The McGraw-Hill Companies

PowerPoint to accompany

Introduction to MATLAB for Engineers , Third Edition

Chapter 6
Model Building and Regression

Copyright © 2010. The McGraw-Hill Companies, Inc.

Using the Linear, Power, and Exponential Functions to Describe data.

Each function gives a straight line when plotted using a specific set of axes:

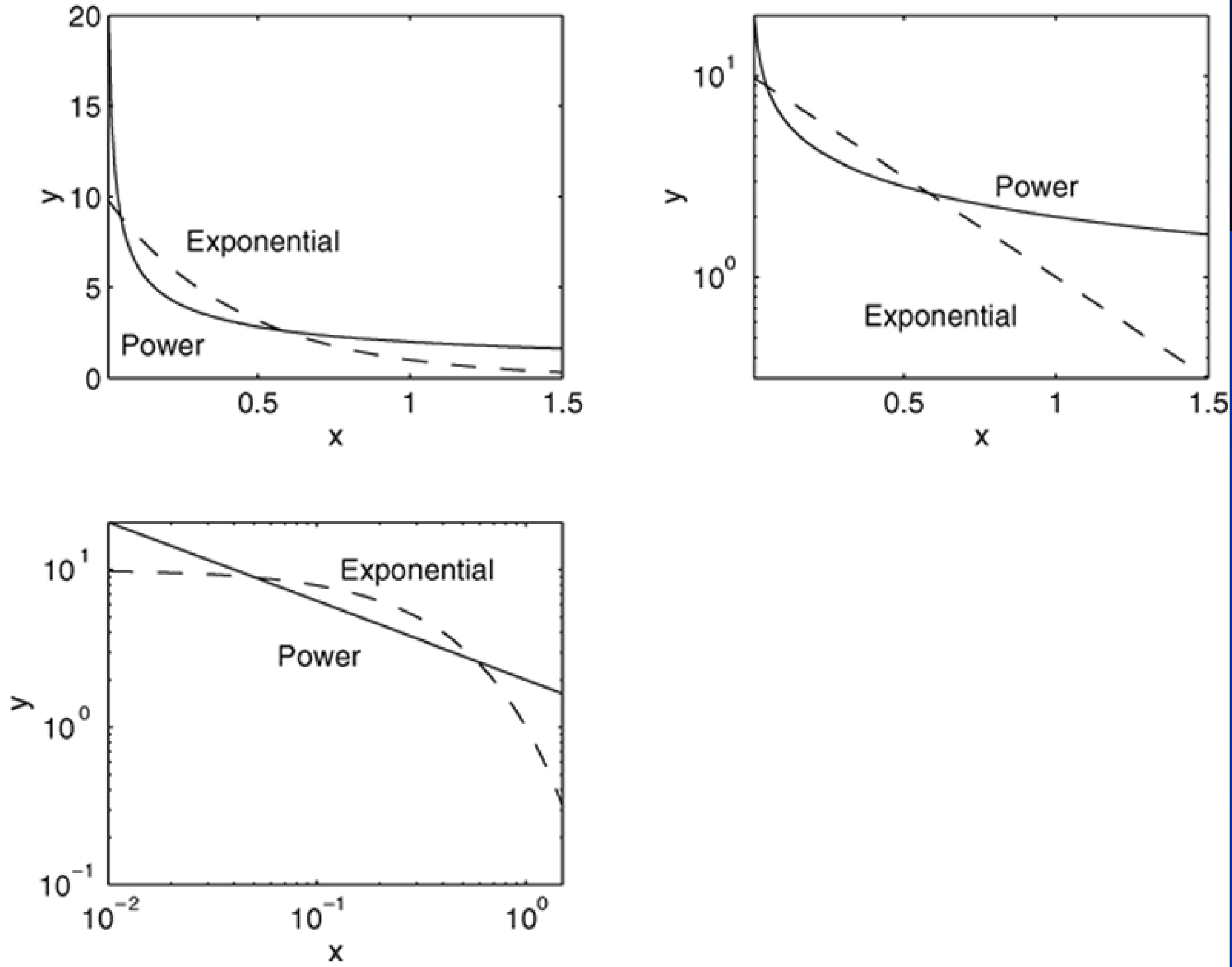
1. The linear function $y = mx + b$ gives a straight line when plotted on rectilinear axes. Its slope is m and its intercept is b.

2. The power function $y = bx^m$ gives a straight line when plotted on log-log axes.

3. The exponential function $y = b(10)^{mx}$ and its equivalent form $y = be^{mx}$ give a straight line when plotted on a semilog plot whose y-axis is logarithmic.

More? See pages 264-265.

Function Discovery. The power function $y = 2x^{-0.5}$ and the exponential function $y = 10^{1-x}$ plotted on linear, semi-log, and log-log axes..

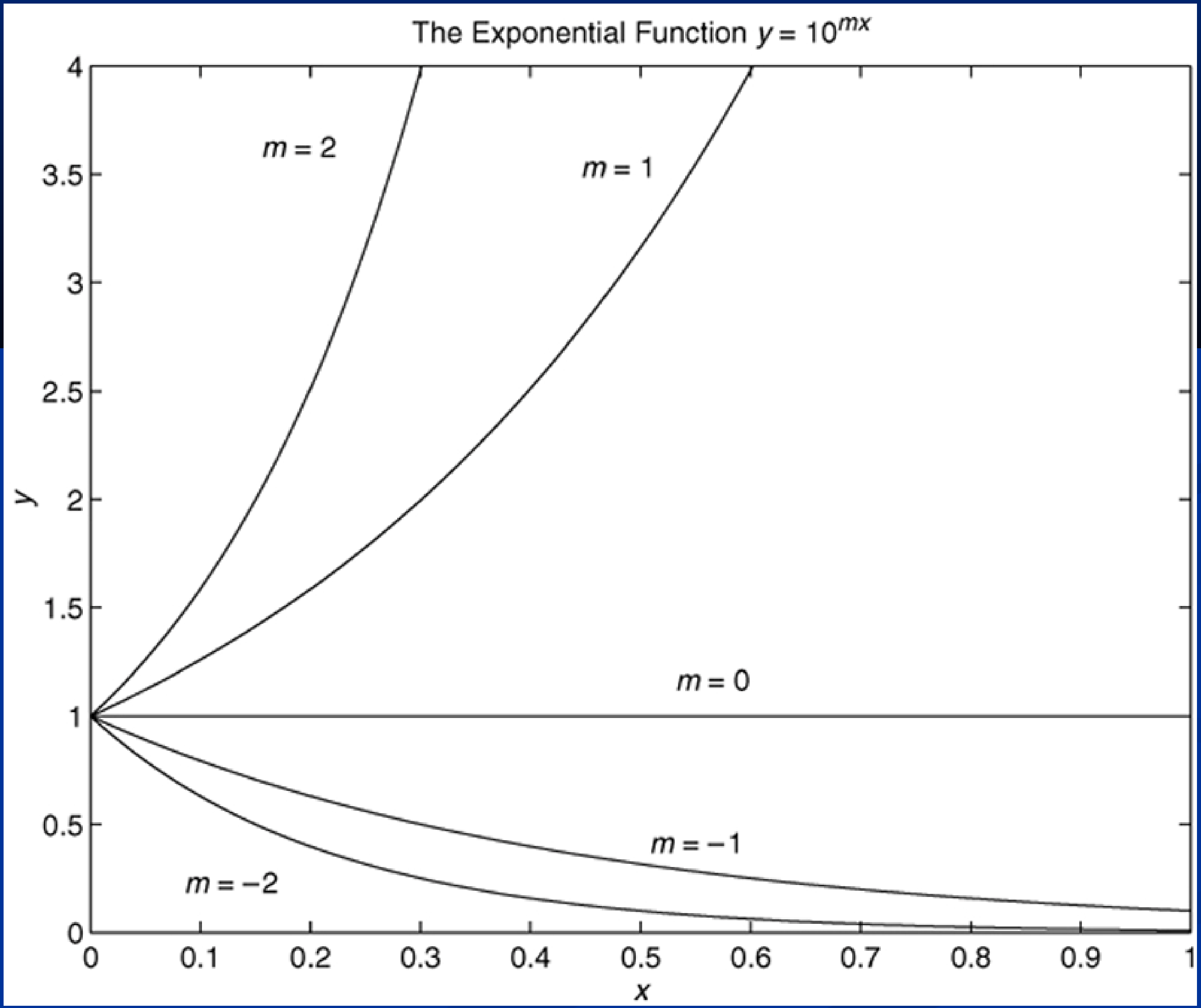


Steps for Function Discovery

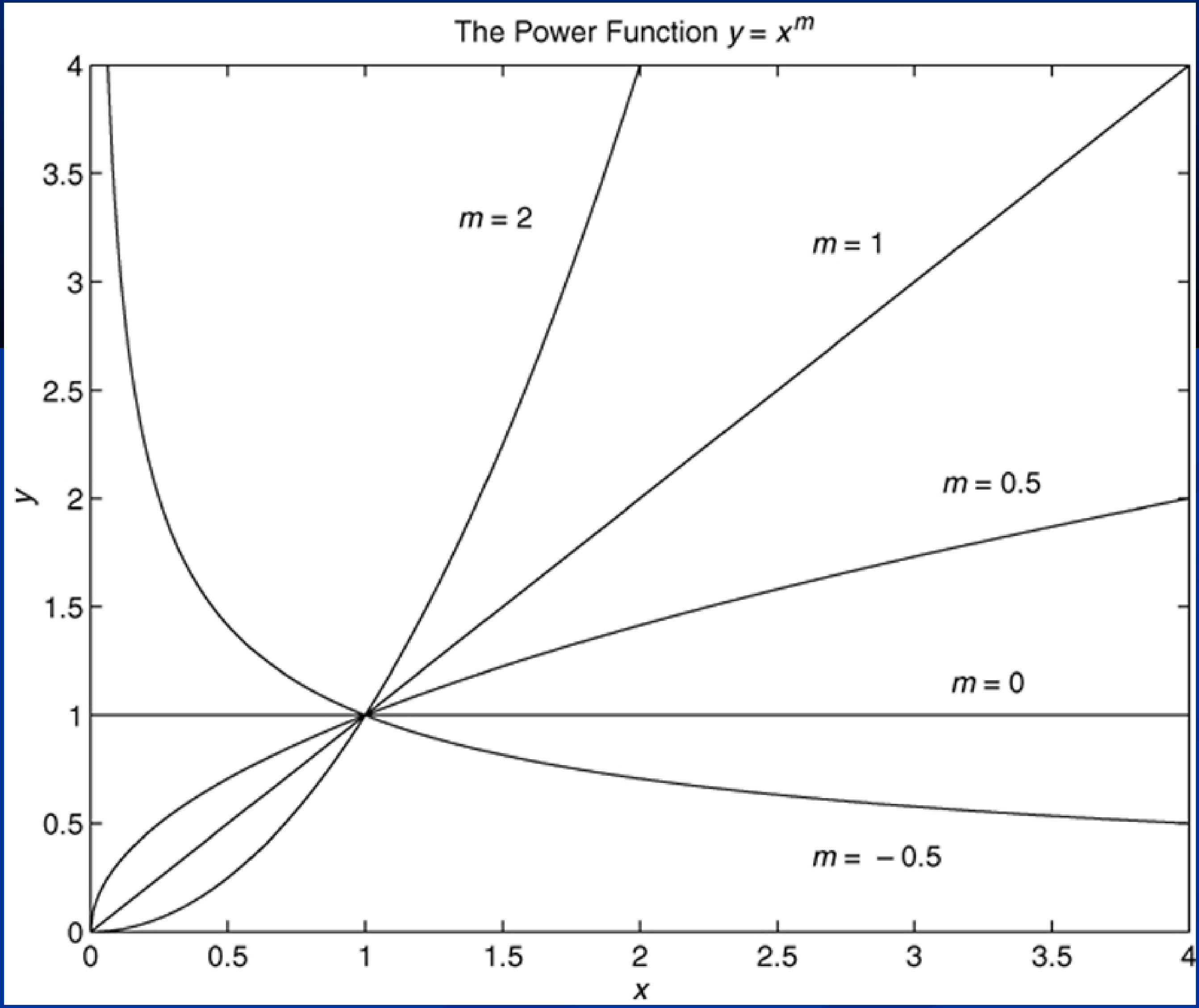
1. Examine the data near the origin. The exponential function can never pass through the origin (unless of course $b = 0$, which is a trivial case). (See Figure 6.1–1 for examples with $b = 1$.)

The linear function can pass through the origin only if $b = 0$. The power function can pass through the origin but only if $m > 0$. (See Figure 6.1–2 for examples with $b = 1$.)

Examples of exponential functions. Figure 6.1–1



Examples of power functions. Figure 6.1–2



Steps for Function Discovery (continued)

2. Plot the data using rectilinear scales. If it forms a straight line, then it can be represented by the linear function and you are finished. Otherwise, if you have data at $x = 0$, then

a. If $y(0) = 0$, try the power function.

b. If $y(0) \neq 0$, try the exponential function.

If data is not given for $x = 0$, proceed to step 3.

(continued…)

Steps for Function Discovery (continued)

3. If you suspect a power function, plot the data using log-log scales. Only a power function will form a straight line on a log-log plot. If you suspect an exponential function, plot the data using the semilog scales. Only an exponential function will form a straight line on a semilog plot.

(continued…)

Steps for Function Discovery (continued)

4. In function discovery applications, we use the log-log and semilog plots *only* to identify the function type, but not to find the coefficients *b* and *m*. The reason is that it is difficult to interpolate on log scales.

The `polyfit` function. Table 6.1–1

Command	Description
`p = polyfit(x,y,n)`	Fits a polynomial of degree *n* to data described by the vectors *x* and *y*, where *x* is the independent variable. Returns a row vector p of length *n* + 1 that contains the polynomial coefficients in order of descending powers.

Using the `polyfit` Function to Fit Equations to Data.

Syntax: `p = polyfit(x,y,n)`

where x and y contain the data, n is the order of the polynomial to be fitted, and p is the vector of polynomial coefficients.

The linear function: $y = mx + b$. In this case the variables w and z in the polynomial $w = p_1 z + p_2$ are the original data variables x and y, and we can find the linear function that fits the data by typing `p = polyfit(x,y,1)`. The first element p_1 of the vector p will be m, and the second element p_2 will be b.

The power function: $y = bx^m$. In this case

$$\log_{10} y = m \log_{10} x + \log_{10} b$$

which has the form

$$w = p_1 z + p_2$$

where the polynomial variables w and z are related to the original data variables x and y by $w = \log_{10} y$ and $z = \log_{10} x$. Thus we can find the power function that fits the data by typing

```
p = polyfit(log10(x),log10(y),1)
```

The first element p_1 of the vector p will be m, and the second element p_2 will be $\log_{10} b$. We can find b from $b = 10^{p_2}$.

The exponential function: $y = b(10)^{mx}$. In this case

$$\log_{10} y = mx + \log_{10} b$$

which has the form

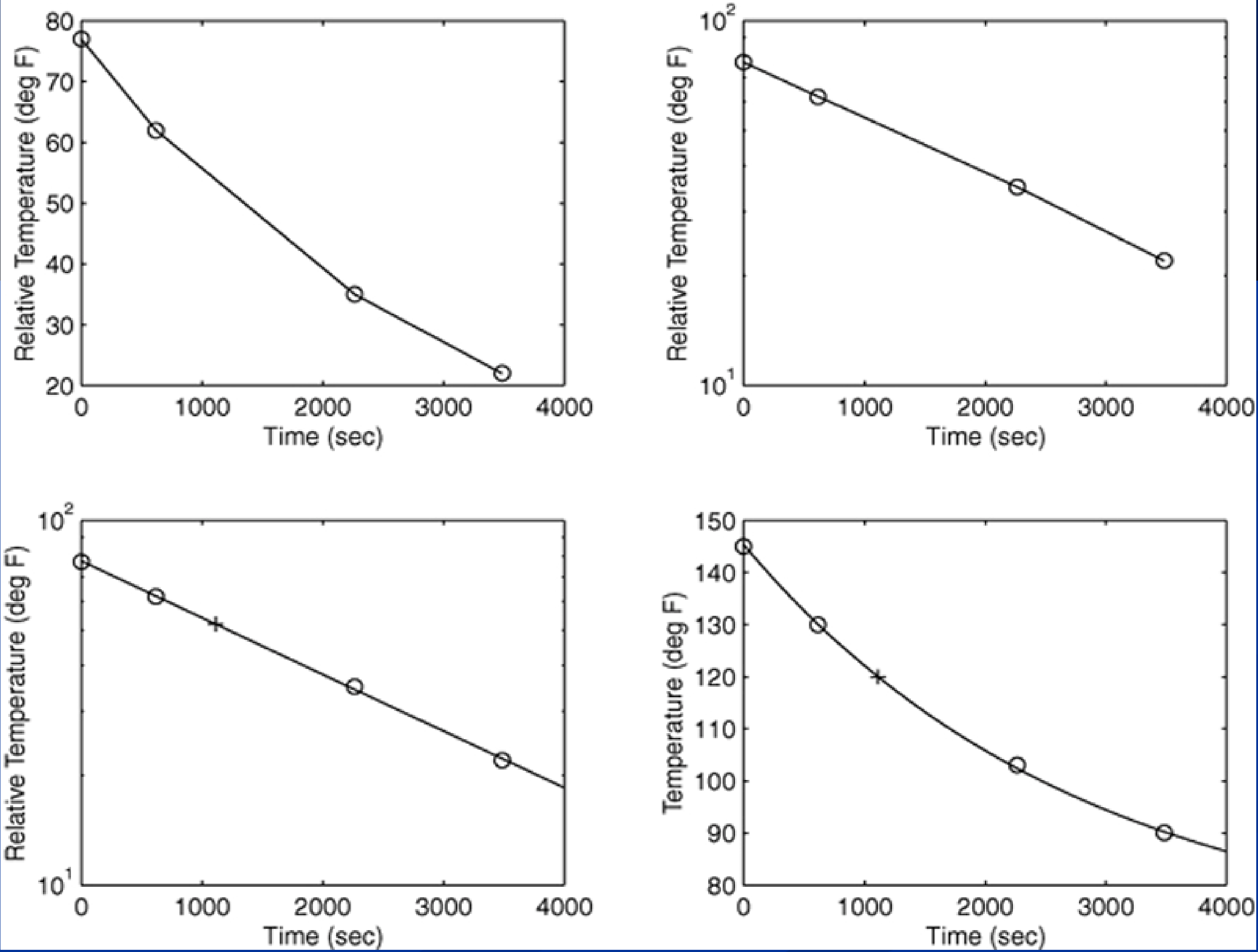
$$w = p_1 z + p_2$$

where the polynomial variables w and z are related to the original data variables x and y by $w = \log_{10} y$ and $z = x$. We can find the exponential function that fits the data by typing

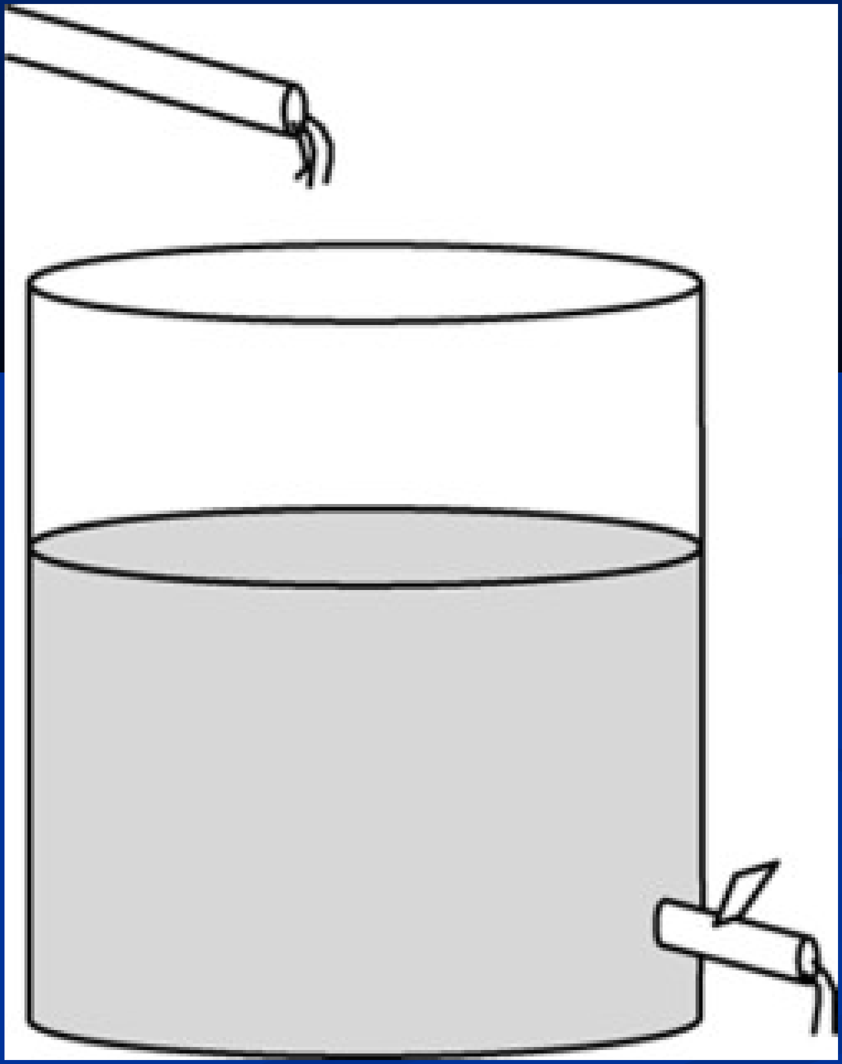
```
p = polyfit(x, log10(y),1)
```

The first element p_1 of the vector p will be m, and the second element p_2 will be $\log_{10} b$. We can find b from $b = 10^{p_2}$.

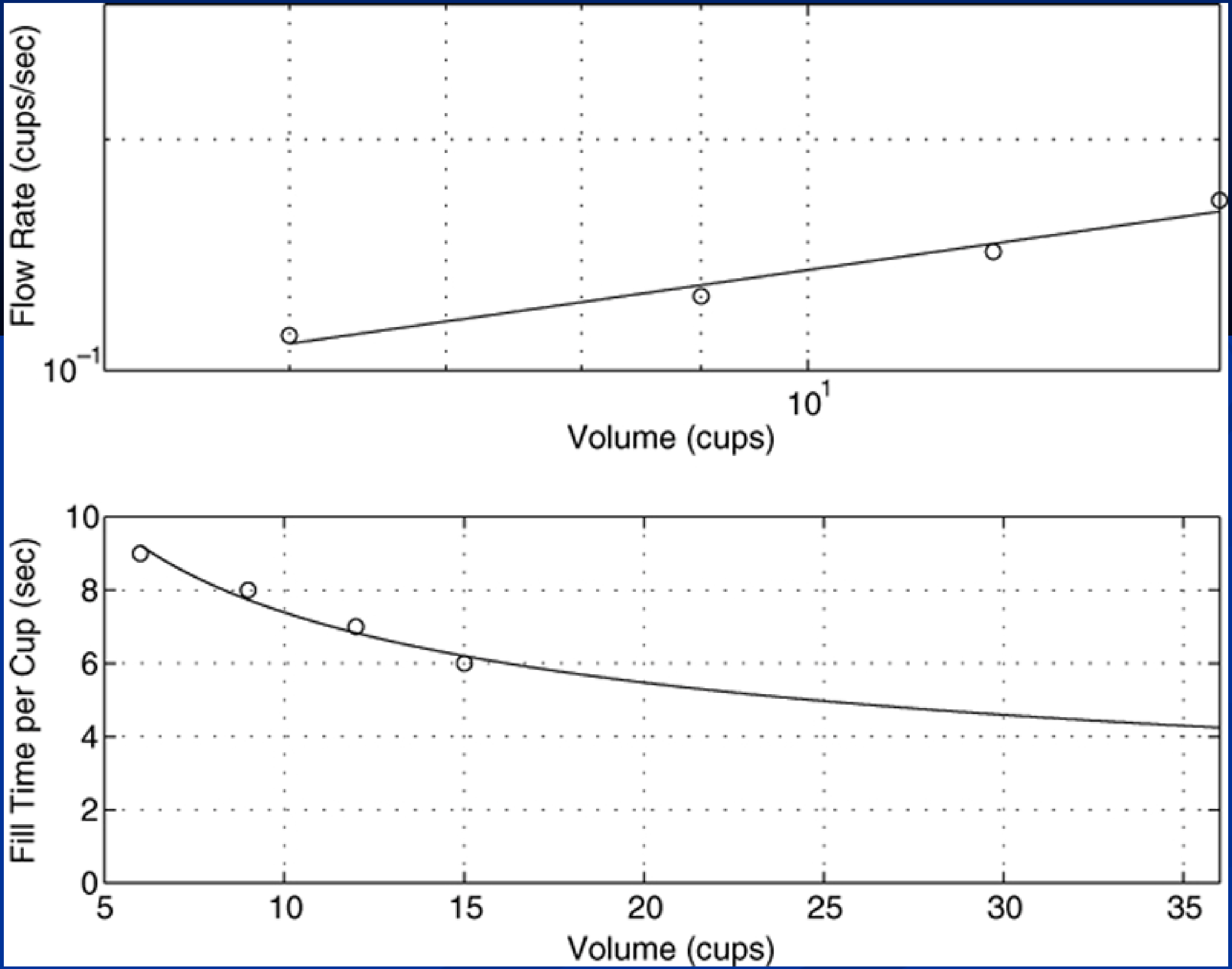
Fitting an exponential function. Temperature of a cooling cup of coffee, plotted on various coordinates. Example 6.1-1. Figure 6.1-3 on page 267.



Fitting a power function. An experiment to verify Torricelli's principle. Example 6.1-2. Figure 6.1-4 on page 269.



Flow rate and fill time for a coffee pot. Figure 6.1-5 on page 270.

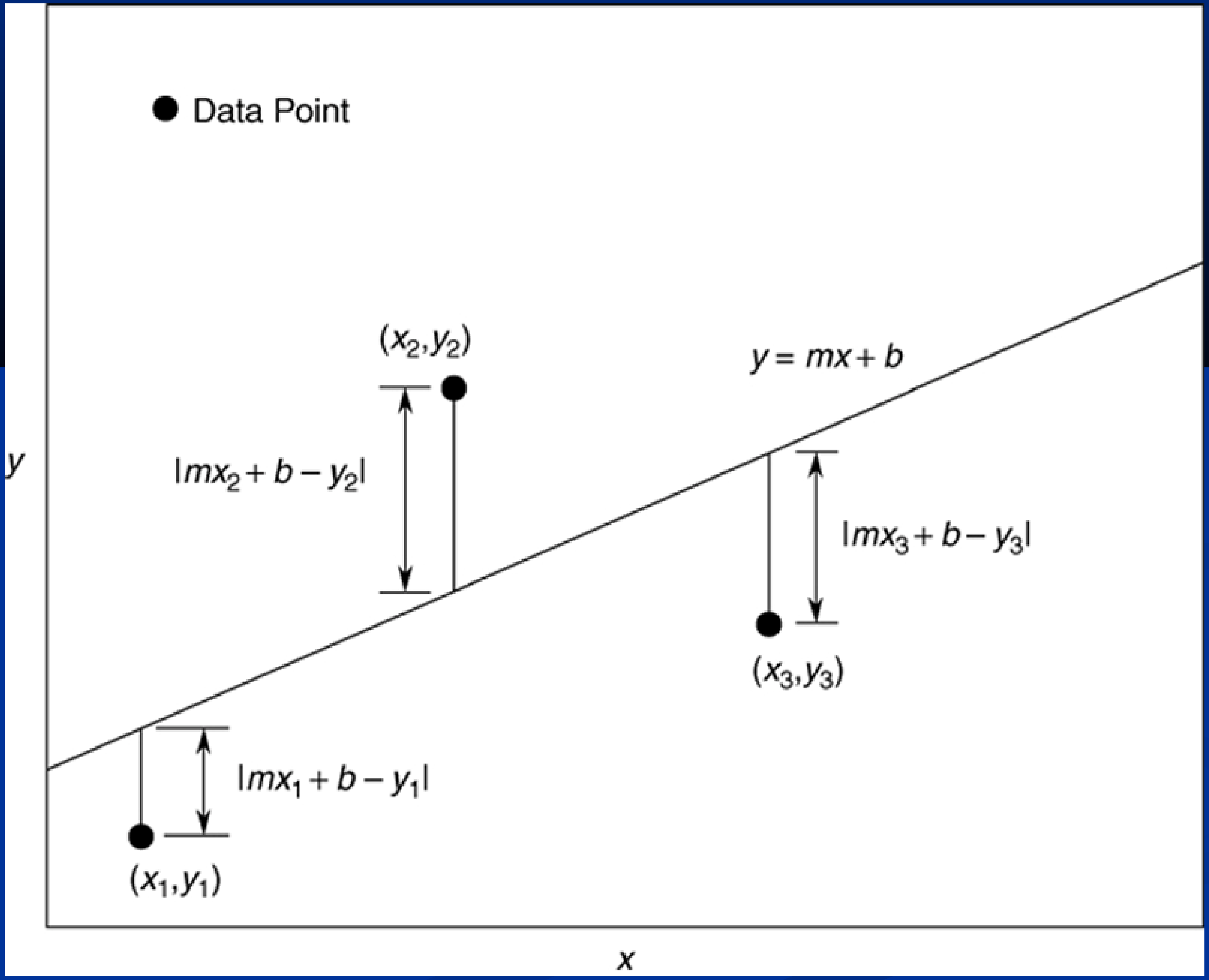


The Least Squares Criterion: used to fit a function $f(x)$. It minimizes the sum of the squares of the residuals, J. J is defined as

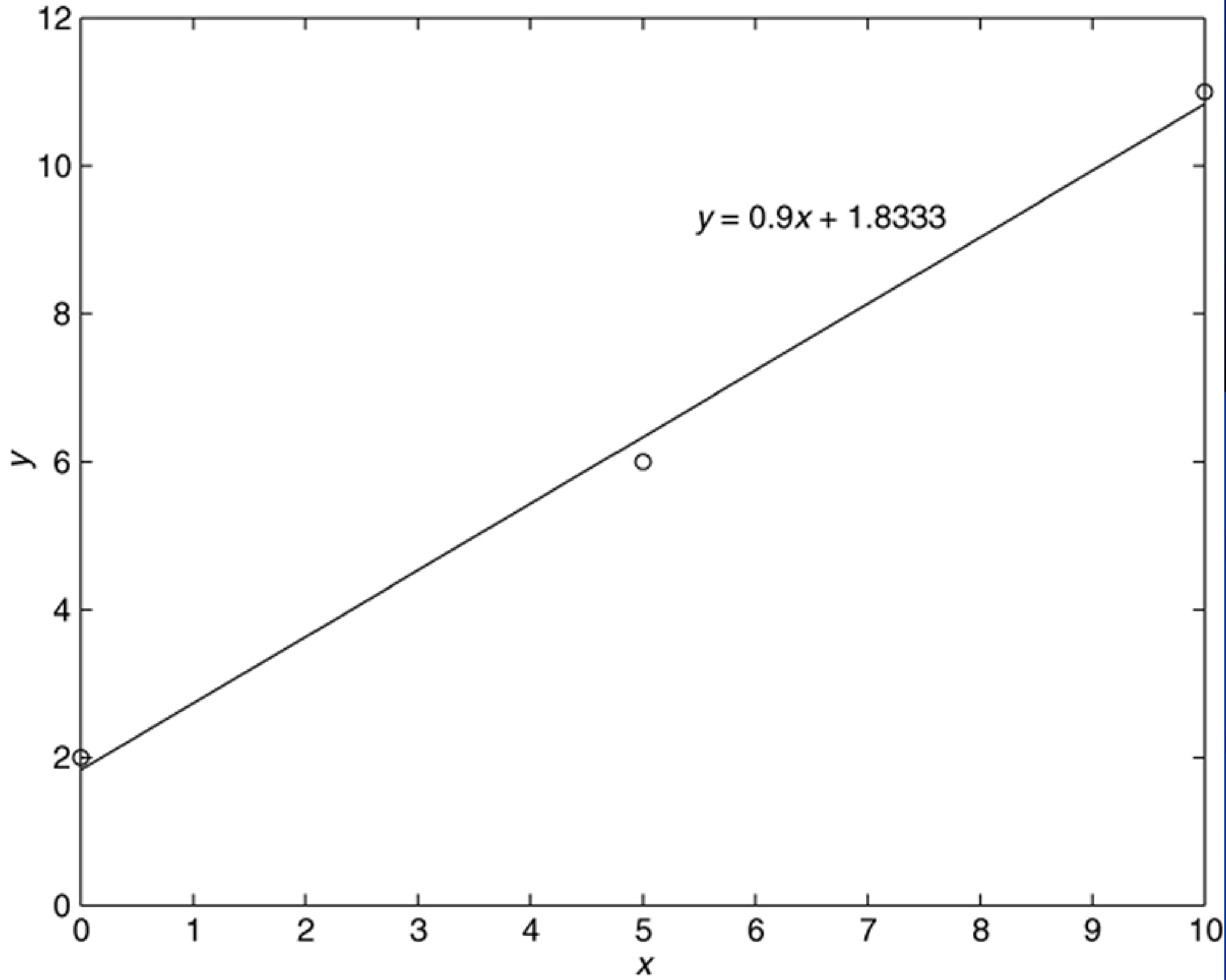
$$J = \sum_{i=1}^{m} [f(x_i) - y_i]^2$$

We can use this criterion to compare the quality of the curve fit for two or more functions used to describe the same data. The function that gives the smallest J value gives the best fit.

Illustration of the least squares criterion.



The least squares fit for the example data.



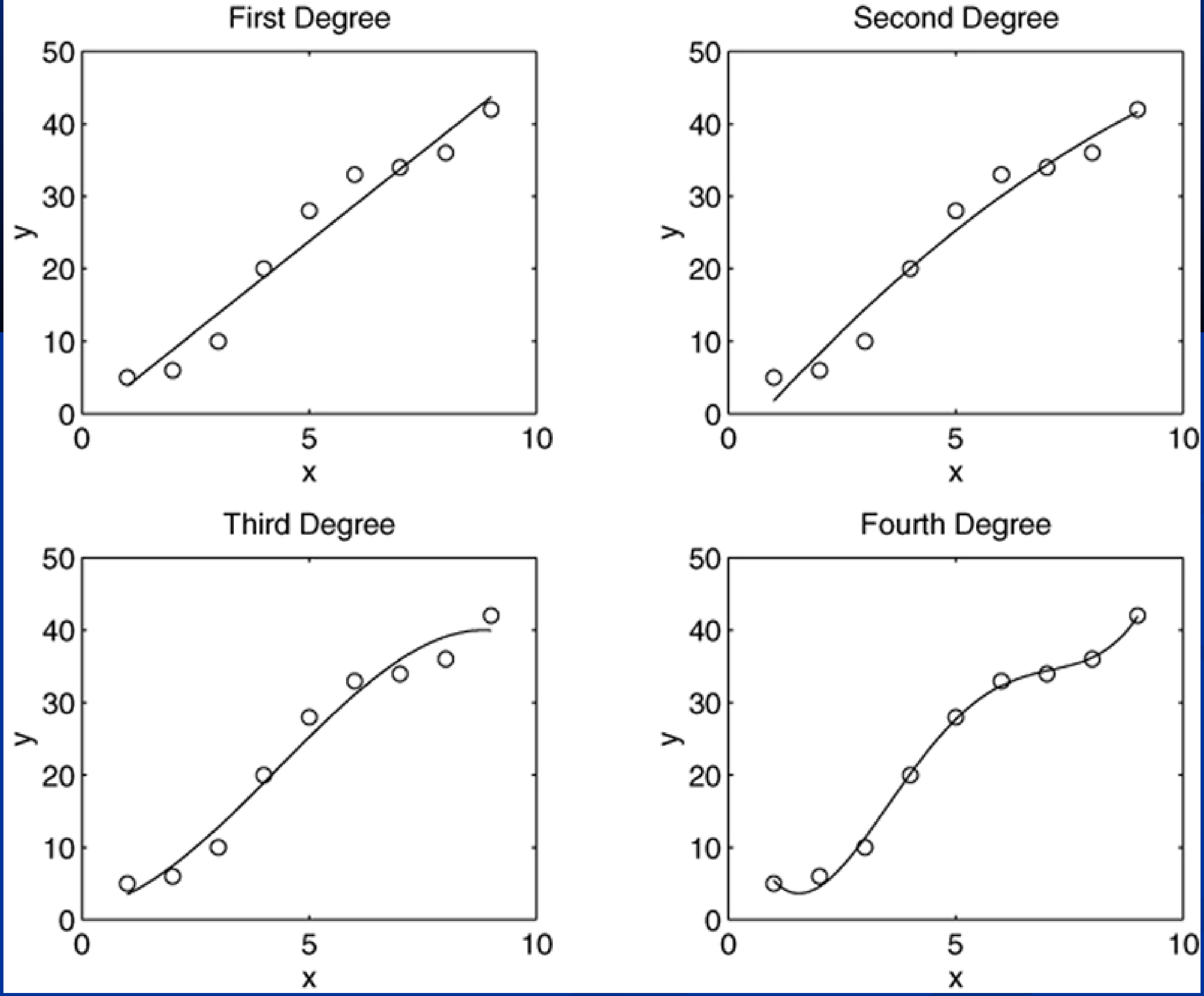
See pages 271-272.

The `polyfit` function is based on the least-squares method. Its syntax is

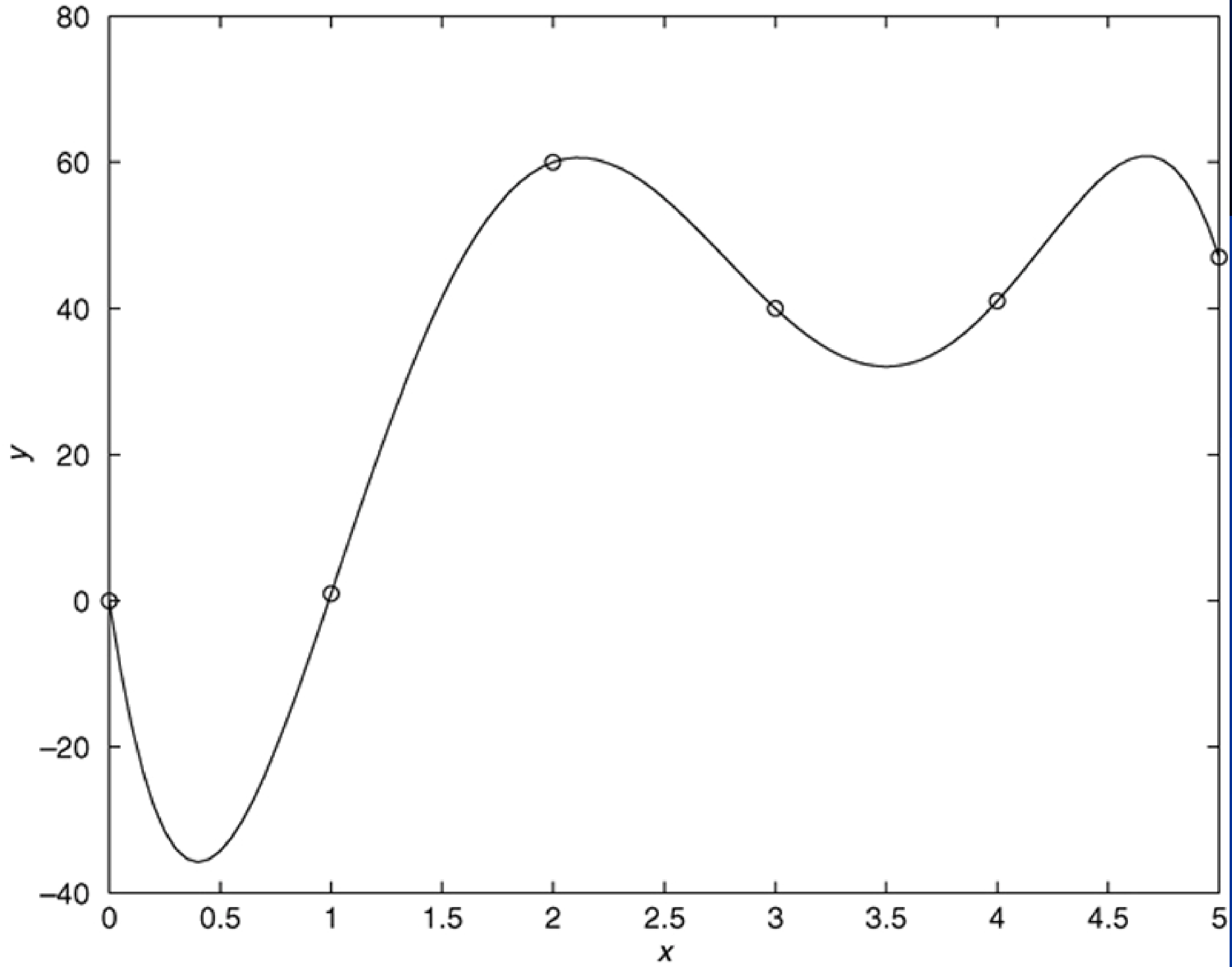
`p = polyfit(x,y,n)`	Fits a polynomial of degree n to data described by the vectors x and y, where x is the independent variable. Returns a row vector p of length n+1 that contains the polynomial coefficients in order of descending powers.

See page 273, Table 6.2-1.

Regression using polynomials of first through fourth degree. Figure 6.2-1 on page 273.



Beware of using polynomials of high degree. An example of a fifth-degree polynomial that passes through all six data points but exhibits large excursions between points. Figure 6.2-2, page 274.



Assessing the Quality of a Curve Fit:

Denote the sum of the squares of the deviation of the *y* values from their mean $\bar{y}$ by *S*, which can be computed from

$$S = \sum_{i=1}^{m} (y_i - \bar{y})^2 \qquad (6.2\text{–}2)$$

This formula can be used to compute another measure of the quality of the curve fit, the *coefficient of determination,* also known as the *r-squared value.* It is defined as

$$r^2 = 1 - \frac{J}{S} \qquad (6.2\text{-}3)$$

The value of *S* indicates how much the data is spread around the mean, and the value of *J* indicates how much of the data spread is unaccounted for by the model.

Thus the ratio *J* / *S* indicates the fractional variation unaccounted for by the model.

For a perfect fit, $J = 0$ and thus $r^2 = 1$. Thus the closer r^2 is to 1, the better the fit. The largest r^2 can be is 1.

It is possible for J to be larger than S, and thus it is possible for r^2 to be negative. Such cases, however, are indicative of a very poor model that should not be used.

As a rule of thumb, a good fit accounts for at least 99 percent of the data variation. This value corresponds to $r^2 \geq 0.99$.

More? See pages 275-276.

Scaling the Data

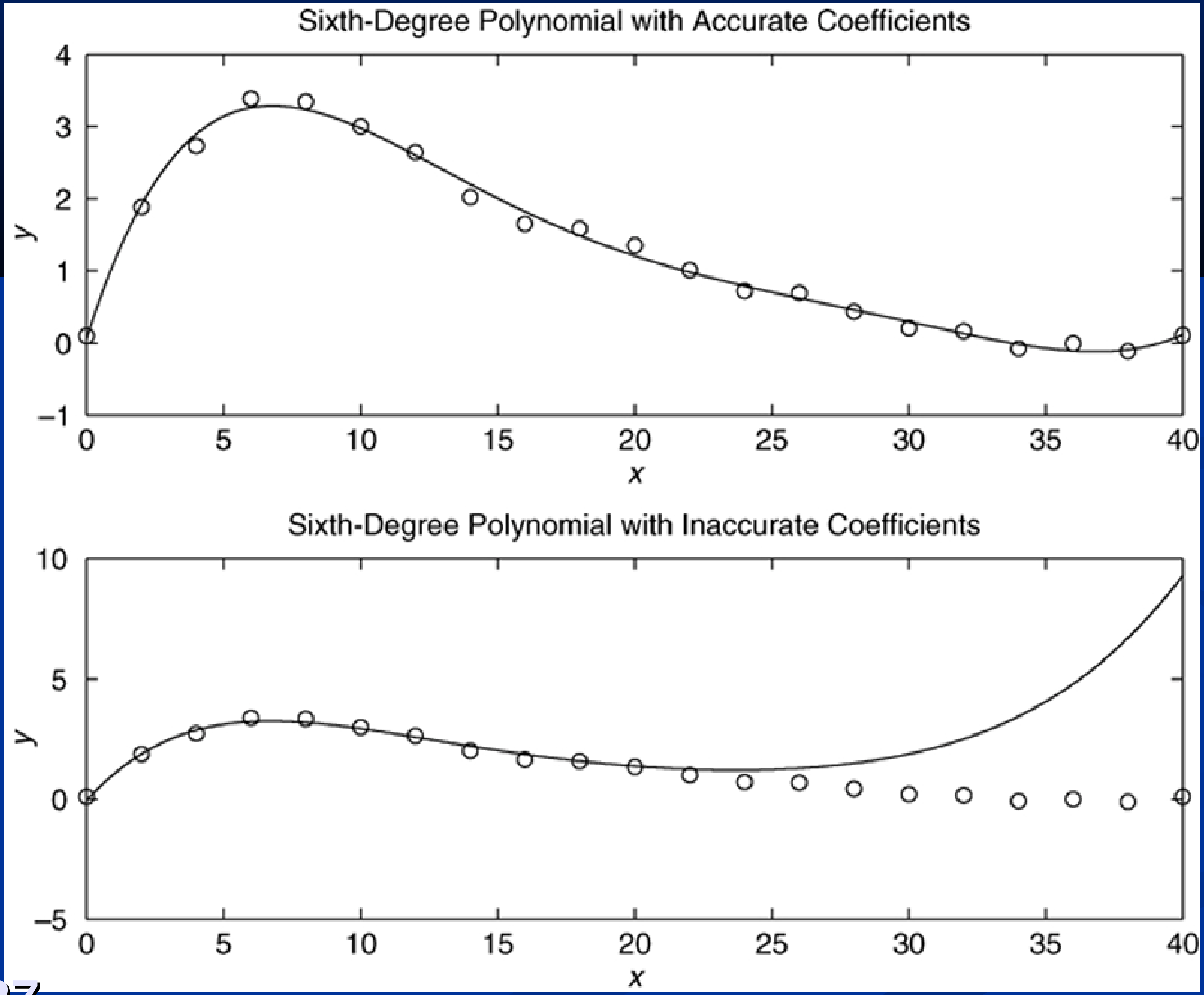
The effect of computational errors in computing the coefficients can be lessened by properly scaling the *x* values. You can scale the data yourself before using `polyfit`. Some common scaling methods are

1. Subtract the minimum *x* value or the mean *x* value from the *x* data, if the range of *x* is small, or

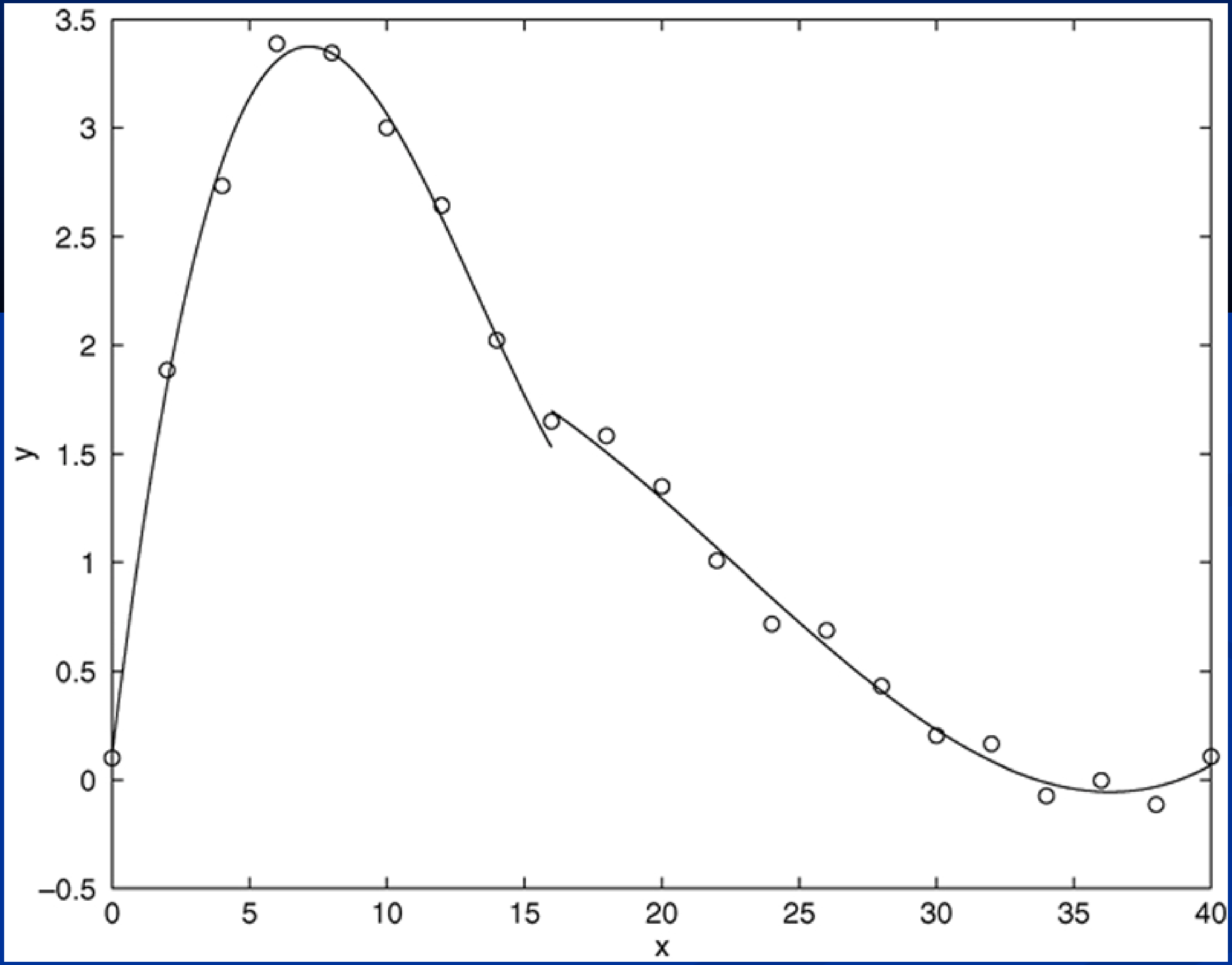
2. Divide the *x* values by the maximum value or the mean value, if the range is large.

More? See pages 276-277.

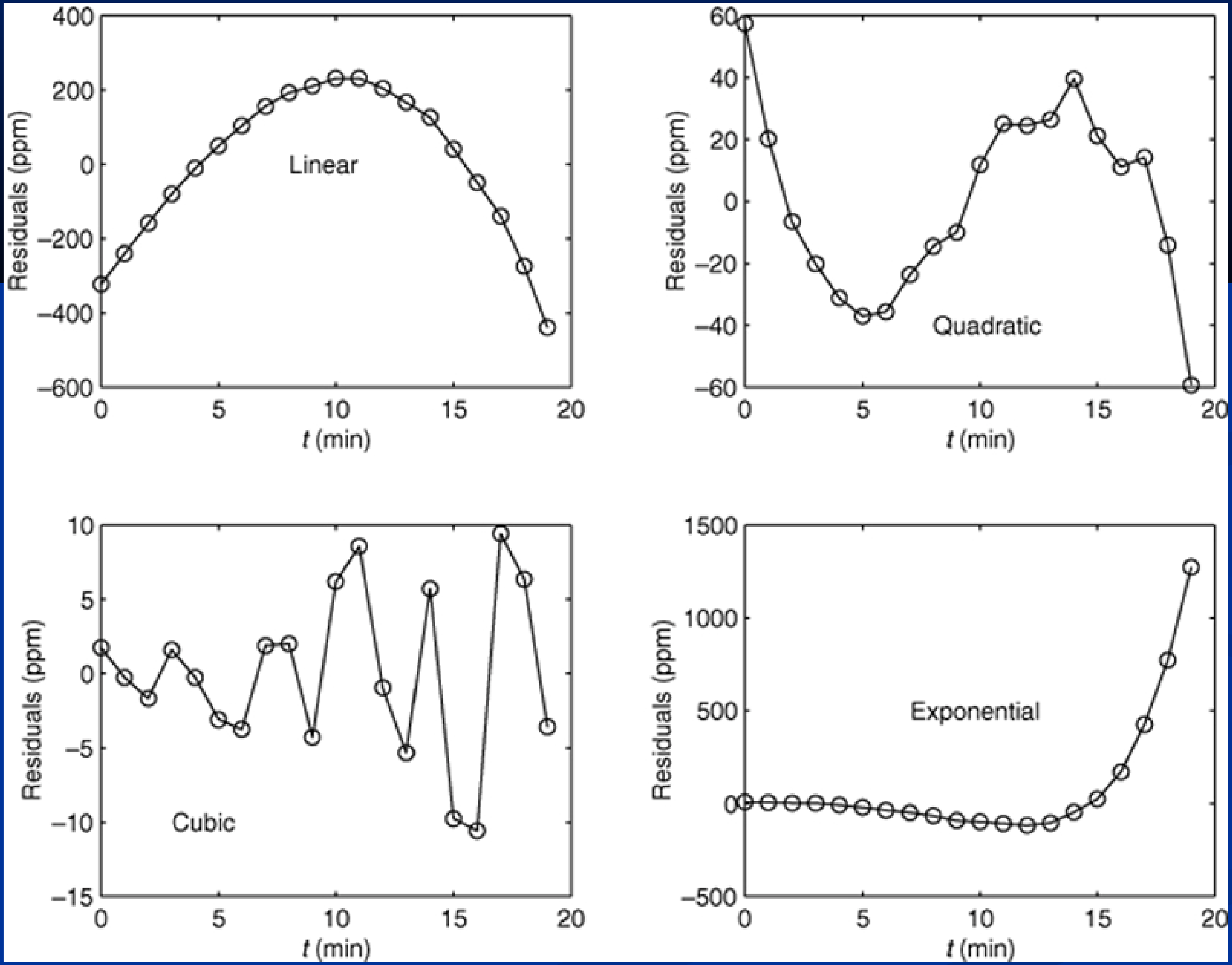
Effect of coefficient accuracy on a sixth-degree polynomial. Top graph: 14 decimal-place accuracy. Bottom graph: 8 decimal-place accuracy.



Avoiding high degree polynomials: Use of two cubics to fit data.

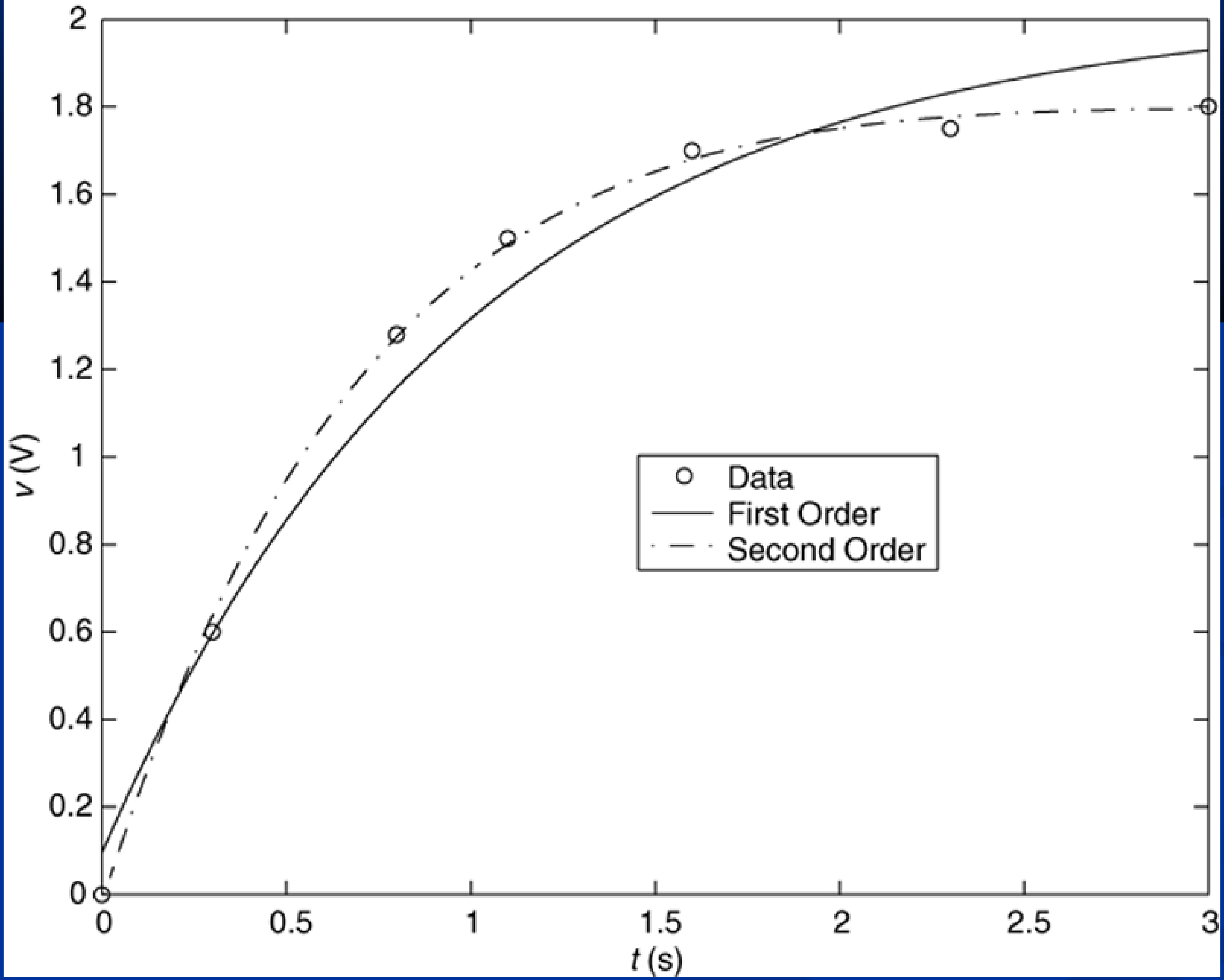


Using Residuals: Residual plots of four models. Figure 6.2-3, page 279.



See pages 277-279.

Linear-in-Parameters Regression: Comparison of first- and second-order model fits. Figure 6.2-4, page 282.



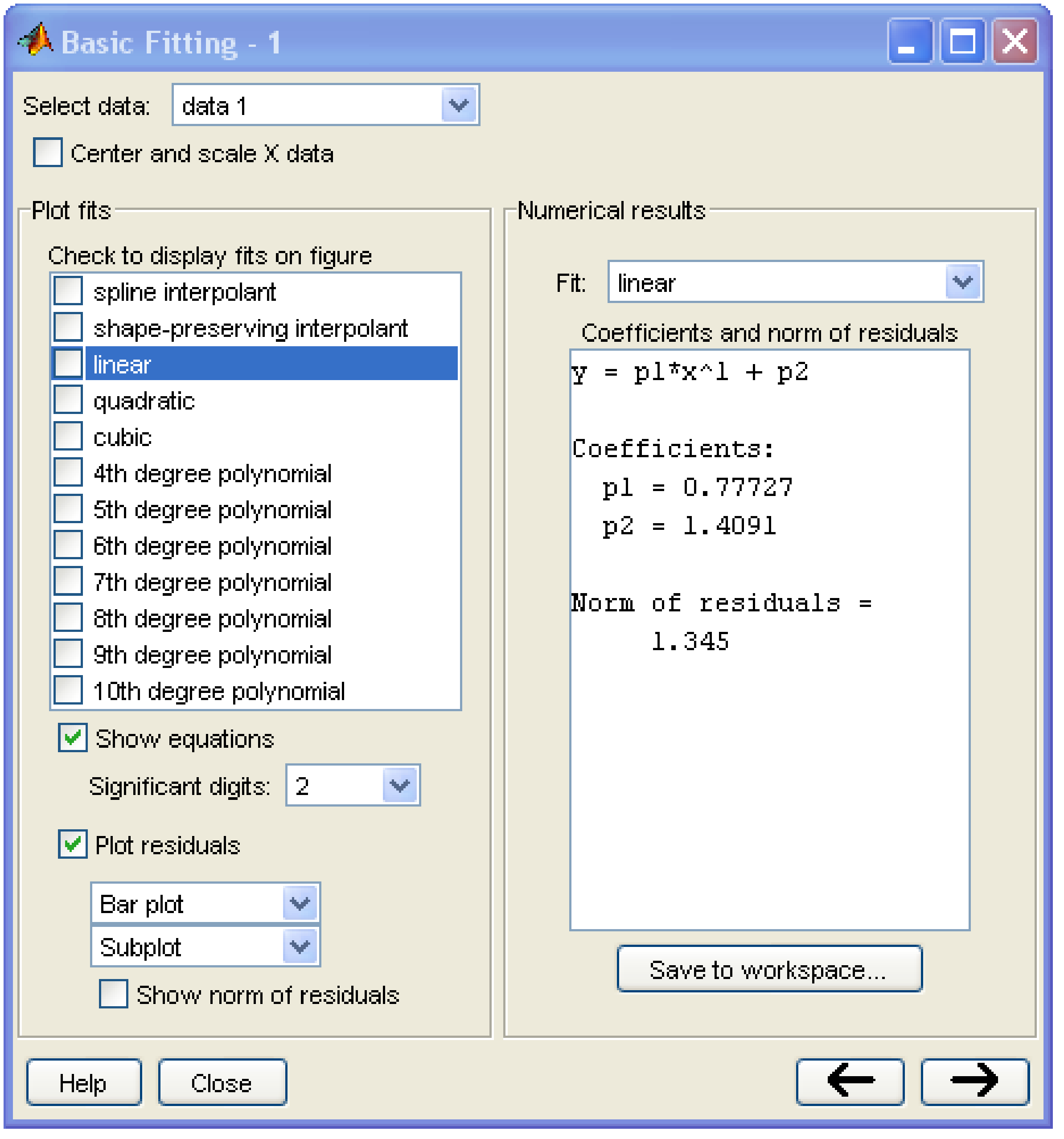
See pages 280-282.

Basic Fitting Interface

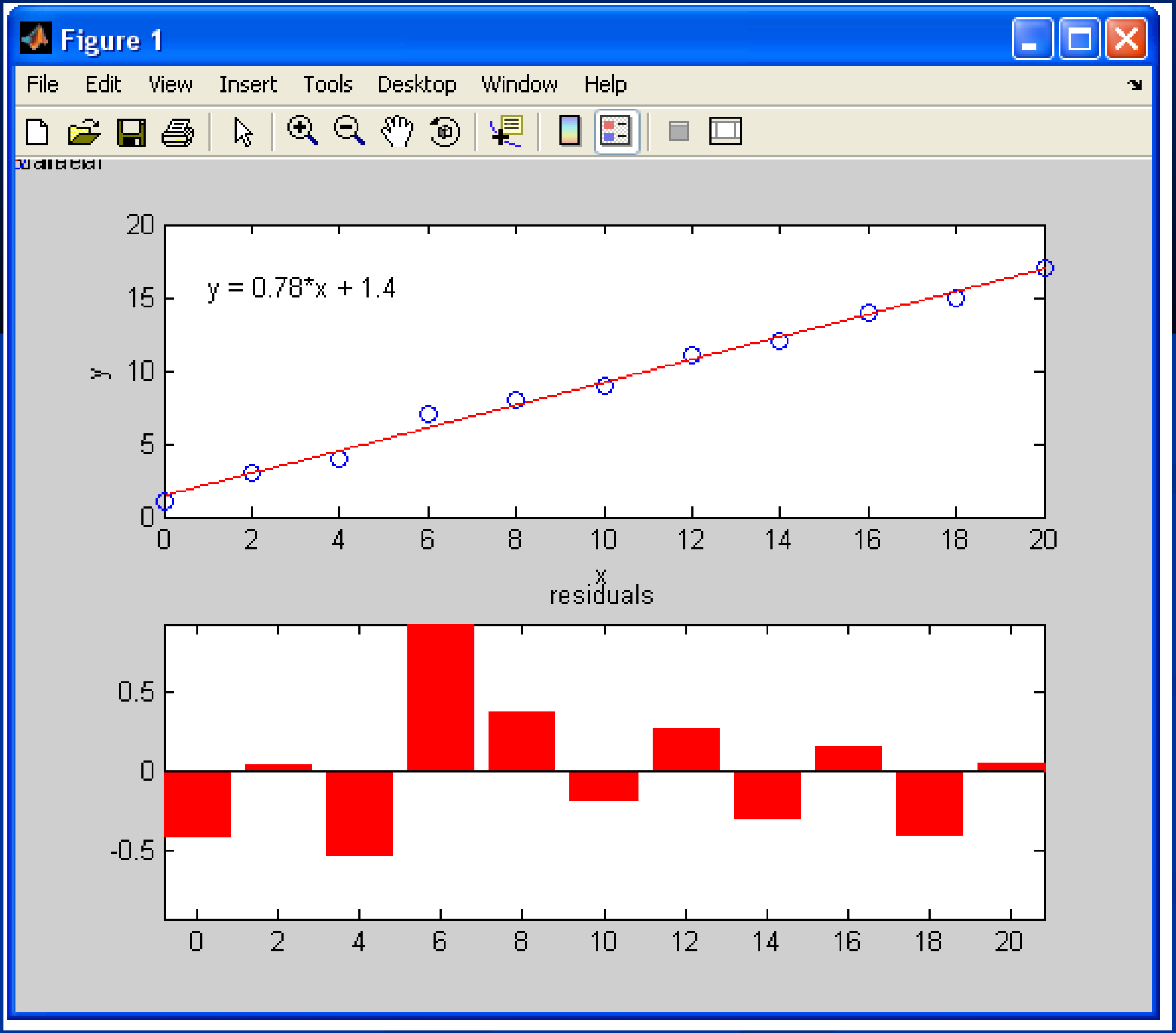
MATLAB supports curve fitting through the Basic Fitting interface. Using this interface, you can quickly perform basic curve fitting tasks within the same easy-to-use environment. The interface is designed so that you can:

- Fit data using a cubic spline or a polynomial up to degree 10.
- Plot multiple fits simultaneously for a given data set.
- Plot the residuals.
- Examine the numerical results of a fit.
- Interpolate or extrapolate a fit.
- Annotate the plot with the numerical fit results and the norm of residuals.
- Save the fit and evaluated results to the MATLAB workspace.

The Basic Fitting interface. Figure 6.3-1, page 283.



A figure produced by the Basic Fitting interface. Figure 6.3-2, page 285.



More? See pages 284-285.