

Developing Models from Data

Here we develop a systematic way of developing a model from data containing—so much scatter that it is difficult to draw by eye a straight line that passes near most of the data points. This method—called the *least squares method*—is easy to use with a computer, and we illustrate how to do it with MATLAB. ■

C.1 FITTING MODELS TO SCATTERED DATA

In practice the data often will not lie very close to a straight line, and if we ask two people to draw a straight line passing as close as possible to all the data points, we will probably receive two different answers. A systematic and objective way of obtaining a straight line describing the data is the least-squares method. Suppose we want to find the coefficients of the straight line $y = mx + b$ that best fits the following data.

x	0	5	10
y	2	6	11

According to the least-squares criterion, the line that gives the best fit is the one that minimizes J , the sum of the squares of the vertical differences between the line and the data points (see Figure C.1.1). These differences are called the *residuals*. Here there are three data points and J is given by

$$J = \sum_{i=1}^3 (mx_i + b - y_i)^2$$

Substituting the data values (x_i, y_i) given in the table, we obtain

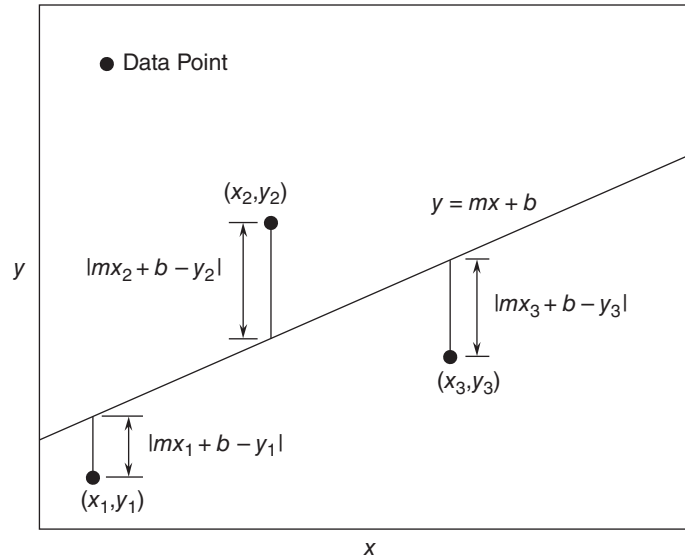
$$J = (0m + b - 2)^2 + (5m + b - 6)^2 + (10m + b - 11)^2$$

The values of m and b that minimize J can be found from $\partial J/\partial m = 0$ and $\partial J/\partial b = 0$.

$$\frac{\partial J}{\partial m} = 2(5m + b - 6)(5) + 2(10m + b - 11)(10) = 250m + 30b - 280 = 0$$

$$\frac{\partial J}{\partial b} = 2(b - 2) + 2(5m + b - 6) + 2(10m + b - 11) = 30m + 6b - 38 = 0$$

Figure C.1.1 Illustration of the least-squares criterion.



These conditions give the following equations that must be solved for the two unknowns m and b .

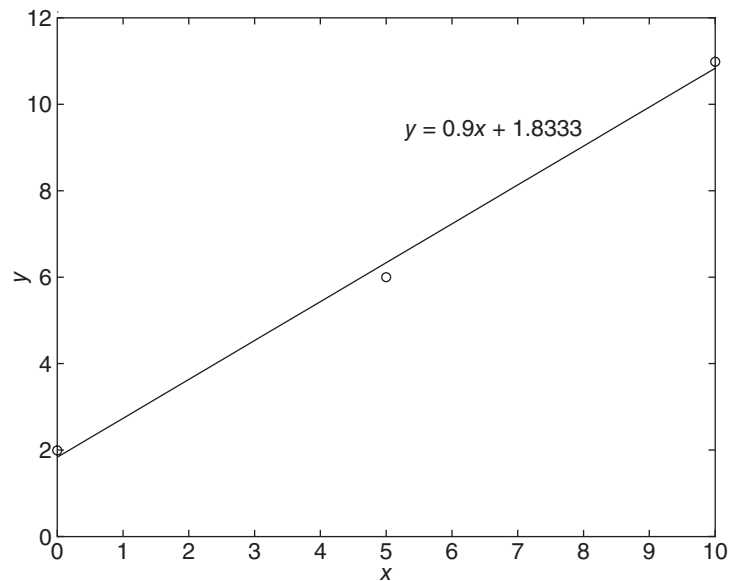
$$250m + 30b = 280$$

$$30m + 6b = 38$$

The solution is $m = 9/10$ and $b = 11/6$. The best straight line in the least-squares sense is $y = (9/10)x + 11/6$. This is shown in Figure C.1.2 along with the data.

If we evaluate this equation at the data values $x = 0, 5,$ and 10 , we obtain the values $y = 1.8333, 6.3333,$ and 10.8333 . These values are different than the given data values $y = 2, 6,$ and 11 because the line is not a perfect fit to the data. The value of J is $J = (1.8333 - 2)^2 + (6.3333 - 6)^2 + (10.8333 - 11)^2 = 0.1666$. No other straight line will give a lower value of J for these data.

Figure C.1.2 Example of a least-squares fit.



C.1.1 THE GENERAL LINEAR CASE

We can generalize the preceding results to obtain formulas for the coefficients m and b in the linear equation $y = mx + b$. Note that for n data points,

$$J = \sum_{i=1}^n (mx_i + b - y_i)^2$$

The values of m and b that minimize J are found from $\partial J/\partial m = 0$ and $\partial J/\partial b = 0$. These conditions give the following equations that must be solved for m and b :

$$\frac{\partial J}{\partial m} = 2 \sum_{i=1}^n (mx_i + b - y_i) x_i = 2 \sum_{i=1}^n mx_i^2 + 2 \sum_{i=1}^n bx_i - 2 \sum_{i=1}^n y_i x_i = 0$$

$$\frac{\partial J}{\partial b} = 2 \sum_{i=1}^n (mx_i + b - y_i) = 2 \sum_{i=1}^n mx_i + 2 \sum_{i=1}^n b - 2 \sum_{i=1}^n y_i = 0$$

These equations become

$$m \sum_{i=1}^n x_i^2 + b \sum_{i=1}^n x_i = \sum_{i=1}^n y_i x_i \quad (\text{C.1.1})$$

$$m \sum_{i=1}^n x_i + bn = \sum_{i=1}^n y_i \quad (\text{C.1.2})$$

These are two linear equations in terms of m and b .

Because the exponential and power functions form straight lines on semilog and log-log axes respectively, we can use the previous results after computing the logarithms of the data.

Fitting Data with the Power Function

EXAMPLE C.1.1

■ Problem

Find a functional description of the following data:

x	1	2	3	4
y	5.1	19.5	46	78

■ Solution

These data do not lie close to a straight line when plotted on linear or semilog axes. However, they do when plotted on log-log axes. Thus a power function $y = bx^m$ can describe the data. Using the transformations $X = \log x$ and $Y = \log y$, we obtain the new data table:

$X = \log x$	0	0.3010	0.4771	0.6021
$Y = \log y$	0.7076	1.2900	1.6628	1.8921

From this table we obtain

$$\sum_{i=1}^4 X_i = 1.3803 \quad \sum_{i=1}^4 Y_i = 5.5525$$

$$\sum_{i=1}^4 X_i Y_i = 2.3208 \quad \sum_{i=1}^4 X_i^2 = 0.6807$$

Using X , Y , and $B = \log b$ instead of x , y , and b in (C.1.1) and (C.1.2) we obtain

$$0.6807m + 1.3803B = 2.3208$$

$$1.3803m + 4B = 5.5525$$

The solution is $m = 1.9802$ and $B = 0.7048$. This gives $b = 10^B = 5.068$. Thus, the desired function is $y = 5.068x^{1.9802}$.

C.1.2 CONSTRAINING MODELS TO PASS THROUGH A GIVEN POINT

Many applications require a model whose form is dictated by physical principles. For example, the force-extension model of a spring must pass through the origin $(0, 0)$ because the spring exerts no force when it is not stretched. Thus a linear model $y = mx + b$ sometimes must have a zero value for b . However, in general the least-squares method will give a nonzero value for b because of the scatter or measurement error that is usually present in the data.

To obtain a zero-intercept model of the form $y = mx$, we must derive the equation for m from basic principles. The sum of the squared residuals in this case is

$$J = \sum_{i=1}^n (mx_i - y_i)^2$$

Computing the derivative $\partial J/\partial m$ and setting it equal to zero gives the result

$$m \sum_{i=1}^n x_i^2 = \sum_{i=1}^n x_i y_i \quad (\text{C.1.3})$$

which can be easily solved for m .

If the model is required to pass through a point not at the origin, say the point (x_0, y_0) , subtract x_0 from all the x values, subtract y_0 from all the y values, and then use (C.1.3) to find the coefficient m . The resulting equation will be of the form

$$y = m(x - x_0) + y_0 \quad (\text{C.1.4})$$

EXAMPLE C.1.2

Point Constraint

■ Problem

Consider the data given at the beginning of this section.

$$\begin{array}{c|ccc} x & 0 & 5 & 10 \\ \hline y & 2 & 6 & 11 \end{array}$$

We found that the best-fit line is $y = (9/10)x + 11/6$. Find the best-fit line that passes through the point $x = 10$, $y = 11$.

■ Solution

Subtracting 10 from all the x values and 11 from all the y values, we obtain a new set of data in terms of the new variables $X = x - 10$ and $Y = y - 11$.

$$\begin{array}{c|ccc} X & -10 & -5 & 0 \\ \hline Y & -9 & -5 & 0 \end{array}$$

Expressing (C.1.3) in terms of the new variables X and Y , we have

$$m \sum_{i=1}^3 X_i^2 = \sum_{i=1}^3 X_i Y_i$$

$$\sum_{i=1}^3 X_i^2 = (-10)^2 + 5^2 + 0 = 125$$

$$\sum_{i=1}^3 X_i Y_i = (-10)(-9) + (-5)(-5) + 0 = 115$$

Thus, $m = 115/125 = 23/25$ and the best-fit line is $Y = (23/25)X$. In terms of the original variables, this line is expressed as $y - 11 = (23/25)(x - 10)$ or $y = (23/25)x + 9/5$.

CONSTRAINING A COEFFICIENT

Sometimes we know from physical theory that the data can be described by a function with a specified form and specified values of one or more of its coefficients. For example, the fluid-drag relation states that $D = \rho A C_D v^2/2$. In this case, we know that the relation is a power function with an exponent of 2, and we need to estimate the value of the drag coefficient C_D . In such cases, we can modify the least-squares method to find the best-fit function of a specified form.

Fitting a Power Function with a Known Exponent

EXAMPLE C.1.3

■ Problem

Fit the power function $y = bx^m$ to the data y_i . The value of m is known.

■ Solution

The least-squares criterion is

$$J = \sum_{i=1}^n (bx^m - y_i)^2$$

To obtain the value of b that minimizes J , we must solve $\partial J/\partial b = 0$.

$$\frac{\partial J}{\partial b} = 2 \sum_{i=1}^n x_i^m (bx_i^m - y_i) = 0$$

This gives

$$b = \frac{\sum_{i=1}^n x_i^m y_i}{\sum_{i=1}^n x_i^{2m}} \quad (1)$$

C.1.3 THE QUALITY OF A CURVE FIT

In general, if the arbitrary function $y = f(x)$ is used to represent the data, then the error in the representation is given by $e_i = f(x_i) - y_i$, for $i = 1, 2, 3, \dots, n$. The error e_i is the difference between the data value y_i and the value of y obtained from the function; that is, $f(x_i)$. The least-squares criterion used to fit a function $f(x)$ is the sum of the

squares of the residuals, J . It is defined as

$$J = \sum_{i=1}^n [f(x_i) - y_i]^2 \quad (\text{C.1.5})$$

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.

We 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

$$S = \sum_{i=1}^n (y_i - \bar{y})^2 \quad (\text{C.1.6})$$

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

$$r^2 = 1 - \frac{J}{S} \quad (\text{C.1.7})$$

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. The value of S is an indication of how much the data is spread around the mean, and the value of J indicates how much of the data spread is left unaccounted for by the model. Thus, the ratio J/S indicates the fractional variation left unaccounted for by the model. 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 very good fit accounts for at least 99% of the data variation. This corresponds to $r^2 \geq 0.99$.

For example, the function $y = 9/10x + 11/6$ derived at the beginning of this section has the values $S = 40.6667$, $J = 0.1666$, and $r^2 = 0.9959$, which indicates a very good fit. The line $y = (23/25)x + 9/5$, which is constrained to pass through the point $x = 10$, $y = 11$ gives the values $S = 40.6667$, $J = 0.2$, and $r^2 = 0.9951$. So the constraint degraded the quality of the fit but very slightly.

The power function $y = 5.068x^{1.9802}$ derived in Example C.1.1 has the values $S = 3085.8$, $J = 2.9192$, and $r^2 = 0.9991$. Thus its fit is very good.

When the least-squares method is applied to fit quadratic and higher-order polynomials, the resulting equations for the coefficients are linear algebraic equations, which are easily solved. Their solution forms the basis for MATLAB algorithm contained in the `polyfit` function, which is discussed in Section C.2.

C.1.4 INTEGRAL FORM OF THE LEAST-SQUARES CRITERION

Sometimes we must obtain a linear description of a process over a range of the independent variable so large that linearization is impractical. In such cases we can apply the least-squares method to obtain the linear description. Because there are no data in such cases, we use the integral form of the least-squares criterion.

EXAMPLE C.1.4

Fitting a Linear Function to a Power Function

■ Problem

- a. Fit the linear function $y = mx$ to the power function $y = ax^n$ over the range $0 \leq x \leq L$. The values of a and n are given.

- b. Apply the result to the Aerobee drag function $D = 0.00056v^2$ over the range $0 \leq v \leq 1000$, discussed in Example 1.3.4.

■ **Solution**

- a. The appropriate least-squares criterion is the integral of the square of the difference between the linear model and the power function over the stated range. Thus,

$$J = \int_0^L (mx - ax^n)^2 dx$$

To obtain the value of m that minimizes J , we must solve $\partial J / \partial m = 0$.

$$\frac{\partial J}{\partial m} = 2 \int_0^L x(mx - ax^n) dx = 0$$

This gives

$$m = \frac{3a}{n+2} L^{n-1} \quad (1)$$

- b. For the Aerobee drag function $D = 0.00056v^2$, $a = 0.00056$, $n = 2$, and $L = 1000$. Thus,

$$m = \frac{3(0.00056)}{2+2} 1000^{2-1} = 0.42$$

and the linear description is $D = 0.42v$, where D is in pounds and v is in ft/sec. This is the linear model that minimizes the integral of the squared error over $0 \leq v \leq 1000$ ft/sec.

C.2 MATLAB AND THE LEAST-SQUARES METHOD

We now show how to use MATLAB's `polyfit` function to fit polynomials and functions that can be transformed into polynomials. The `polyfit` function is based on the least-squares method. Its syntax is `p = polyfit(x, y, n)`. The function fits a polynomial of degree n to data described by the vectors x and y , where x is the independent variable. The result p is the row vector of length $n + 1$ that contains the polynomial coefficients in order of descending powers.

Fitting First and Second Degree Polynomials

EXAMPLE C.2.1

■ **Problem**

Use the `polyfit` function to find the first and second degree polynomials that fit the following data in the least-squares sense. Evaluate the quality of fit for each polynomial.

x	0	1	2	3	4	5	6	7	8	9	10
y	48	49	52	63	76	98	136	150	195	236	260

■ **Solution**

The following MATLAB program computes the polynomial coefficients.

```
% Enter the data.
x = (0:10);
y = [48, 49, 52, 63, 76, 98, 136, 150, 195, 236, 260];
% Fit a first-degree polynomial.
p_first = polyfit(x,y,1)
Fit a second-degree polynomial.
p_second = polyfit(x,y,2)
```

The polynomial coefficients of the first degree polynomial are contained in the vector `p_first`, and the coefficients of the second degree polynomial are contained in the vector `p_second`. The results are `p_first = [22.1909, 12.0455]`, which corresponds to the polynomial $y = 22.1909x + 12.0455$, and `p_second = [2.1993, 0.1979, 45.035]`, which corresponds to the polynomial $y = 2.1993x^2 + 0.1979x + 45.035$.

We can use MATLAB to plot the polynomials and to evaluate the “quality of fit” quantities J , S , and r^2 . The following script file does this.

```
% Enter the data and find the mean of y.
x = (0:10);
y = [48, 49, 52, 63, 76, 98, 136, 150, 195, 236, 260];
mu = mean(y);
% Define a range of x and y values for plotting.
xp = (0:0.01:10);
for k = 1:2
    yp(k,:) = polyval(polyfit(x,y,k),xp);
    % Compute J, S, and r squared.
    J(k) = sum((polyval(polyfit(x,y,k),x)-y).^2);
    S(k) = sum((polyval(polyfit(x,y,k),x)-mu).^2);
    r2(k) = 1-J(k)/S(k);
end
% Plot the first-degree polynomial.
subplot(2,1,1)
plot(xp,yp(1,:),x,y,'o'),axis([0 10 0 300]),xlabel('x'),...
    ylabel('y'),title('First-degree fit')
% Plot the second-degree polynomial.
subplot(2,1,2)
plot(xp,yp(2,:),x,y,'o'),axis([0 10 0 300]),xlabel('x'),...
    ylabel('y'),title('Second-degree fit')
% Display the computed values.
disp('The J values are:'),J
disp('The S values are:'),S
disp('The r^2 values are:'),r2
```

The polynomial coefficients in the above script file are contained in the vector `polyfit(x,y,k)`. If you need the polynomial coefficients, say for the second-degree polynomial, type `polyfit(x,y,2)` after the program has been run.

The plots are shown in Figure C.2.1. The following table gives the values of J , S , and r^2 for each polynomial.

Degree n	J	S	r^2
1	4348	54,168	0.9197
2	197.9	58,318	0.9997

Because the second-degree polynomial has the largest r^2 value, it represents the data better than the first-degree polynomial, according to the r^2 criterion. This is also obvious from the plots.

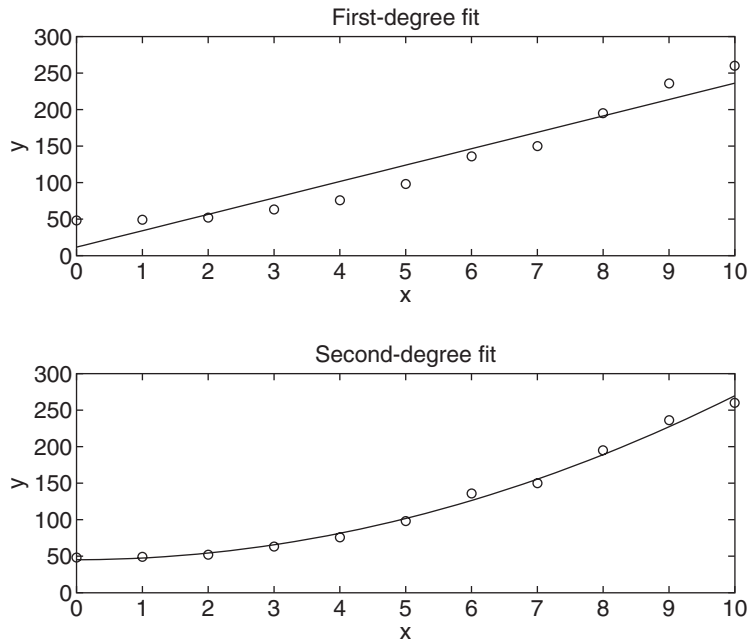


Figure C.2.1 Plots of first- and second-degree polynomial curve fits.

When we type `p = polyfit(z, w, 1)`, MATLAB will fit a linear function $w = p_1z + p_2$. The coefficients p_1 and p_2 are the first and second elements in the vector `p`; that is, `p` will be $[p_1, p_2]$. With a suitable transformation, the power and exponential functions can be transformed into a linear function, but the polynomial $w = p_1z + p_2$ has a different interpretation in each of the three cases.

The linear function: $y = mx + b$. In this case the variables w and z in the polynomial $w = p_1z + p_2$ are the original data variables, 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 $\log y = m \log x + \log b$, which has the form $w = p_1z + p_2$, where the polynomial variables w and z are related to the original data variables x and y by $w = \log y$ and $z = \log x$. 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 $\log b$. We can find b from $b = 10^{p_2}$.

The exponential function: $y = be^{mx}$. In this case, $\ln y = mx + \ln b$, which has the form $w = p_1z + p_2$, where the polynomial variables w and z are related to the original data variables x and y by $w = \ln y$ and $z = x$. Thus, we can find the exponential function that fits the data by typing `p = polyfit(x, log(y), 1)`. The first element p_1 of the vector `p` will be m , and the second element p_2 will be $\ln b$. We can find b from $b = e^{p_2}$.

Note

The notation for logarithms used by MATLAB is different than that used in mathematical expressions. Do not make the common mistake of using the MATLAB function `log` to represent the base-ten logarithm. The *natural* logarithm $\ln x$ is expressed in MATLAB by `log(x)`, whereas the *base-ten* logarithm $\log x$ is expressed as `log10(x)` in MATLAB.

Example C.2.2 illustrates how to use MATLAB to estimate the force-deflection characteristics of the cantilever support beam treated in Example 1.3.1.

EXAMPLE C.2.2**A Cantilever Beam Deflection Model****■ Problem**

The force-deflection data from Example 1.3.1 for the cantilever beam shown in Figure 1.3.1 is given in the following table.

Force f (lb)	0	100	200	300	400	500	600	700	800
Deflection x (in.)	0	0.15	0.23	0.35	0.37	0.5	0.57	0.68	0.77

Use MATLAB to obtain a linear relation between x and f , estimate the stiffness k of the beam, and evaluate the quality of the fit.

■ Solution

Note that here X is the *dependent* variable and f is the *independent* variable. In the following MATLAB script file the data are entered in the arrays x and f . The arrays x_p and f_p are created to plot the straight line at many points.

```
% Enter the data.
x = [0, 0.15, 0.23, 0.35, 0.37, 0.5, 0.57, 0.68, 0.77];
f = (0:100:800);
% Fit a first-degree polynomial.
p = polyfit(f,x,1)
% Compute the stiffness.
k = 1/p(1)
% Compute a set of f, x points.
fp = (0:800);
xp = p(1)*fp+p(2);
% Plot the fitted function and the data.
plot(fp,xp,f,x,'o'), xlabel('Applied Force f (lb)'), ...
    ylabel('Deflection x (in.)'), ...
    axis([0 800 0 0.8])
% Compute the J, S, and r squared values.
J = sum((polyval(p,f)-x).^2)
S = sum(x-mean(x)).^2)
r2 = 1 - J/S
```

The computed values in the array p are $p = [9.1667 \times 10^{-4}, 3.5556 \times 10^{-2}]$. Thus the fitted straight line is $x = 9.1667 \times 10^{-4}f + 3.5556 \times 10^{-2}$. Note that this line, which is shown in Figure C.2.2, does not pass through the origin as required, but it is close (it predicts that $x = 0.035556$ in. when $f = 0$). The quality-of-fit values are $J = 0.0048$, $S = 0.5090$, and $r^2 = 0.9906$, which indicates a very good fit.

Solving for f gives $f = (x - 3.5556 \times 10^{-2})/9.1667 \times 10^{-4} = 1091x - 38.7879$. The computed value of the stiffness k is the coefficient of x ; thus $k = 1091$ lb/in.

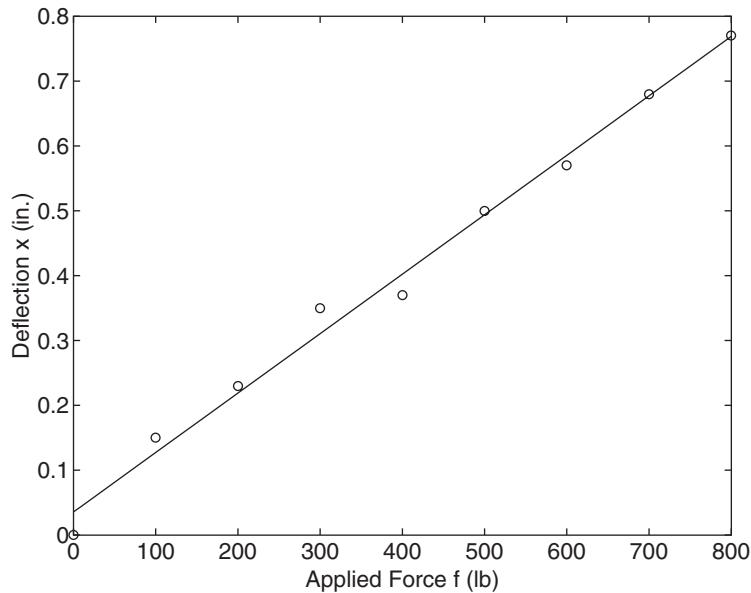


Figure C.2.2 Models of beam deflection. Unconstrained linear function.

Constraining the Curve Fit

EXAMPLE C.2.3

■ Problem

Use MATLAB to fit a straight line to the beam force-deflection data given in Example C.2.2, but constrain the line to pass through the origin.

■ Solution

We can apply (C.1.3), noting here that the measured variable is the deflection x and the independent variable is the force f . Thus (C.1.3) becomes

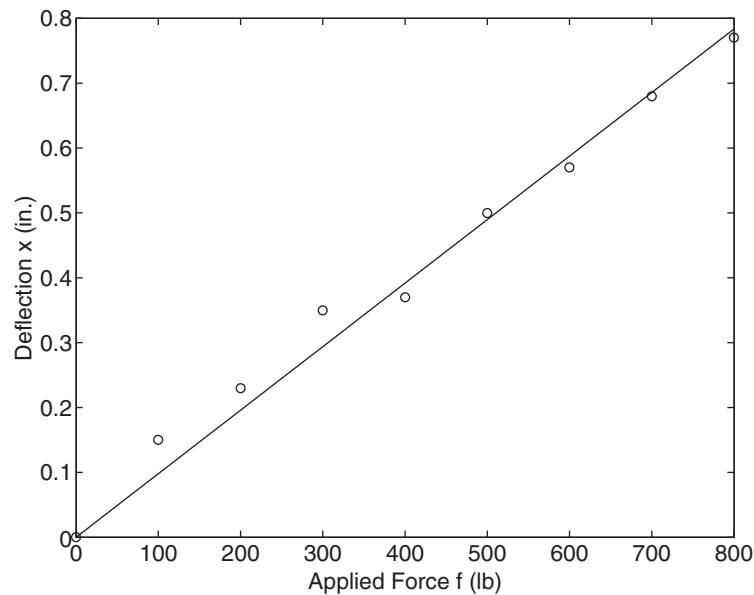
$$m \sum_{i=1}^n f_i^2 = \sum_{i=1}^n f_i x_i \quad (1)$$

The MATLAB program to solve this equation for m and k is

```
% Enter the data
x = [0, 0.15, 0.23, 0.35, 0.37, 0.5, 0.57, 0.68, 0.77];
f = (0:100:800);
% Compute m from (1).
m = sum(f.*x)/sum(f.^2);
% Compute the stiffness.
k = 1/m
% Compute J, S, and r squared.
J = sum((m*f-x).^2)
S = sum((x-mean(x)).^2)
r2 = 1 - J/S
```

The answer is $k = 1021$ lb/in. The corresponding line is shown in Figure C.2.3. The quality-of-fit values are $J = 0.0081$, $S = 0.5090$, and $r^2 = 0.9840$, which indicates a good fit.

Figure C.2.3 Linear function constrained to pass through the origin.



EXAMPLE C.2.4

Temperature Dynamics of Water

■ Problem

Consider again Example 1.4.1. Water in a glass measuring cup was allowed to cool after being heated to 204°F. The ambient air temperature was 70°F. The measured water temperature at various times is given in the following table.

Time (sec)	0	120	240	360	480	600
Temperature (°F)	204	191	178	169	160	153

Time (sec)	720	840	960	1080	1200
Temperature (°F)	147	141	137	132	127

Obtain a functional description of the water temperature versus time.

■ Solution

From Example 1.4.1, we learned that the relative temperature, $\Delta T = T - 70$ has the exponential form

$$\Delta T = be^{mt} \quad (1)$$

We can find values of m and b by using $p = \text{polyfit}(x, \log(y), 1)$. The first element p_1 of the vector p will be m , and the second element p_2 will be $\ln b$. We can find b from $b = e^{p_2}$. The following MATLAB program performs the calculations.

```
% Enter the data.
time = (0:120:1200);
temp = [204,191,178,169,160,153,147,141,137,132,127];
% Compute the relative temperature and its logarithm.
rel_temp = temp - 70;
log_rel_temp = log(rel_temp);
% Fit a first-degree polynomial.
```

```

p = polyfit(time,log_rel_temp,1);
% Compute m and b from the polynomial coefficients.
m = p(1),b = exp(p(2))
% Compute DT (delta T) from (1).
DT = b*exp(m*time);
% Compute J, S, and r squared.
J = sum((DT-rel_temp).^2)
S = sum((rel_temp - mean(rel_temp)).^2)
r2 = 1 - J/S

```

The results are $m = -6.9710 \times 10^{-4}$ and $b = 1.2916 \times 10^2$, and the corresponding function is

$$\Delta T = be^{mt} \quad \text{or} \quad T = \Delta T + 70 = be^{mt} + 70$$

The quality-of-fit values are $J = 47.4850$, $S = 6.2429 \times 10^3$, and $r^2 = 0.9924$, which indicates a very good fit.

Orifice Flow

EXAMPLE C.2.5

■ Problem

Consider again Example 1.4.2. A hole 6 mm in diameter was made in a translucent milk container (Figure 1.4.7). A series of marks 1 cm apart was made above the hole. While adjusting the tap flow to keep the water height constant, the time for the outflow to fill a 250 ml cup was measured (1 ml = 10^{-6} m³). This was repeated for several heights. The data are given in the following table.

Height h (cm)	11	10	9	8	7	6	5	4	3	2	1
Time t (s)	7	7.5	8	8.5	9	9.5	11	12	14	19	26

Obtain a functional description of the volume outflow rate f as a function of water height h above the hole.

■ Solution

First obtain the flow rate data in ml/s by dividing the 250 ml volume by the time to fill:

$$f = \frac{250}{t} \quad (1)$$

In Example 1.4.2, we learned that the following power function can describe the data:

$$f = bh^m \quad (2)$$

We can find the values of m and b by using `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 $\log b$. We can find b from $b = 10^{p_2}$. The following MATLAB program performs the calculations.

```

% Enter the data.
h = (1:11);
time = [26, 19, 14, 12, 11, 9.5, 9, 8.5, 8, 7.5, 7];
% Compute the flow rate from (1) and its logarithm.
flow = 250./time;
logflow = log10(flow); logheight = log10(h);
% Fit a first-degree polynomial.
p = polyfit(logheight, logflow, 1);
% Compute m and b from the polynomial coefficients.
m = p(1), b = 10^p(2)

```

```
% Compute f from (2).
f = b*h.^m;
% Compute J, S, and r squared.
J = sum((f - flow).^2)
S = sum((flow - mean(flow)).^2)
r2 = 1 - J/S
```

The results are $m = 0.5499$ and $b = 9.4956$, and the corresponding function is

$$f = bh^m = 9.4956h^{0.5499}$$

The quality-of-fit values are $J = 2.5011$, $S = 698.2203$, and $r^2 = 0.9964$, which indicates a very good fit.

Sometimes we know from physical theory that the data can be described by a power function with a specified exponent. For example, Torricelli's principle of hydraulic resistance states that the volume flow rate f of a liquid through a restriction is proportional to the square root of the pressure drop p across the restriction; that is, $q = c\sqrt{p} = cp^{1/2}$. In many applications, the pressure drop is due to the weight of liquid in a container. This is the case for water in the milk container of Example 1.4.2. In such situations, Torricelli's principle states that the flow rate is proportional to the square root of the height h of the liquid above the orifice. Thus,

$$f = b\sqrt{h} = bh^{1/2}$$

where b is a constant that must be determined from data.

EXAMPLE C.2.6**Orifice Flow with Constrained Exponent****■ Problem**

Consider the data of Example C.2.5. Determine the best-fit value of the coefficient b in the square-root function

$$f = bh^{1/2} \quad (1)$$

Height h (cm)	11	10	9	8	7	6	5	4	3	2	1
Time t (s)	7	7.5	8	8.5	9	9.5	11	12	14	19	26

■ Solution

First obtain the flow rate data in ml/s by dividing the 250 ml volume by the time to fill:

$$f = \frac{250}{t} \quad (2)$$

Referring to Example C.1.3, whose model is $y = bx^m$, we see here that $y = f$, $h = x$, and $m = 0.5$. From Equation (1) of Example C.1.3,

$$b = \frac{\sum_{i=1}^n h_i^{0.5} y_i}{\sum_{i=1}^n h_i} \quad (3)$$

The MATLAB program to carry out these calculations is shown next.

```
% Enter the data.
h = (1:11);
time = [26, 19, 14, 12, 11, 9.5, 9, 8.5, 8, 7.5, 7];
% Compute flow rate from (2).
flow = 250./time;
% Compute b from (3).
b = sum(sqrt(h).*flow)/sum(h)
% Compute f from (1).
f = b*sqrt(h);
% Compute J, S, and r squared.
J = sum((f - flow).^2)
S = sum((flow - mean(flow)).^2)
r2 = 1 - J/S
```

The result is $a = 10.4604$ and the flow model is $f = 10.4604\sqrt{h}$. The quality-of-fit values are $J = 5.5495$, $S = 698.2203$, and $r^2 = 0.9921$, which indicates a very good fit.
