our problem, you would be right only if you happened to be standing on a perfectly flat plateau. However, if the function changes at all over the interval, additional terms of the Taylor series are required to provide a better estimate.

So now you are allowed to get off the platform and stand on the hill surface with one leg positioned in front of you and the other behind. You immediately sense that the front

Theory Presented as it Informs Key Concepts The text is intended for Numerical Methods users, not developers. Therefore, theory is not included for "theory's sake," for example no proofs. Theory is included as it informs key concepts such as the Taylor Series, convergence, condition, etc. Hence, the student is shown how the theory connects with practical issues in problem solving.

Illustrations and Tables Illustrations and tables are clear and accurate in order to help students better visualize the important concepts presented in the text.

7.1WHAT ARE LINEAR ALGEBRAIC EQUATIONS?

Linear algebraic equations are of the general form,

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ \vdots \qquad\qquad & \quad \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n \end{aligned} \tag{7.2}$$

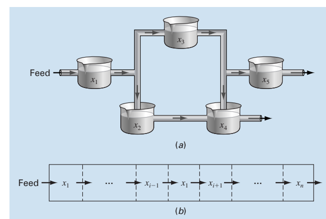
where the a's are constant coefficients, the b's are constants, the x's are unknowns, and n is the number of equations. All other algebraic equations are nonlinear.

7.1.1 Linear Algebraic Equations and Engineering Practice

Many of the fundamental equations of engineering and science are based on conservation laws. Some familiar quantities that conform to such laws are mass, energy, and momentum. In mathematical terms, these principles lead to balance or continuity equations that relate system behavior as represented by the levels or response of the quantity being modeled to the properties or characteristics of the system and the external stimuli or forcing functions acting on the system.

As an example, the principle of mass conservation can be used to formulate a model for a series of chemical reactors (Fig. 7.3*a*). For this case, the quantity being modeled is the mass of the chemical in each reactor. The system properties are the reaction characteristics

FIGURE 7.3
Two types of systems that can be modeled using linear algebraic equations: (*a*) lumped variable system that involves coupled finite components and (*b*) distributed variable system that involves a continuum.



Introductory MATLAB Material The text includes two introductory chapters on how to use MATLAB. Chapter 2 shows students how to perform computations and create graphs in MATLAB's standard command mode. Chapter 3 provides a primer on developing numerical programs via MATLAB M-file functions. Thus, the text provides students with the means to develop their own numerical algorithms as well as to tap into MATLAB's powerful built-in routines.

TABLE 2.1 Specifiers for colors, symbols, and line types.

Colors		Symbols		Line Types	
Blue	b	Point	.	Solid	–
Green	g	Circle	o	Dotted	:
Red	r	X-mark	x	Dashdot	-.
Cyan	c	Plus	+	Dashed	--
Magenta	m	Star	*		
Yellow	y	Square	s		
Black	k	Diamond	d		
		Triangle(down)	v		
		Triangle(up)	^		
		Triangle(left)	<		
		Triangle(right)	>		
		Pentagram	p		
		Hexagram	h		

Table 2.1 lists the available specifiers. For example, if you want to use open circles enter

```
>> plot (t, v, 'o')
```

MATLAB allows you to display more than one data set on the same plot. For example, if you want to connect each data marker with a straight line you could type

```
>> plot (t, v, t, v, 'o')
```

There are other features of graphics that are useful—for example, plotting objects instead of lines, families of curves plots, plotting on the complex plane, multiple graphs windows, log-log or semilog plots, three-dimensional mesh plots, and contour plots. As described next, a variety of resources are available to learn about these as well as other MATLAB capabilities.

2.6 OTHER RESOURCES

The foregoing was designed to focus on those features of MATLAB that we will be using in the remainder of this book. As such, it is obviously not a comprehensive overview of all of MATLAB's capabilities. If you are interested in learning more, you should consult one of the excellent books devoted to MATLAB (e.g., Palm, 2004).

Further, the package itself includes an extensive Help facility that can be accessed by clicking on the Help menu in the command window. This will provide you with a number of different options for exploring and searching through MATLAB's Help material. In addition, it provides access to a number of instructive demos.

As described in this chapter, help is also available in interactive mode by typing the `help` command followed by the name of a command or function.

If you do not know the name, you can use the `lookfor` command to search the MATLAB Help files for occurrences of text. For example, suppose that you want to find all the commands and functions that relate to logarithms, you could enter

```
>> lookfor logarithm
```

and MATLAB will display all references that include the word `logarithm`.

Algorithms Presented Using MATLAB M-files Instead of using pseudocode, this book presents algorithms as well-structured MATLAB M-files. Aside from being useful computer programs, these provide students with models for their own M-files that they will develop as homework exercises.

5.4 BISECTION 95

specify an absolute error. For these cases, bisection along with Eq. (5.6) can provide a useful root location algorithm.

5.4.1 MATLAB M-file: `bisection`

An M-file to implement bisection is displayed in Fig. 5.7. It is passed the function (`func`) along with lower (`xl`) and upper (`xu`) guesses. In addition an optional stopping criterion

FIGURE 5.7
An M-file to implement the bisection method.

```
function root = bisection(func,xl,xu,es,maxit)
% bisection(func,xl,xu,es,maxit):
%   uses bisection method to find the root of a function
% input:
%   func = name of function
%   xl, xu = lower and upper guesses
%   es = (optional) stopping criterion (%)
%   maxit = (optional) maximum allowable iterations
% output:
%   root = real root

if func(xl)*func(xu)>0  %if guesses do not bracket a sign
  error('no bracket')   %change, display an error message
  return                %and terminate
end
% if necessary, assign default values
if nargin<5, maxit = 50; end     %if maxit blank set to 50
if nargin<4, es = 0.001; end     %if es blank set to 0.001

% bisection
iter = 0;
xr = xl;
while (1)
  xrold = xr;
  xr = (xl + xu)/2;
  iter = iter + 1;
  if xr ~= 0, ea = abs((xr - xrold)/xr) * 100; end
  test = func(xl)*func(xr);
  if test < 0
    xu = xr;
  elseif test > 0
    xl = xr;
  else
    ea = 0;
  end
  if ea <= es | iter >= maxit, break, end
end
root = xr;
```

5.5 FALSE POSITION 97

EXAMPLE 5.5 The False-Position Method

Problem Statement. Use false position to solve the same problem approached graphically and with bisection in Examples 5.1 and 5.3.

Solution. As in Example 5.3, initiate the computation with guesses of $x_l = 50$ and $x_u = 200$.

First iteration:

$$x_l = 50 \qquad f(x_l) = -4.579387$$
$$x_u = 200 \qquad f(x_u) = 0.860291$$
$$x_r = 200 - \frac{0.860291(50 - 200)}{-4.579387 - 0.860291} = 176.2773$$

which has a true relative error of 23.5%.

Second iteration:

$$f(x_l)f(x_r) = -2.592732$$

Therefore, the root lies in the first subinterval, and x_r becomes the upper limit for the next iteration, $x_u = 176.2773$.

$$x_l = 50 \qquad f(x_l) = -4.579387$$
$$x_u = 176.2773 \qquad f(x_u) = 0.566174$$
$$x_r = 176.2773 - \frac{0.566174(50 - 176.2773)}{-4.579387 - 0.566174} = 162.3828$$

which has true and approximate relative errors of 13.76% and 8.56%, respectively. Additional iterations can be performed to refine the estimates of the root.

Although false position often performs better than bisection, there are other cases where it does not. As in the following example, there are certain cases where bisection yields superior results.

EXAMPLE 5.6 A Case Where Bisection Is Preferable to False Position

Problem Statement. Use bisection and false position to locate the root of

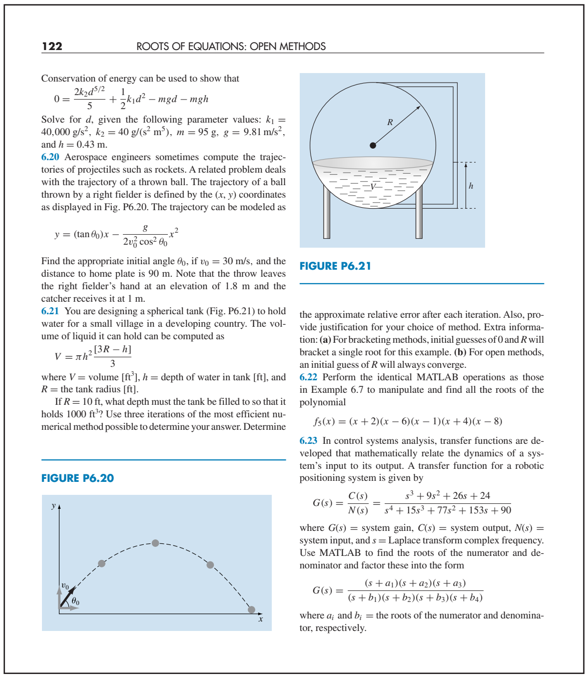
$$f(x) = x^{10} - 1$$

between $x = 0$ and 1.3.

Solution. Using bisection, the results can be summarized as

Iteration	x_l	x_u	x_r	ε_a (%)	ε_t (%)
1	0	1.3	0.65	100.0	35
2	0.65	1.3	0.975	33.3	2.5
3	0.975	1.3	1.1375	14.3	13.8
4	0.975	1.1375	1.05625	7.7	5.6
5	0.975	1.05625	1.015625	4.0	1.6

Worked Examples Each example begins with a problem statement and ends with a solution. The solution is laid out in detail so that students can clearly follow the steps in the numerical computation.



122 ROOTS OF EQUATIONS: OPEN METHODS

Conservation of energy can be used to show that

$$0 = \frac{2k_2 d^{5/2}}{5} + \frac{1}{2}k_1 d^2 - mgd - mgh$$

Solve for d, given the following parameter values: $k_1 = 40{,}000\ \text{g/s}^2$, $k_2 = 40\ \text{g/(s}^2\ \text{m}^5)$, $m = 95$ g, $g = 9.81\ \text{m/s}^2$, and $h = 0.43$ m.

6.20 Aerospace engineers sometimes compute the trajectories of projectiles such as rockets. A related problem deals with the trajectory of a thrown ball. The trajectory of a ball thrown by a right fielder is defined by the (x, y) coordinates as displayed in Fig. P6.20. The trajectory can be modeled as

$$y = (\tan\theta_0)x - \frac{g}{2v_0^2\cos^2\theta_0}x^2$$

Find the appropriate initial angle θ_0, if $v_0 = 30$ m/s, and the distance to home plate is 90 m. Note that the throw leaves the right fielder's hand at an elevation of 1.8 m and the catcher receives it at 1 m.

6.21 You are designing a spherical tank (Fig. P6.21) to hold water for a small village in a developing country. The volume of liquid it can hold can be computed as

$$V = \pi h^2 \frac{[3R - h]}{3}$$

where V = volume [ft^3], h = depth of water in tank [ft], and R = the tank radius [ft].

If $R = 10$ ft, what depth must the tank be filled to so that it holds 1000 ft^3? Use three iterations of the most efficient numerical method possible to determine your answer. Determine the approximate relative error after each iteration. Also, provide justification for your choice of method. Extra information: **(a)** For bracketing methods, initial guesses of 0 and R will bracket a single root for this example. **(b)** For open methods, an initial guess of R will always converge.

FIGURE P6.20

FIGURE P6.21

6.22 Perform the identical MATLAB operations as those in Example 6.7 to manipulate and find all the roots of the polynomial

$$f_5(x) = (x + 2)(x - 6)(x - 1)(x + 4)(x - 8)$$

6.23 In control systems analysis, transfer functions are developed that mathematically relate the dynamics of a system's input to its output. A transfer function for a robotic positioning system is given by

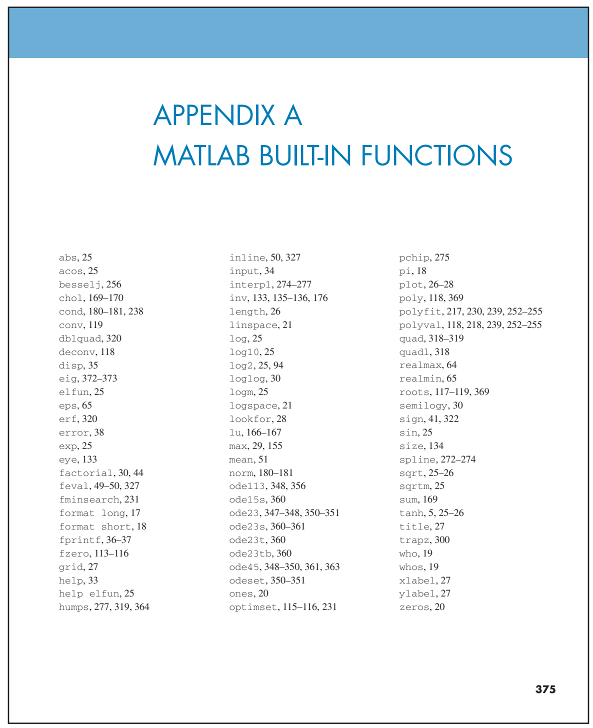
$$G(s) = \frac{C(s)}{N(s)} = \frac{s^3 + 9s^2 + 26s + 24}{s^4 + 15s^3 + 77s^2 + 153s + 90}$$

where $G(s)$ = system gain, $C(s)$ = system output, $N(s)$ = system input, and s = Laplace transform complex frequency. Use MATLAB to find the roots of the numerator and denominator and factor these into the form

$$G(s) = \frac{(s + a_1)(s + a_2)(s + a_3)}{(s + b_1)(s + b_2)(s + b_3)(s + b_4)}$$

where a_i and b_i = the roots of the numerator and denominator, respectively.

Problem Sets The text includes a wide-variety of problems. Many are drawn from engineering and scientific disciplines. Others are used to illustrate numerical techniques and theoretical concepts. Problems include those that can be solved with a pocket calculator as well as others that require computer solution with MATLAB.

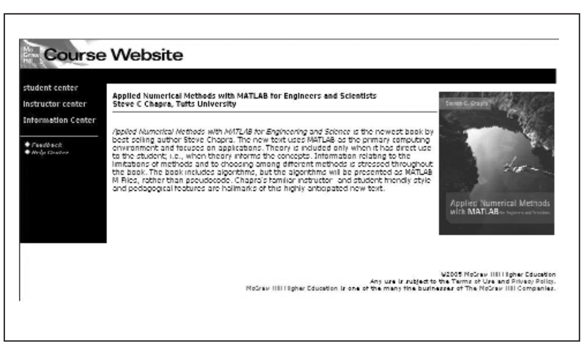


APPENDIX A

MATLAB BUILT-IN FUNCTIONS

375

Useful Indexes Appendix A contains MATLAB commands and Appendix B contains M-file functions.



Supplements A text Web site is available at http://www.mhhe.com/chapra. Resources include PowerPoint slides of text figures and chapter objectives, M-files and additional MATLAB resources. Available to instructors only, the detailed solutions for all text problems will be delivered via CD-Rom, in our new, electronic, Complete Online Solution Manual Organization System. COSMOS is a database management tool geared toward assembling homework assignments, tests and quizzes.