5.5 Additional Examples

▶Example 5.6—Fitting a Line to a Set of Noisy Measurements

The velocity of a falling object in the presence of a constant gravitational field is given by the equation

$$v(t) = at + v_0 \tag{5.4}$$

where $v(t)$ is the velocity at any time t, a is the acceleration due to gravity, and v_0 is the velocity at time 0. This equation is derived from elementary physics—it is known to every freshman physics student. If we plot velocity versus time for the falling object, our (v, t) measurement points should fall along a straight line. However, the same freshman physics student also knows that if we go out into the laboratory and attempt to *measure* the velocity versus time of an object, our measurements will *not* fall along a straight line. They may come close, but they will never line up perfectly. Why not? This happens because we can never make perfect measurements. There is always some *noise* included in the measurements, which distorts them.

There are many cases in science and engineering where there are noisy sets of data such as this, and we wish to estimate the straight line which "best fits" the data. This problem is called the *linear regression* problem. Given a noisy set of measurements (x,y) that appear to fall along a straight line, how can we find the equation of the line

$$y = mx + b \tag{5.5}$$

which "best fits" the measurements? If we can determine the regression coefficients m and b, then we can use this equation to predict the value of y at any given x by evaluating Equation (5.5) for that value of x.

A standard method for finding the regression coefficients m and b is the *method of least squares*. This method is named "least squares" because it produces the line $y = mx + b$ for which the sum of the squares of the differences between the observed y values and the predicted y values is as small as possible. The slope of the least squares line is given by

$$m = \frac{\left(\sum xy\right) - \left(\sum x\right)\bar{y}}{\left(\sum x^2\right) - \left(\sum x\right)\bar{x}} \tag{5.6}$$

and the intercept of the least squares line is given by

$$b = \bar{y} - m\bar{x} \tag{5.7}$$

where

Σx is the sum of the x values
Σx^2 is the sum of the squares of the x values
Σxy is the sum of the products of the corresponding x and y values
$\bar{x}$ is the mean (average) of the x values
$\bar{y}$ is the mean (average) of the y values

Write a program which will calculate the least-squares slope m and y-axis intercept b for a given set of noisy measured data points (x,y). The data points should be read from the keyboard, and both the individual data points and the resulting least-squares fitted line should be plotted.

Solution

1. **State the problem**
   Calculate the slope m and intercept b of a least-squares line that best fits an input data set consisting of an arbitrary number of (x,y) pairs. The input (x,y) data is read from the keyboard. Plot both the input data points and the fitted line on a single plot.

2. **Define the inputs and outputs**
   The inputs required by this program are the number of points to read, plus the pairs of points (x,y).

   The outputs from this program are the slope and intercept of the least-squares fitted line, the number of points going into the fit, and a plot of the input data and the fitted line.

3. **Describe the algorithm**
   This program can be broken down into six major steps:

```
Get the number of input data points
Read the input statistics
Calculate the required statistics
Calculate the slope and intercept
Write out the slope and intercept
Plot the input points and the fitted line
```

   The first major step of the program is to get the number of points to read in. To do this, we will prompt the user and read his or her answer with an `input` function. Next we will read the input (x,y) pairs one pair at a time using an `input` function in a `for` loop. Each pair of input values will be placed in an array (`[x y]`), and the function will return that array to the calling program. Note that a `for` loop is appropriate because we know in advance how many times the loop will be executed.

The pseudocode for these steps is as follows:

```
Print message describing purpose of the program
n_points ← input('Enter number of [x y] pairs:');
for ii = 1:n_points
   temp ← input('Enter [x y] pair:');
   x(ii) ← temp(1)
   y(ii) ← temp(2)
end
```

Next, we must accumulate the statistics required for the calculation. These statistics are the sums Σx, Σy, Σx^2, and Σxy. The pseudocode for these steps is:

```
Clear the variables sum_x, sum_y, sum_x2, and sum_y2
for ii = 1:n_points
   sum_x ← sum_x + x(ii)
   sum_y ← sum_y + y(ii)
   sum_x2 ← sum_x2 + x(ii)^2
   sum_xy ← sum_xy + x(ii)*y(ii)
end
```

Next, we must calculate the slope and intercept of the least-squares line. The pseudocode for this step is just the MATLAB versions of Equations (5.6) and (5.7).

```
x_bar ← sum_x / n_points
y_bar ← sum_y / n_points
slope ← (sum_xy-sum_x * y_bar)/(sum_x2 - sum_x * x_bar)
y_int ← y_bar - slope * x_bar
```

Finally, we must write out and plot the results. The input data points should be plotted with circular markers and without a connecting line, while the fitted line should be plotted as a solid 2-pixel-wide line. To do this, we will need to plot the points first, set `hold on`, plot the fitted line, and set `hold off`. We will add titles and a legend to the plot for completeness.

4. **Turn the algorithm into MATLAB statements**
   The final MATLAB program is as follows:

```
%
%  Purpose:
%    To perform a least-squares fit of an input data set
%    to a straight line, and print out the resulting slope
%    and intercept values. The input data for this fit
%    comes from a user-specified input data file.
%
```

```
%  Record of revisions:
%      Date         Programmer             Description of change
%      ====         ==========             =====================
%    01/30/18     S. J. Chapman          Original code
%
% Define variables:
%   ii            -- Loop index
%   n_points      -- Number in input [x y] points
%   slope         -- Slope of the line
%   sum_x         -- Sum of all input x values
%   sum_x2        -- Sum of all input x values squared
%   sum_xy        -- Sum of all input x*y yalues
%   sum_y         -- Sum of all input y values
%   temp          -- Variable to read user input
%   x             -- Array of x values
%   x_bar         -- Average x value
%   y             -- Array of y values
%   y_bar         -- Average y value
%   y_int         -- y-axis intercept of the line

disp('This program performs a least-squares fit of an');
disp('input data set to a straight line.');
n_points = input('Enter the number of input [x y] points:');

% Read the input data
for ii = 1:n_points
   temp = input('Enter [x y] pair:');
   x(ii) = temp(1);
   y(ii) = temp(2);
end

% Accumulate statistics
sum_x = 0;
sum_y = 0;
sum_x2 = 0;
sum_xy = 0;
for ii = 1:n_points
   sum_x  = sum_x + x(ii);
   sum_y  = sum_y + y(ii);
   sum_x2 = sum_x2 + x(ii)^2;
   sum_xy = sum_xy + x(ii) * y(ii);
end

% Now calculate the slope and intercept.
x_bar = sum_x / n_points;
y_bar = sum_y / n_points;
slope = (sum_xy - sum_x * y_bar) / (sum_x2 - sum_x * x_bar);
y_int = y_bar - slope * x_bar;
```

```
% Tell user.
disp('Regression coefficients for the least-squares line:');
fprintf('  Slope (m)      = %8.3f\n', slope);
fprintf('  Intercept (b)  = %8.3f\n', y_int);
fprintf('  No. of points  = %8d\n', n_points);

% Plot the data points as blue circles with no
% connecting lines.
plot(x,y,'bo');
hold on;

% Create the fitted line
xmin = min(x);
xmax = max(x);
ymin = slope * xmin + y_int;
ymax = slope * xmax + y_int;

% Plot a solid red line with no markers
plot([xmin xmax],[ymin ymax],'r-','LineWidth',2);
hold off;

% Add a title and legend
title ('\bfLeast-Squares Fit');
xlabel('\bf\itx');
ylabel('\bf\ity');
legend('Input data','Fitted line');
grid on
```

5. **Test the program**

   To test this program, we will try a simple data set. For example, if every point in the input data set falls exactly along a line, then the resulting slope and intercept should be exactly the slope and intercept of that line. Thus the data set

```
[1.1 1.1]
[2.2 2.2]
[3.3 3.3]
[4.4 4.4]
[5.5 5.5]
[6.6 6.6]
[7.7 7.7]
```

   should produce a slope of 1.0 and an intercept of 0.0. If we run the program with these values, the results are:

```
» lsqfit
This program performs a least-squares fit of an
input data set to a straight line.
```

```
Enter the number of input [x y] points: 7
Enter [x y] pair: [1.1 1.1]
Enter [x y] pair: [2.2 2.2]
Enter [x y] pair: [3.3 3.3]
Enter [x y] pair: [4.4 4.4]
Enter [x y] pair: [5.5 5.5]
Enter [x y] pair: [6.6 6.6]
Enter [x y] pair: [7.7 7.7]
Regression coefficients for the least-squares line:
  Slope (m)      =    1.000
  Intercept (b)  =    0.000
  No. of points  =        7
```

Now let's add some noise to the measurements. The data set becomes

```
[1.1 1.01]
[2.2 2.30]
[3.3 3.05]
[4.4 4.28]
[5.5 5.75]
[6.6 6.48]
[7.7 7.84]
```

If we run the program with these values, the results are:

```
» lsqfit
This program performs a least-squares fit of an
input data set to a straight line.
Enter the number of input [x y] points: 7
Enter [x y] pair: [1.1 1.01]
Enter [x y] pair: [2.2 2.30]
Enter [x y] pair: [3.3 3.05]
Enter [x y] pair: [4.4 4.28]
Enter [x y] pair: [5.5 5.75]
Enter [x y] pair: [6.6 6.48]
Enter [x y] pair: [7.7 7.84]
Regression coefficients for the least-squares line:
  Slope (m)      =    1.024
  Intercept (b)  =   -0.120
  No. of points  =        7
```

If we calculate the answer by hand, it is easy to show that the program gives the correct answers for our two test data sets. The noisy input data set and the resulting least-squares fitted line are shown in Figure 5.4.

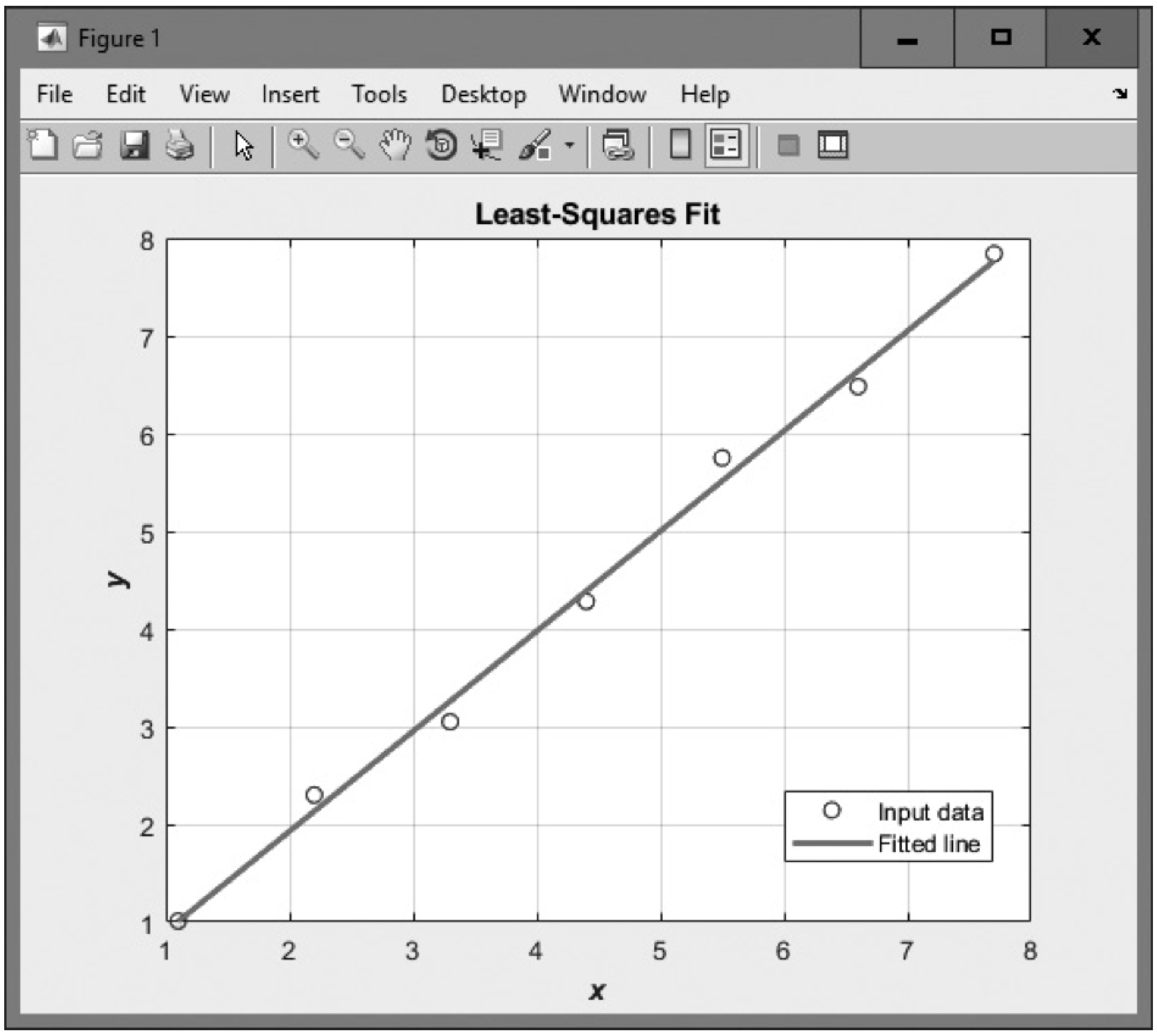

Figure 5.4 A noisy data set with a least-squares fitted line.

Example 5.6 uses several of the plotting capabilities that we introduced in Chapter 3. It uses the `hold` command to allow multiple plots to be placed on the same axes, the `LineWidth` property to set the width of the least-squares fitted line, and escape sequences to make the title boldface and the axis labels bold italic.

Example 5.7—Physics—The Flight of a Ball

If we assume negligible air friction and ignore the curvature of the Earth, a ball that is thrown into the air from any point on the Earth's surface will follow a parabolic flight path (see Figure 5.5a). The height of the ball at any time t after it is thrown is given by Equation (5.8):

$$y(t) = y_0 + v_{y0}t + \frac{1}{2}gt^2 \tag{5.8}$$

where y_0 is the initial height of the object above the ground, v_{y0} is the initial vertical velocity of the object, and g is the acceleration due to the Earth's gravity. The