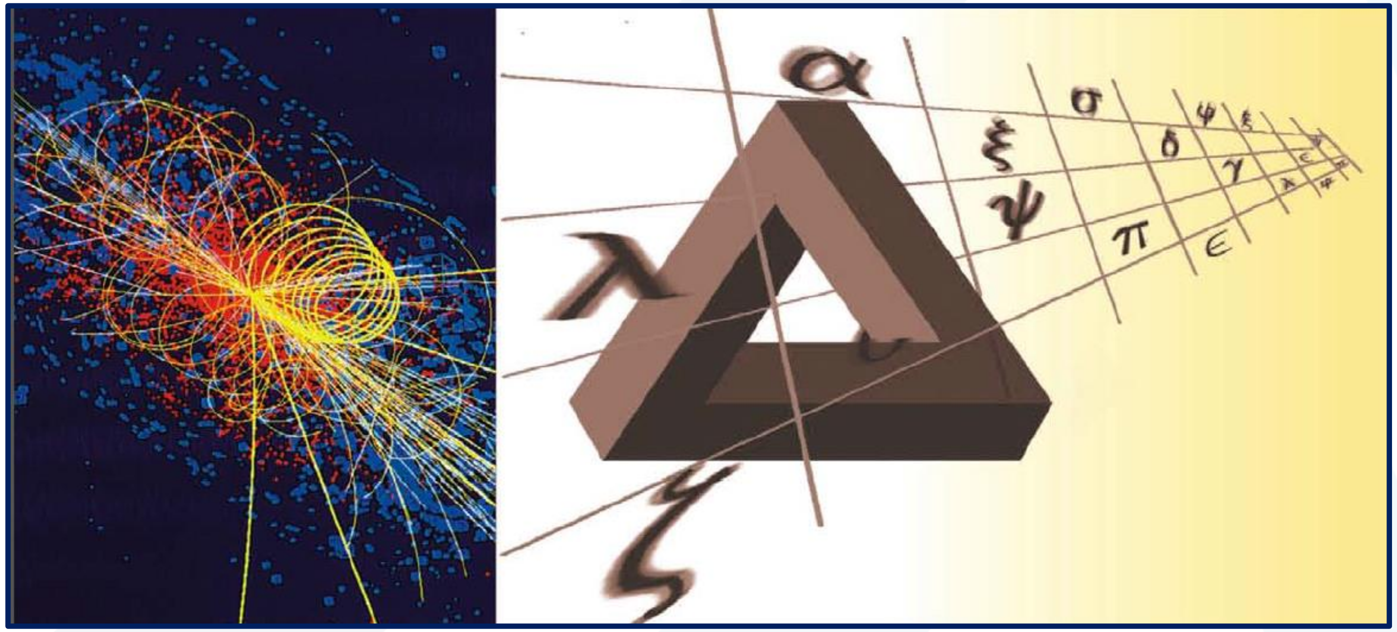
جامعة المنارة

كلية الهندسة

قسم الروبوتيك و الأنظمة الذكية

مقرر التحليل العددي

Curve Fitting



العام الدراسي 2023-2024

د. محمد خيرعبدالله محمد

جَـامعة
المَـنارة
MANARA UNIVERSITY

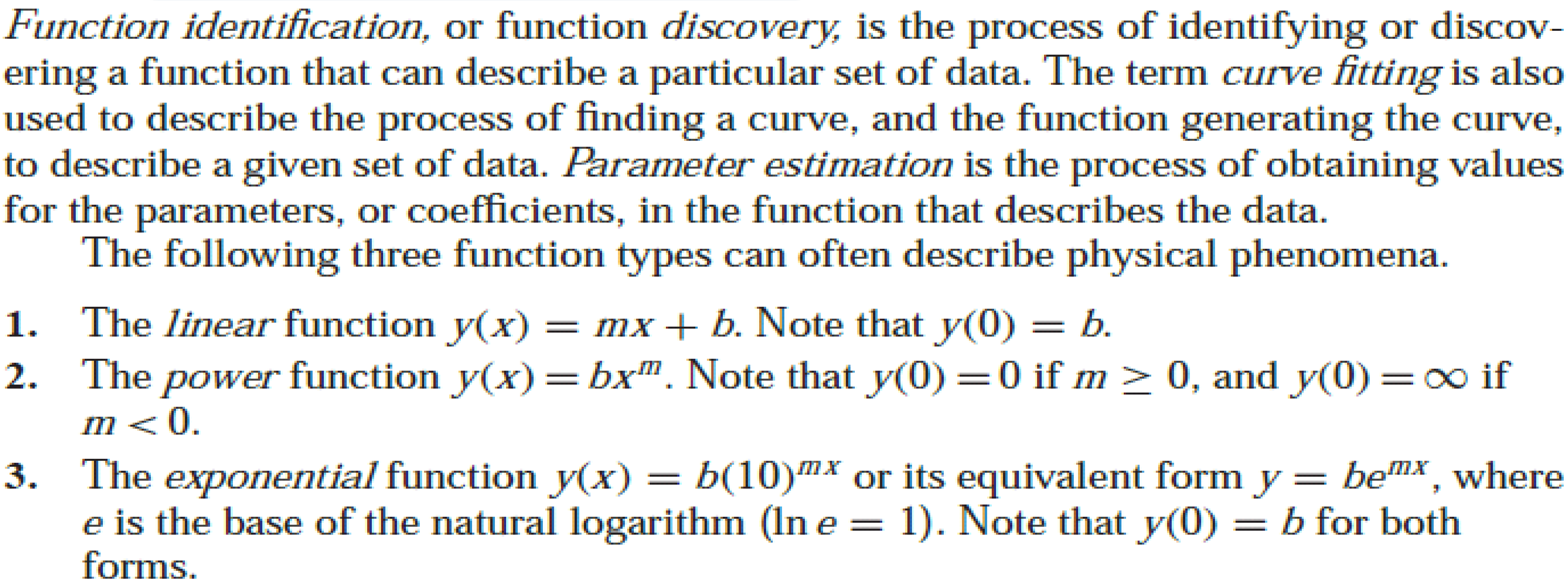
Contents

Function Identification and Parameter Estimation

Fitting Models to Scattered Data

The Least-Squares Method with Matlab

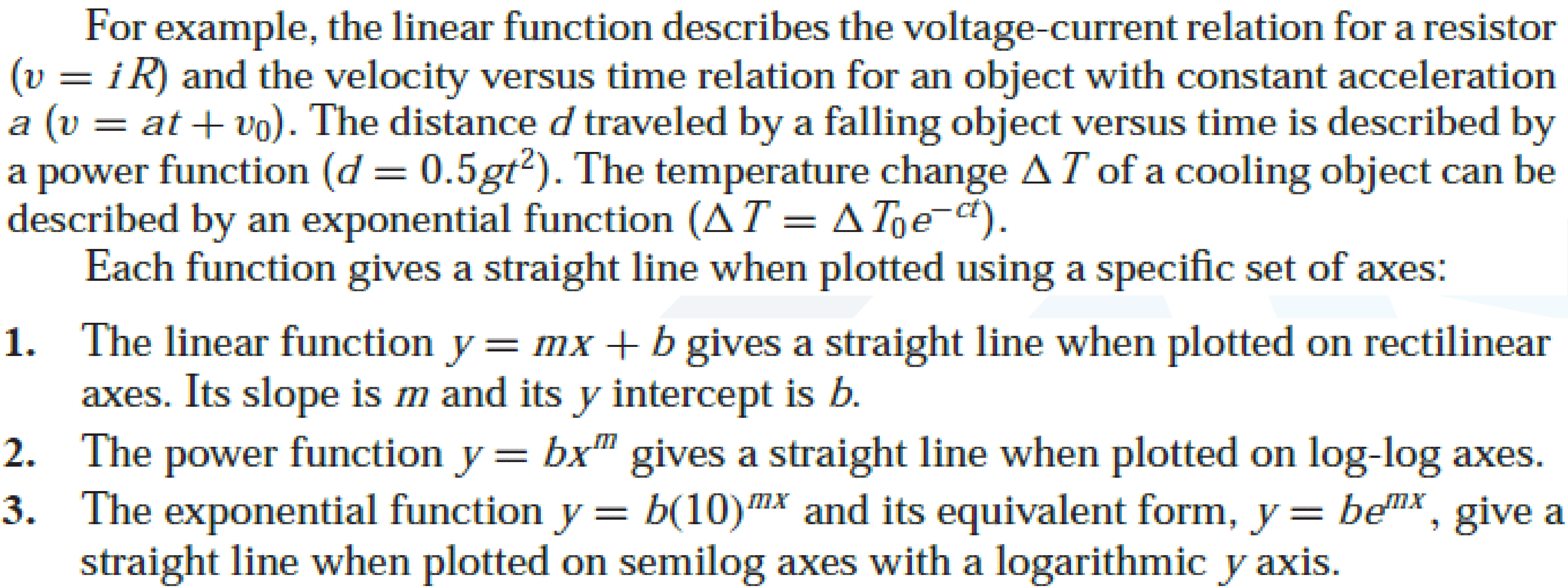
Function Identification and Parameter Estimation



Function identification, or function *discovery*, is the process of identifying or discovering a function that can describe a particular set of data. The term *curve fitting* is also used to describe the process of finding a curve, and the function generating the curve, to describe a given set of data. *Parameter estimation* is the process of obtaining values for the parameters, or coefficients, in the function that describes the data.

The following three function types can often describe physical phenomena.

1. The *linear* function $y(x) = mx + b$. Note that $y(0) = b$.
2. The *power* function $y(x) = bx^m$. Note that $y(0) = 0$ if $m \geq 0$, and $y(0) = \infty$ if $m < 0$.
3. The *exponential* function $y(x) = b(10)^{mx}$ or its equivalent form $y = be^{mx}$, where e is the base of the natural logarithm ($\ln e = 1$). Note that $y(0) = b$ for both forms.

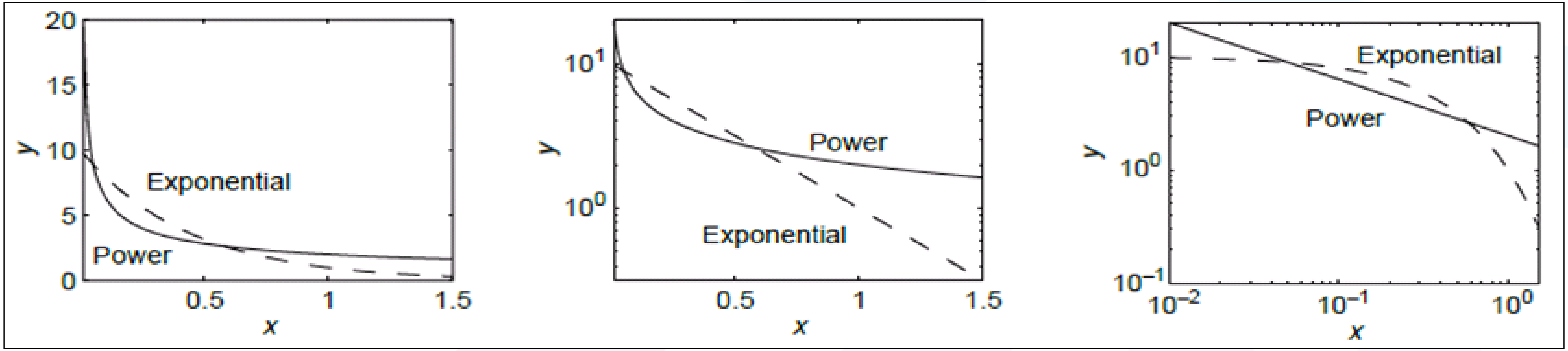


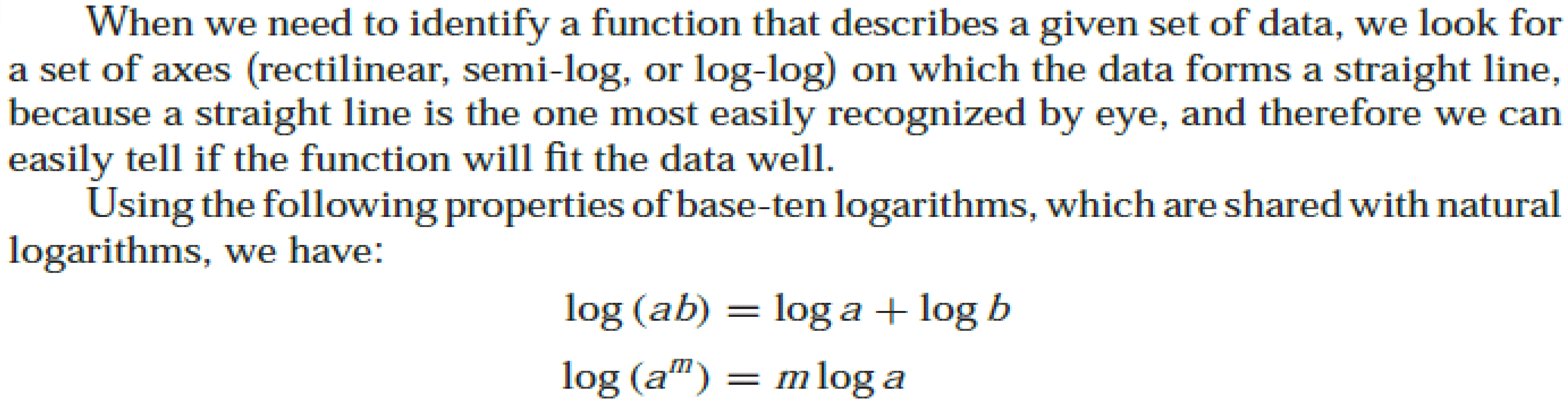
For example, the linear function describes the voltage-current relation for a resistor ($v = iR$) and the velocity versus time relation for an object with constant acceleration a ($v = at + v_0$). The distance d traveled by a falling object versus time is described by a power function ($d = 0.5gt^2$). The temperature change ΔT of a cooling object can be described by an exponential function ($\Delta T = \Delta T_0 e^{-ct}$).

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 y 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 semilog axes with a logarithmic y axis.

These properties of the power and exponential functions are illustrated in Figure , which shows the power function $y = 2x^{-0.5}$ and the exponential function $y = 10(10^{-x})$.



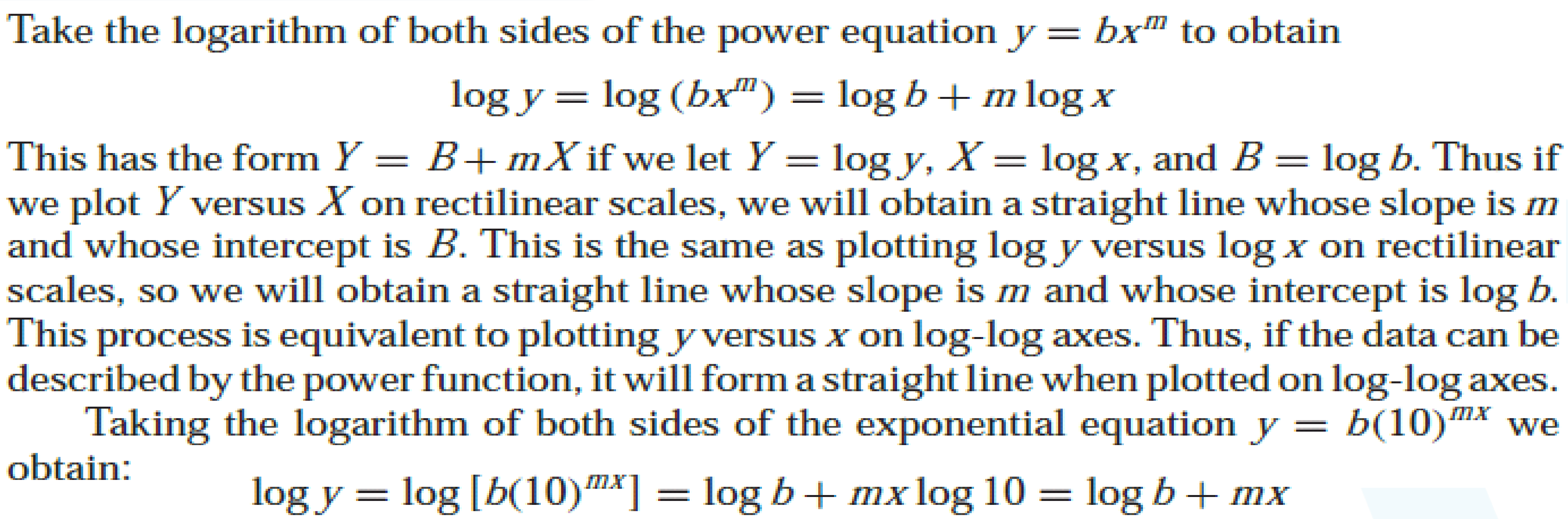


When we need to identify a function that describes a given set of data, we look for a set of axes (rectilinear, semi-log, or log-log) on which the data forms a straight line, because a straight line is the one most easily recognized by eye, and therefore we can easily tell if the function will fit the data well.

Using the following properties of base-ten logarithms, which are shared with natural logarithms, we have:

$$\log(ab) = \log a + \log b$$

$$\log(a^m) = m \log a$$



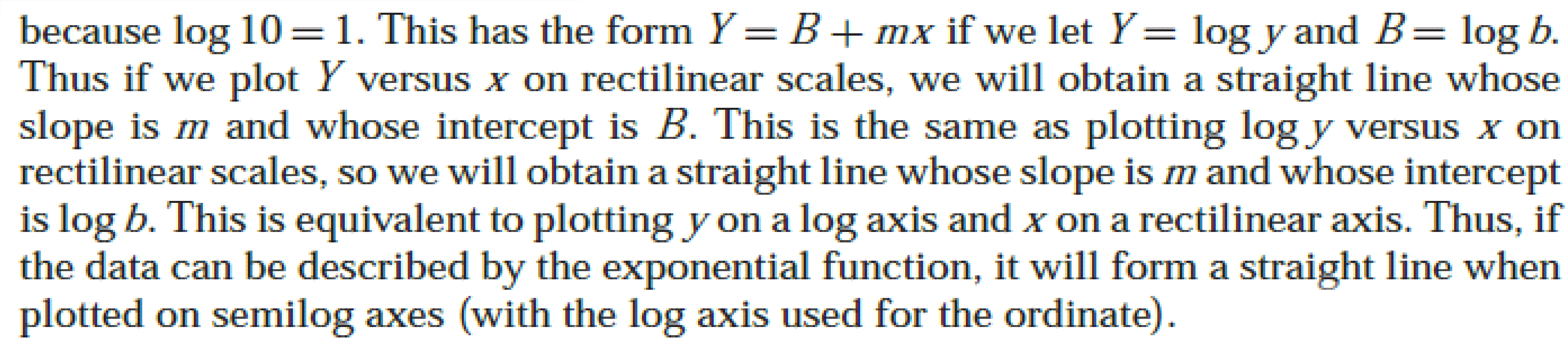
Take the logarithm of both sides of the power equation $y = bx^m$ to obtain

$$\log y = \log(bx^m) = \log b + m \log x$$

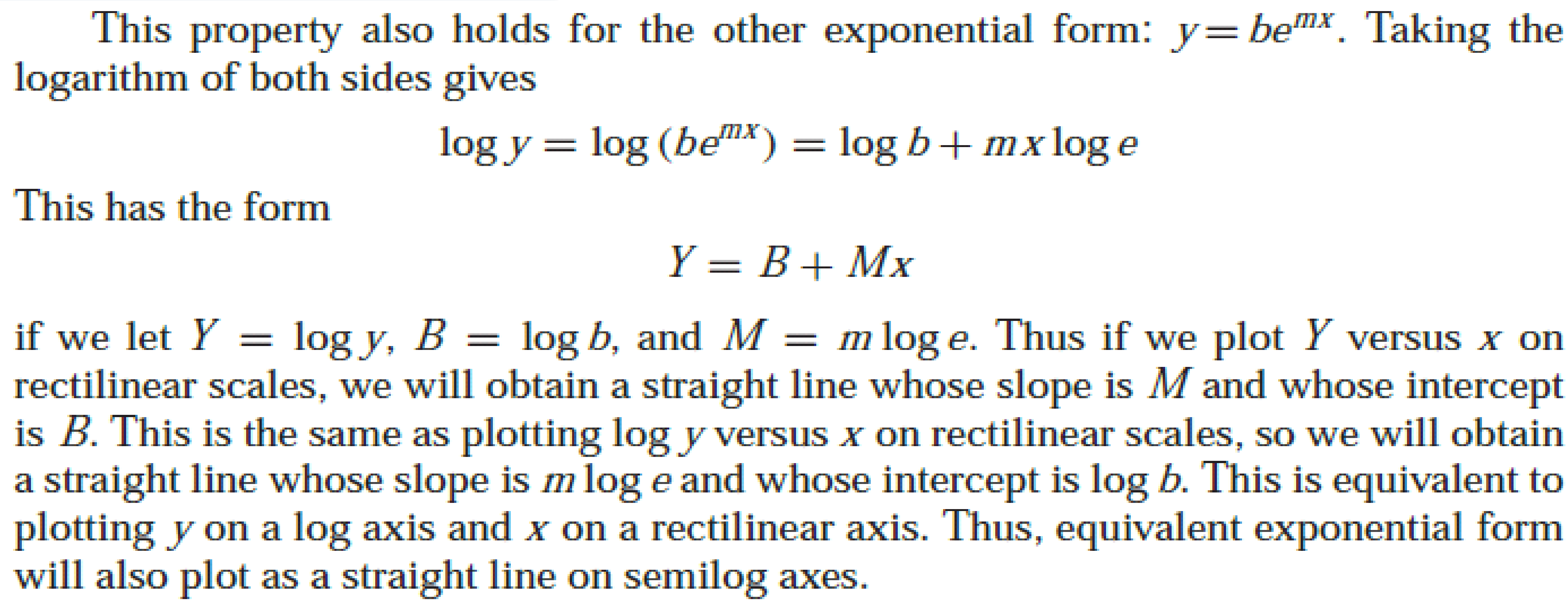
This has the form $Y = B + mX$ if we let $Y = \log y$, $X = \log x$, and $B = \log b$. Thus if we plot Y versus X on rectilinear scales, we will obtain a straight line whose slope is m and whose intercept is B. This is the same as plotting $\log y$ versus $\log x$ on rectilinear scales, so we will obtain a straight line whose slope is m and whose intercept is $\log b$. This process is equivalent to plotting y versus x on log-log axes. Thus, if the data can be described by the power function, it will form a straight line when plotted on log-log axes.

Taking the logarithm of both sides of the exponential equation $y = b(10)^{mx}$ we obtain:

$$\log y = \log[b(10)^{mx}] = \log b + mx \log 10 = \log b + mx$$



because $\log 10 = 1$. This has the form $Y = B + mx$ if we let $Y = \log y$ and $B = \log b$. Thus if we plot Y versus x on rectilinear scales, we will obtain a straight line whose slope is m and whose intercept is B. This is the same as plotting $\log y$ versus x on rectilinear scales, so we will obtain a straight line whose slope is m and whose intercept is $\log b$. This is equivalent to plotting y on a log axis and x on a rectilinear axis. Thus, if the data can be described by the exponential function, it will form a straight line when plotted on semilog axes (with the log axis used for the ordinate).



This property also holds for the other exponential form: $y = be^{mx}$. Taking the logarithm of both sides gives

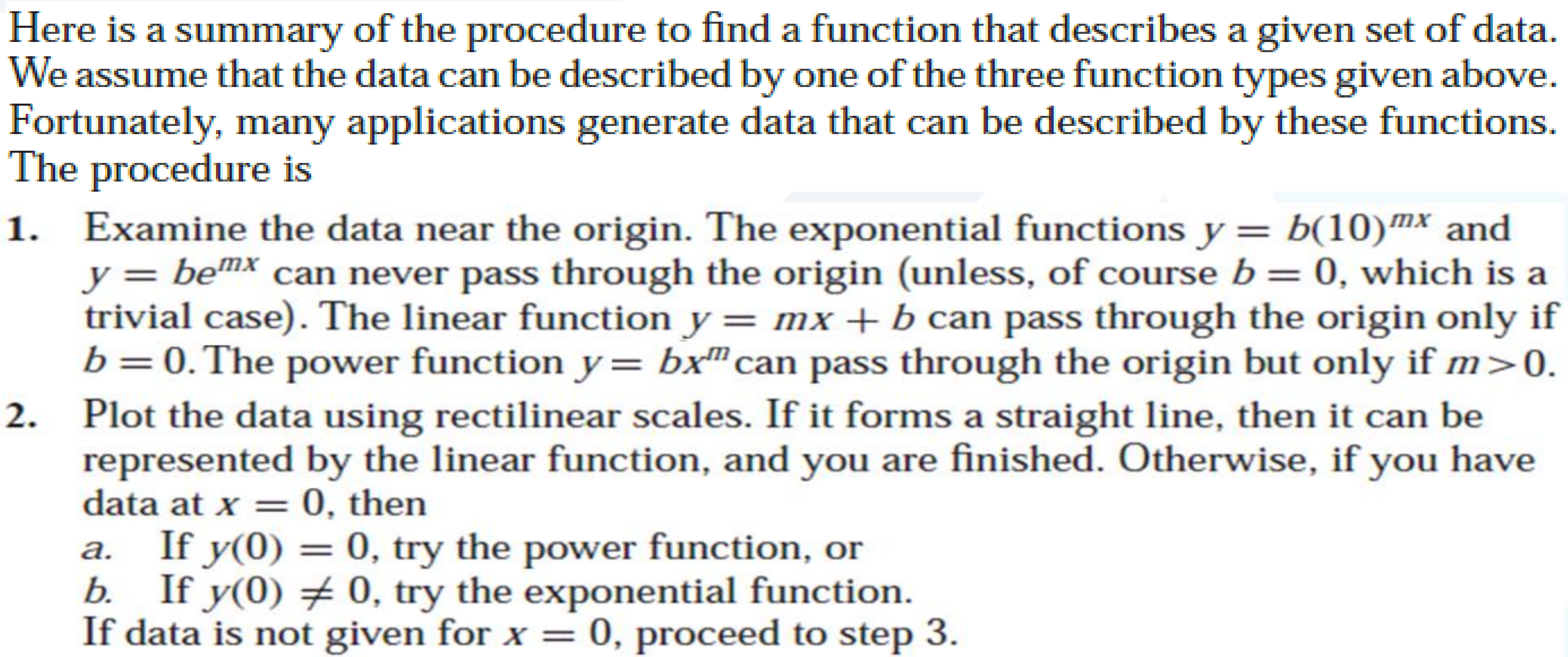
$$\log y = \log\left(be^{mx}\right) = \log b + mx \log e$$

This has the form

$$Y = B + Mx$$

if we let $Y = \log y$, $B = \log b$, and $M = m \log e$. Thus if we plot Y versus x on rectilinear scales, we will obtain a straight line whose slope is M and whose intercept is B. This is the same as plotting $\log y$ versus x on rectilinear scales, so we will obtain a straight line whose slope is $m \log e$ and whose intercept is $\log b$. This is equivalent to plotting y on a log axis and x on a rectilinear axis. Thus, equivalent exponential form will also plot as a straight line on semilog axes.

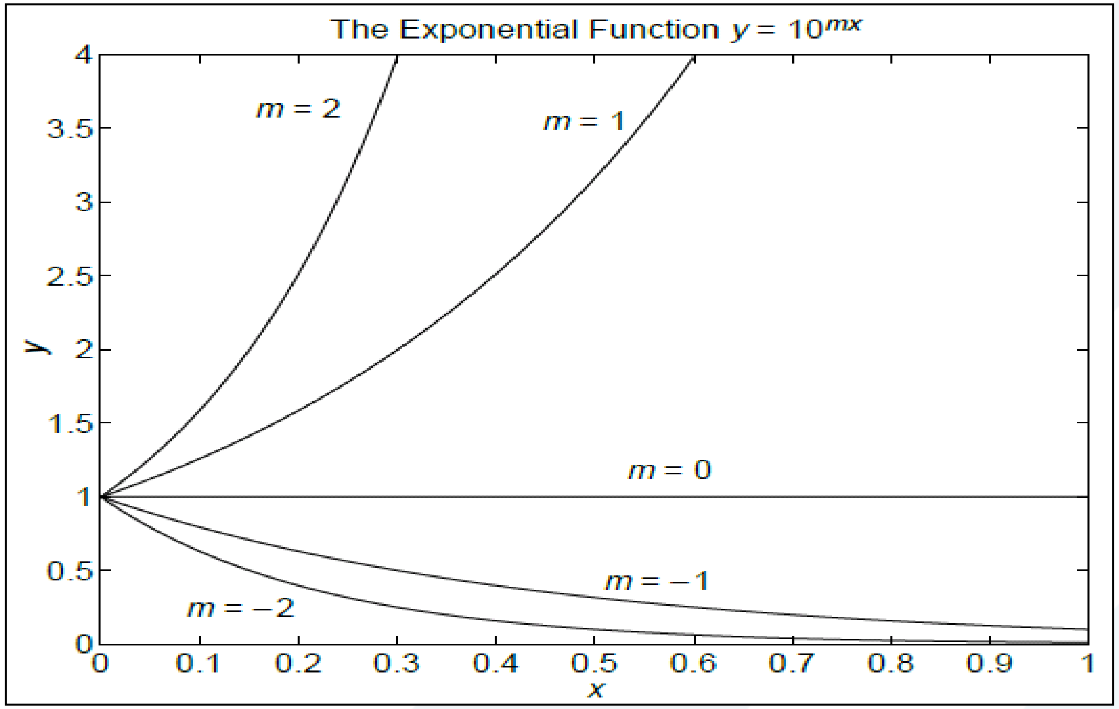
Steps for Function Identification

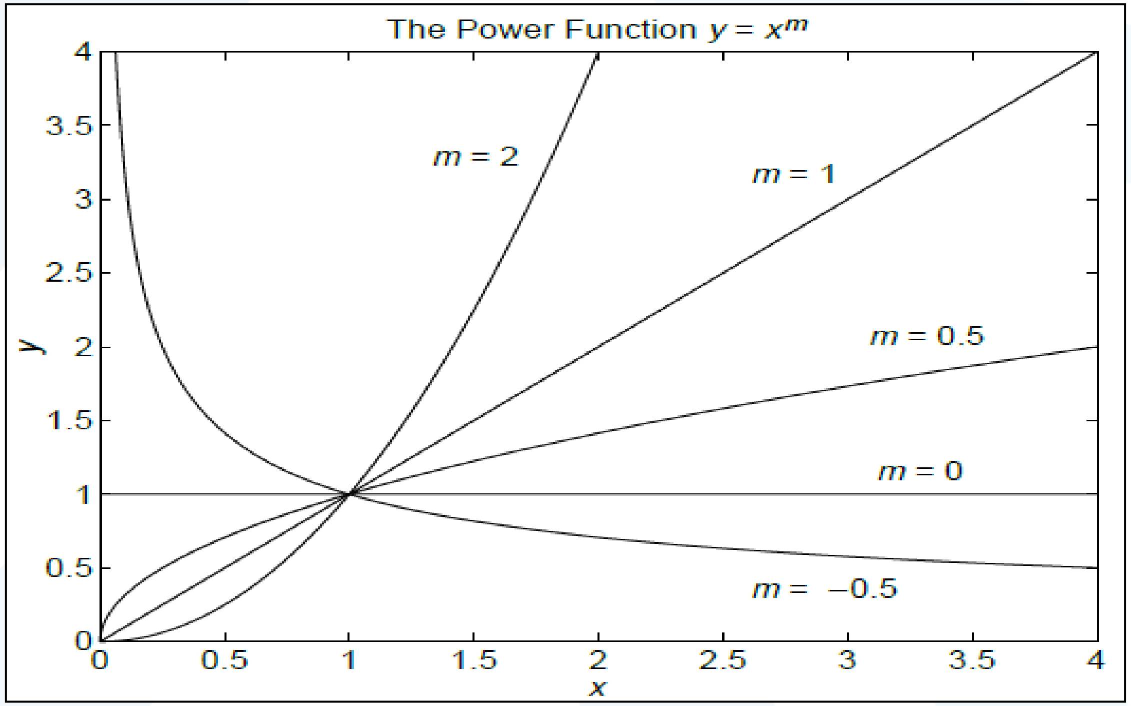


Here is a summary of the procedure to find a function that describes a given set of data. We assume that the data can be described by one of the three function types given above. Fortunately, many applications generate data that can be described by these functions. The procedure is

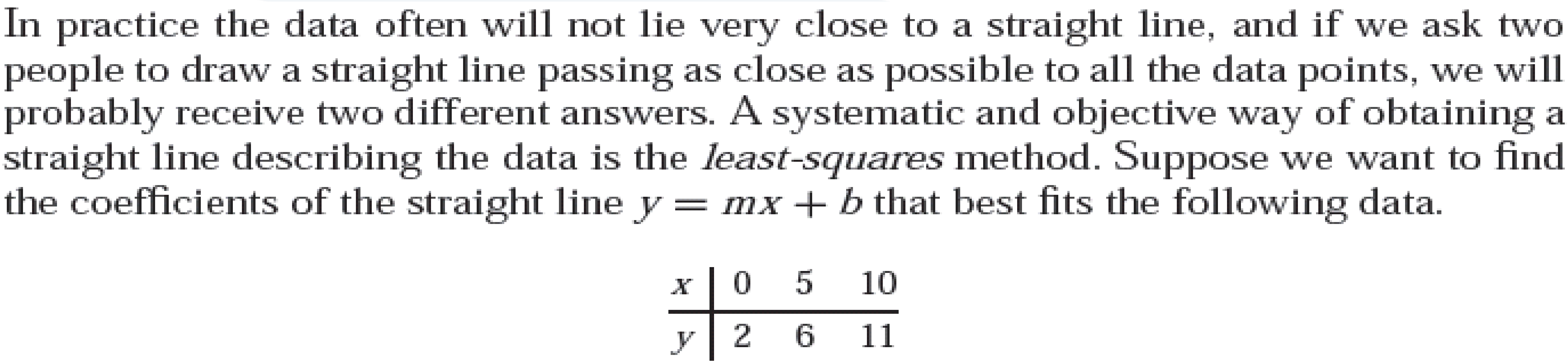
1. Examine the data near the origin. The exponential functions $y = b(10)^{mx}$ and $y = be^{mx}$ can never pass through the origin (unless, of course $b = 0$, which is a trivial case). The linear function $y = mx + b$ can pass through the origin only if $b = 0$. The power function $y = bx^m$ can pass through the origin but only if $m > 0$.
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, or
   b. If $y(0) \neq 0$, try the exponential function.
   If data is not given for $x = 0$, proceed to step 3.

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



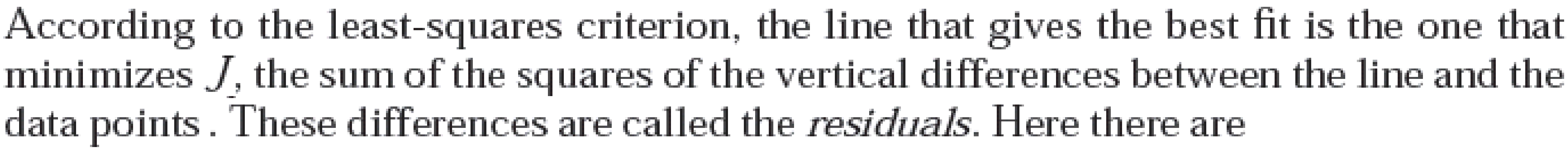


Fitting Models to Scattered Data

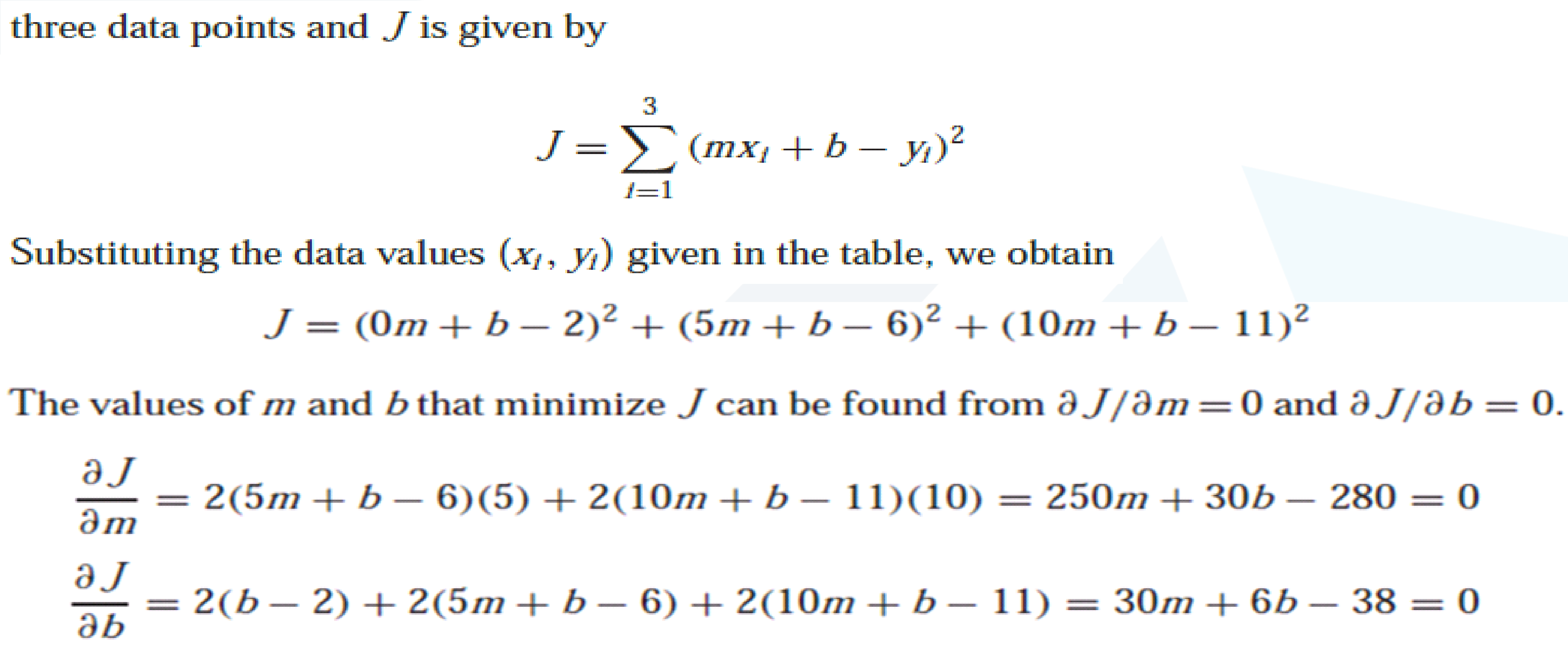


In practice the data often will not lie very close to a straight line, and if we ask two people to draw a straight line passing as close as possible to all the data points, we will probably receive two different answers. A systematic and objective way of obtaining a straight line describing the data 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 . These differences are called the *residuals*. Here there are



three data points and J is given by

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

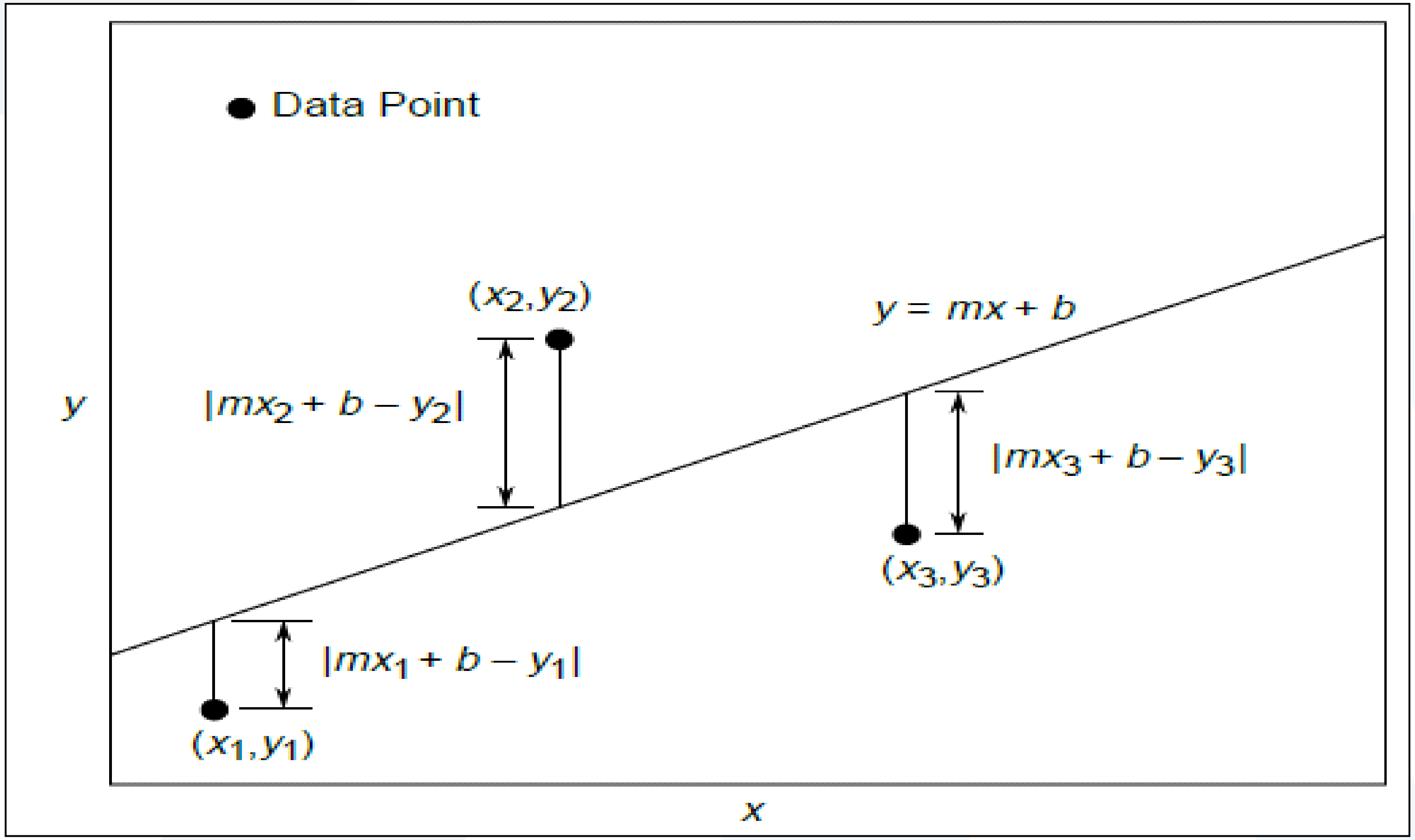
Substituting the data values (x_i, y_i) given in the table, we obtain

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

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

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

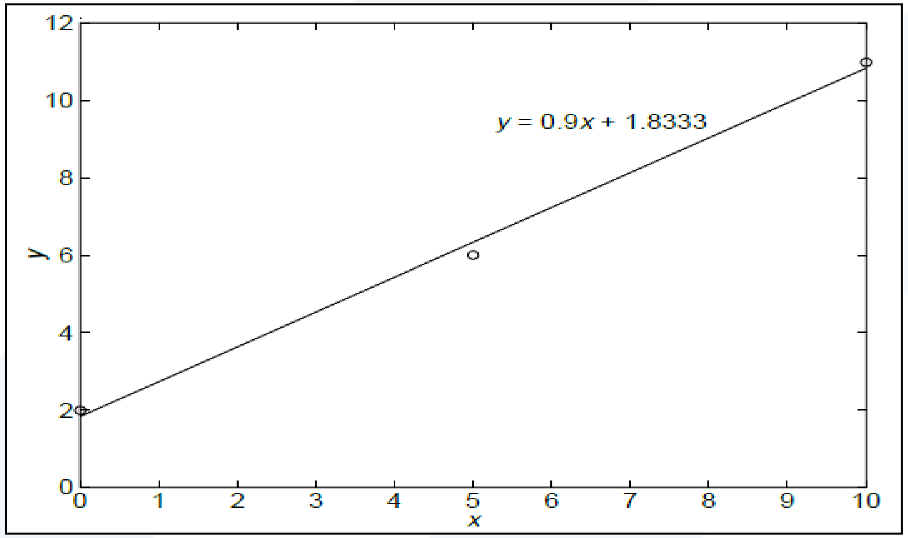
$$\frac{\partial J}{\partial b} = 2(b - 2) + 2(5m + b - 6) + 2(10m + b - 11) = 30m + 6b - 38 = 0$$


● Data Point
(x_2,y_2)
$y = mx + b$
y
$|mx_2 + b - y_2|$
$|mx_3 + b - y_3|$
(x_3,y_3)
$|mx_1 + b - y_1|$
(x_1,y_1)
x

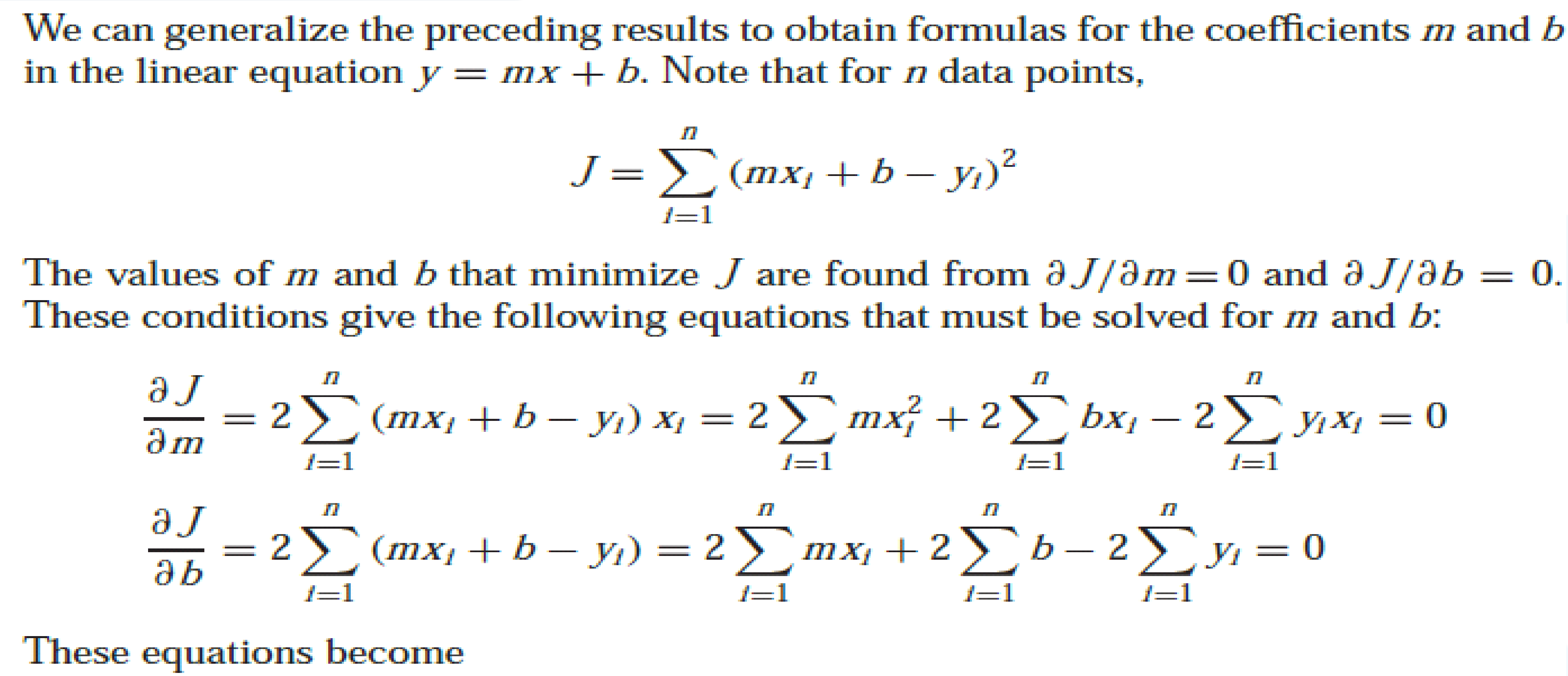
These conditions give the following equations that must be solved for the two unknowns m and b.

$$250m + 30b = 280$$
$$30m + 6b = 38$$

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 along with the data.



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,

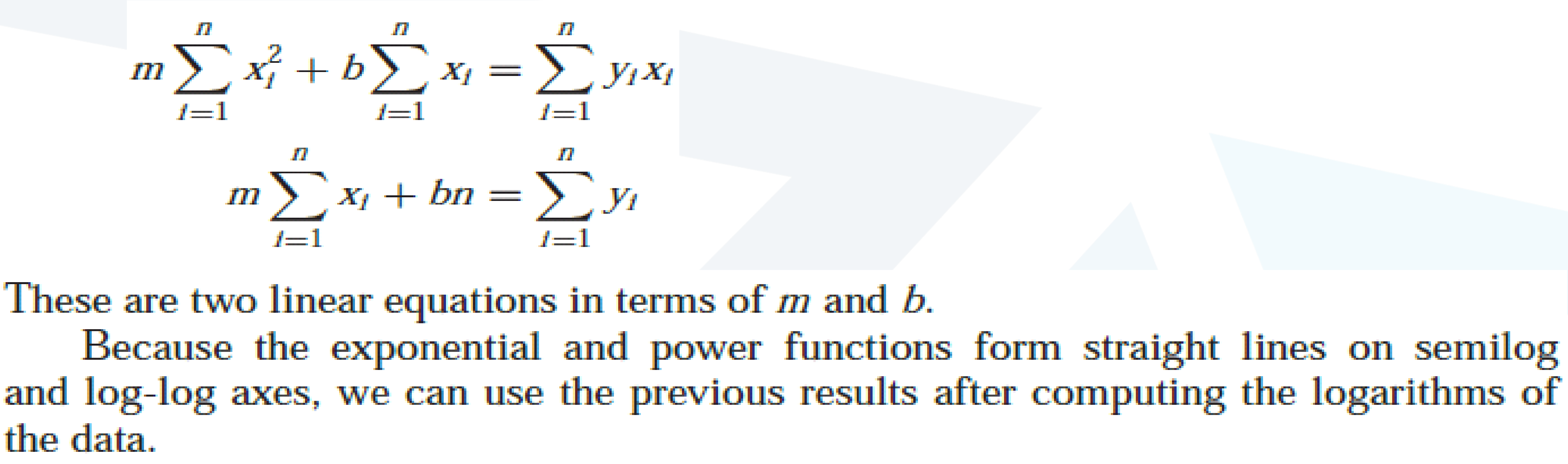
$$J = \sum_{i=1}^{n} (mx_i + b - y_i)^2$$

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



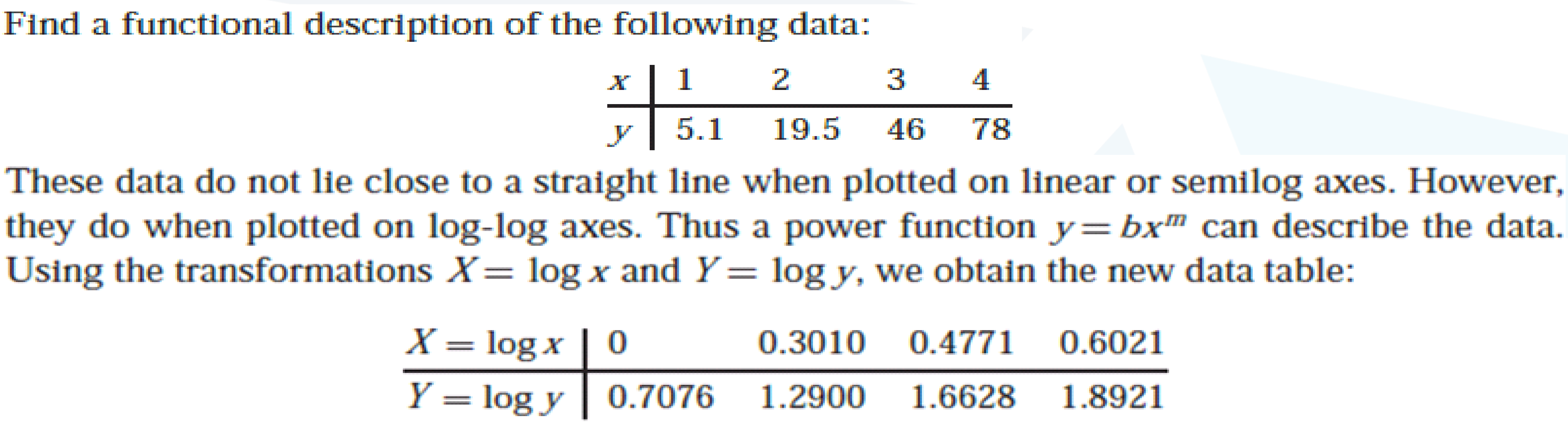
$$m\sum_{i=1}^{n} x_i^2 + b\sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i x_i$$

$$m\sum_{i=1}^{n} x_i + bn = \sum_{i=1}^{n} y_i$$

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, we can use the previous results after computing the logarithms of the data.

Example



Find a functional description of the following data:

x	1	2	3	4
y	5.1	19.5	46	78

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

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

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

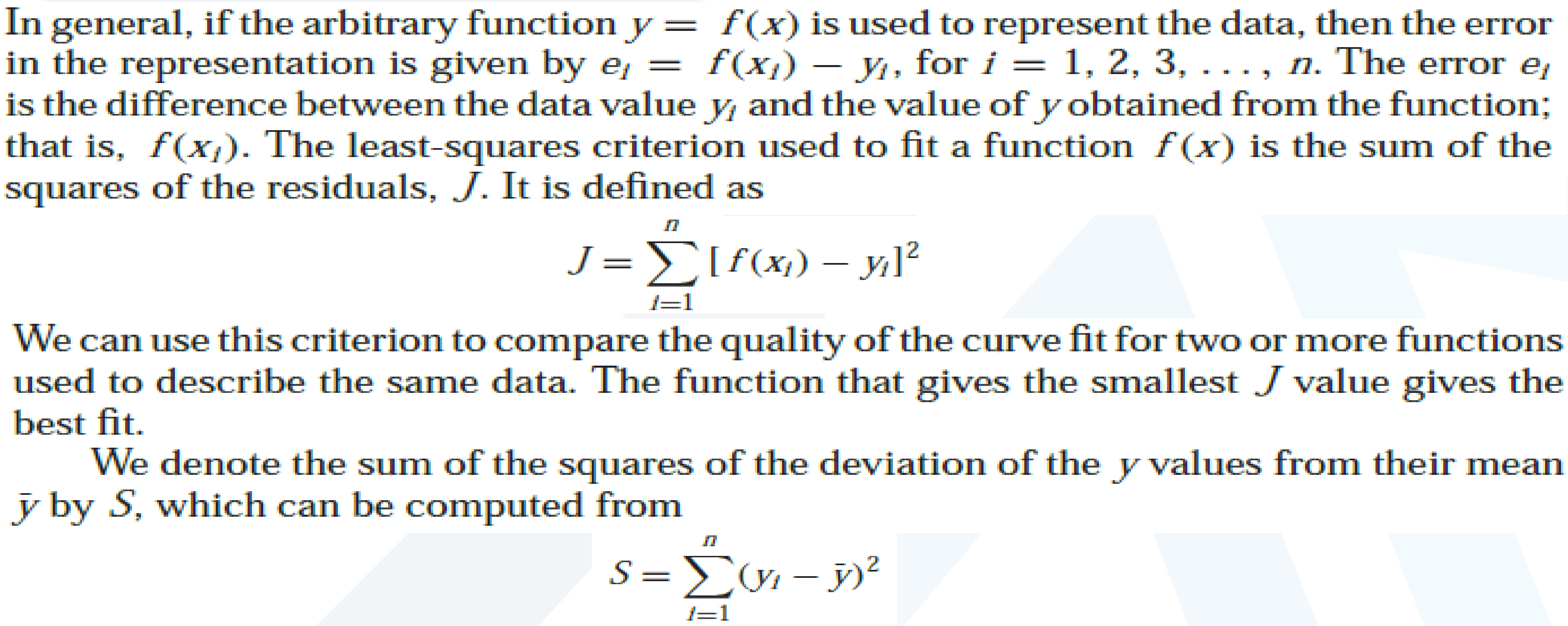
Using X, Y, and $B = \log b$ instead of x, y, and b we obtain

$$0.6807m + 1.3803\,B = 2.3208$$

$$1.3803m + 4\,B = 5.5525$$

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

The Quality of Curve Fit



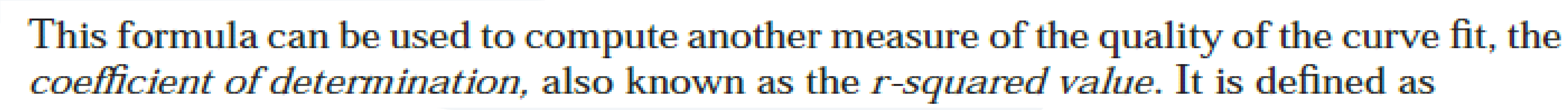
In general, if the arbitrary function $y = f(x)$ is used to represent the data, then the error in the representation is given by $e_i = f(x_i) - y_i$, for $i = 1, 2, 3, \ldots, n$. The error e_i is the difference between the data value y_i and the value of y obtained from the function; that is, $f(x_i)$.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

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

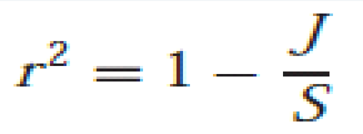
We can use this criterion to compare the quality of the curve fit for two or more functions used to describe the same data. The function that gives the smallest J value gives the best fit.

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

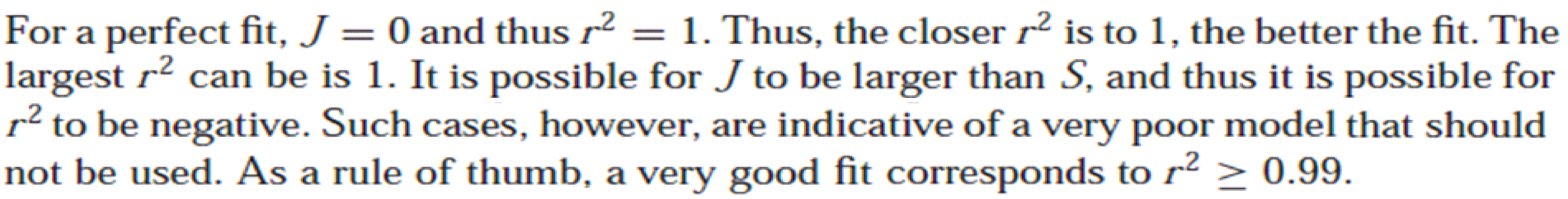
$$S = \sum_{i=1}^{n} (y_i - \bar{y})^2$$



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

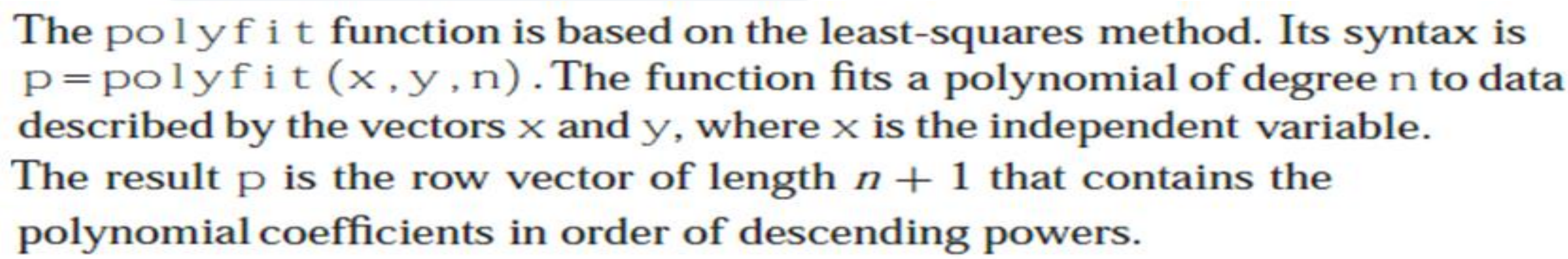


$$r^2 = 1 - \frac{J}{S}$$



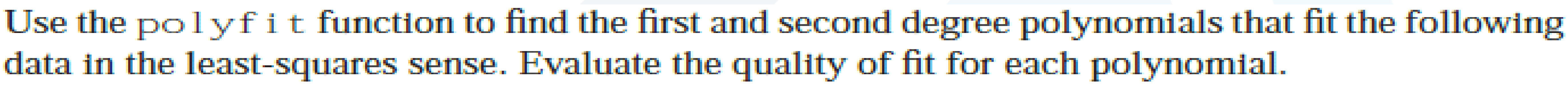
For a perfect fit, $J = 0$ and thus $r^2 = 1$. Thus, the closer r^2 is to 1, the better the fit. The largest r^2 can be is 1. It is possible for J to be larger than S, and thus it is possible for r^2 to be negative. Such cases, however, are indicative of a very poor model that should not be used. As a rule of thumb, a very good fit corresponds to $r^2 \geq 0.99$.

The Least-Squares Method with Matlab

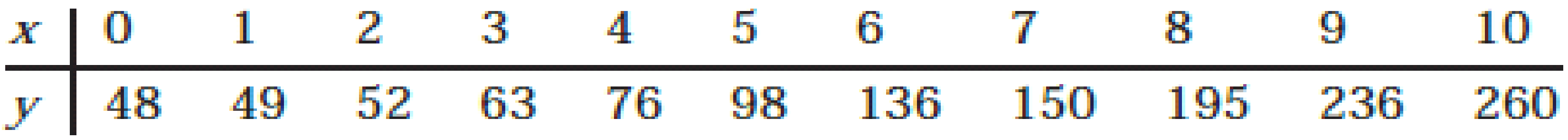


The polyfit function is based on the least-squares method. Its syntax is p=polyfit(x,y,n). The function fits a polynomial of degree n to data described by the vectors x and y, where x is the independent variable. The result p is the row vector of length $n + 1$ that contains the polynomial coefficients in order of descending powers.

Example



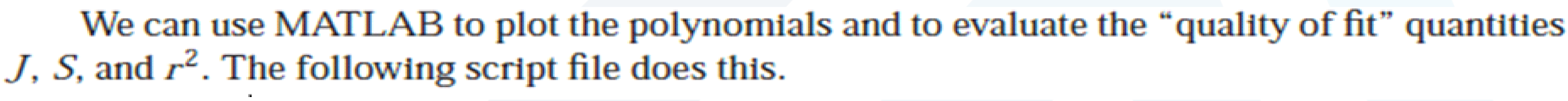
Use the polyfit function to find the first and second degree polynomials that fit the following data in the least-squares sense. Evaluate the quality of fit for each polynomial.



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

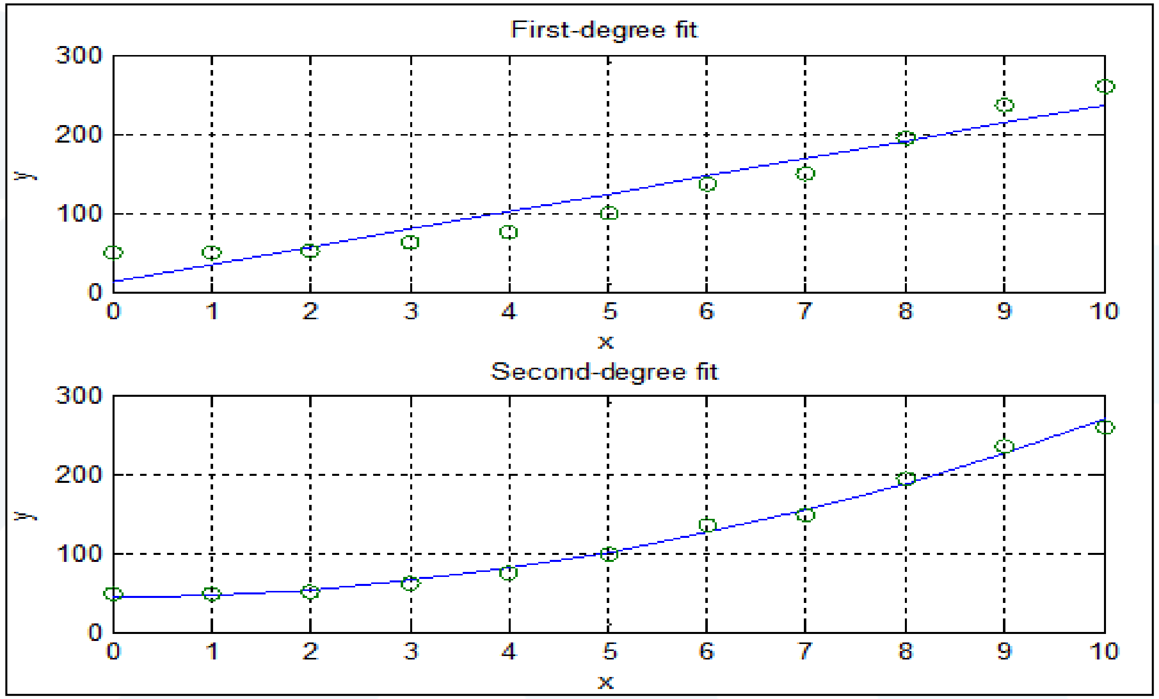
```
x = (0:10);
y = [48, 49, 52, 63, 76, 98, 136, 150, 195, 236, 260];
p_first = polyfit(x,y,1)
p_second = polyfit(x,y,2)
```

```
p_first =
  22.4636   11.5909
p_second =
   2.2343    0.1210   45.1049
```



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.

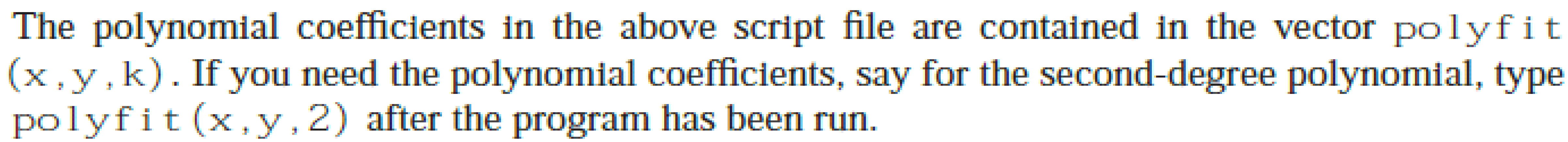
```
x = (0:10);
y = [48, 49, 52, 63, 76, 98, 136, 150, 195, 236, 260];
mu = mean(y);
for k = 1:2
     yp(k,:) = polyval(polyfit(x,y,k),x);
     J(k) = sum((polyval(polyfit(x,y,k),x)-y).^2);
     S(k) = sum((y- mu).^2);
     r2(k) = 1-J(k)/S(k);
end
subplot(2,1,1)
plot(x,yp(1,:),x,y,'o'),axis([0 10 0 300]),xlabel('x'),...
grid
ylabel('y'),title('First-degree fit')
subplot(2,1,2)
plot(x,yp(2,:),x,y,'o'),axis([0 10 0 300]),xlabel('x'),...
grid
ylabel('y'),title('Second-degree fit')
disp('The J values are:'),J
disp('The S values are:'),S
disp('The r^2 values are:'),r2
```



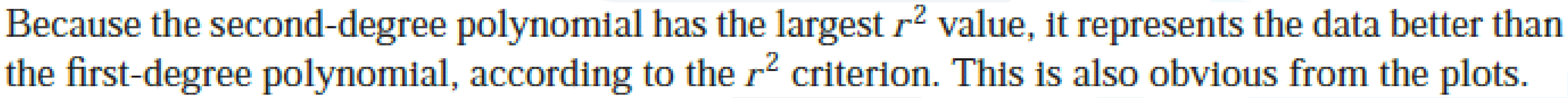
```
The J values are:
J =
   1.0e+03 *
    4.6793    0.3962
```

```
The S values are:
S =
   1.0e+04 *
    6.0187    6.0187
```

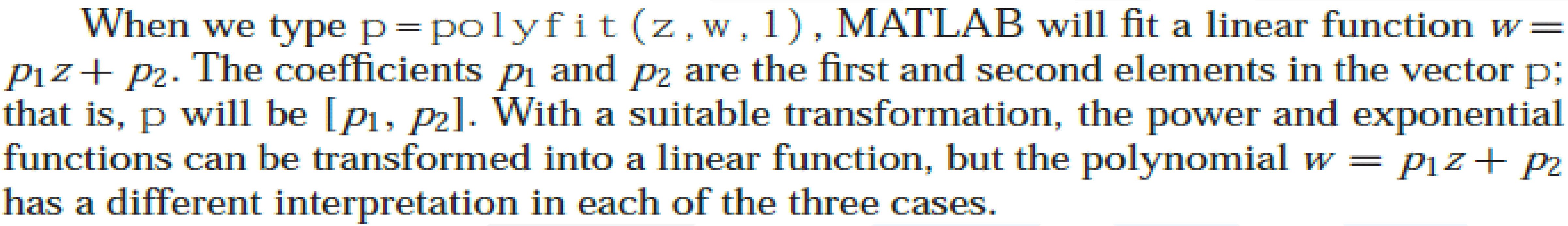
```
The r^2 values are:
r2 =
    0.9223    0.9934
```



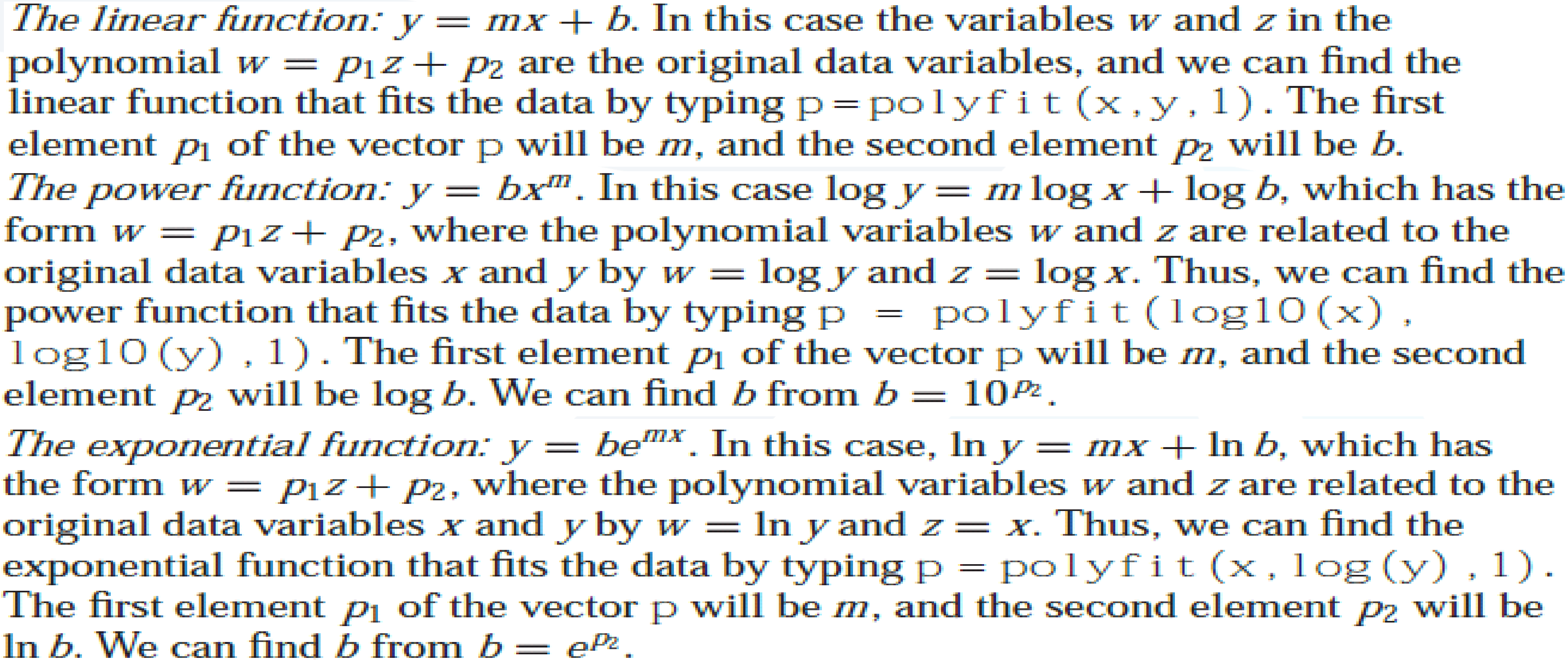
The polynomial coefficients in the above script file are contained in the vector `polyfit(x,y,k)`. If you need the polynomial coefficients, say for the second-degree polynomial, type `polyfit(x,y,2)` after the program has been run.



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.



When we type `p=polyfit(z,w,1)`, MATLAB will fit a linear function $w = p_1 z + p_2$. 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 $w = p_1 z + p_2$ 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 $w = p_1 z + p_2$ 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 $w = p_1 z + p_2$, 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 $w = p_1 z + p_2$, 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 $b = e^{p_2}$.

Example

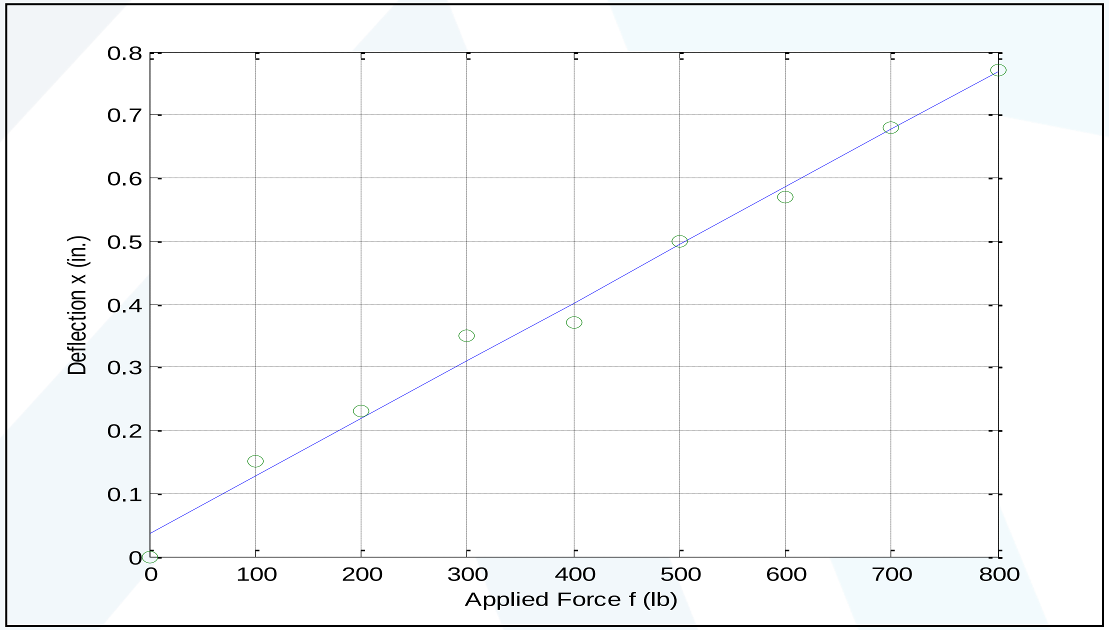
The force-deflection data for the cantilever beam is given in the following table.

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

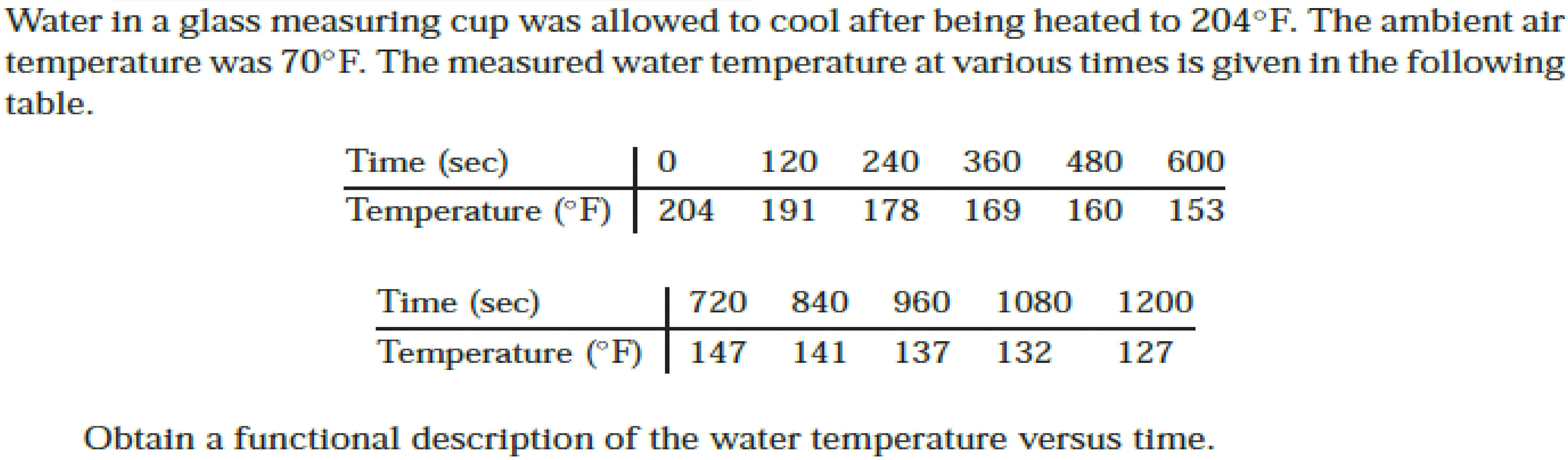
Use MATLAB to obtain a linear relation between x and f, estimate the stiffness k of the beam, and evaluate the quality of the fit.

```
x = [0, 0.15, 0.23, 0.35, 0.37, 0.5, 0.57, 0.68, 0.77];
f = (0:100:800);
p = polyfit(f,x,1)
k = 1/p(1)
xp = p(1)*f+p(2);
plot(f,xp,f,x,'o'),
grid
xlabel('Applied Force f (lb)'), ...
ylabel('Deflection x (in.)'), ...
axis([0 800 0 0.8])
J = sum((xp-x).^2)
S = sum((x-mean(x)).^2)
r2 = 1 - J/S
```

```
p =
    0.0009    0.0356
k =
   1.0909e+03
J =
    0.0048
S =
    0.5090
r2 =
    0.9906
```



Example



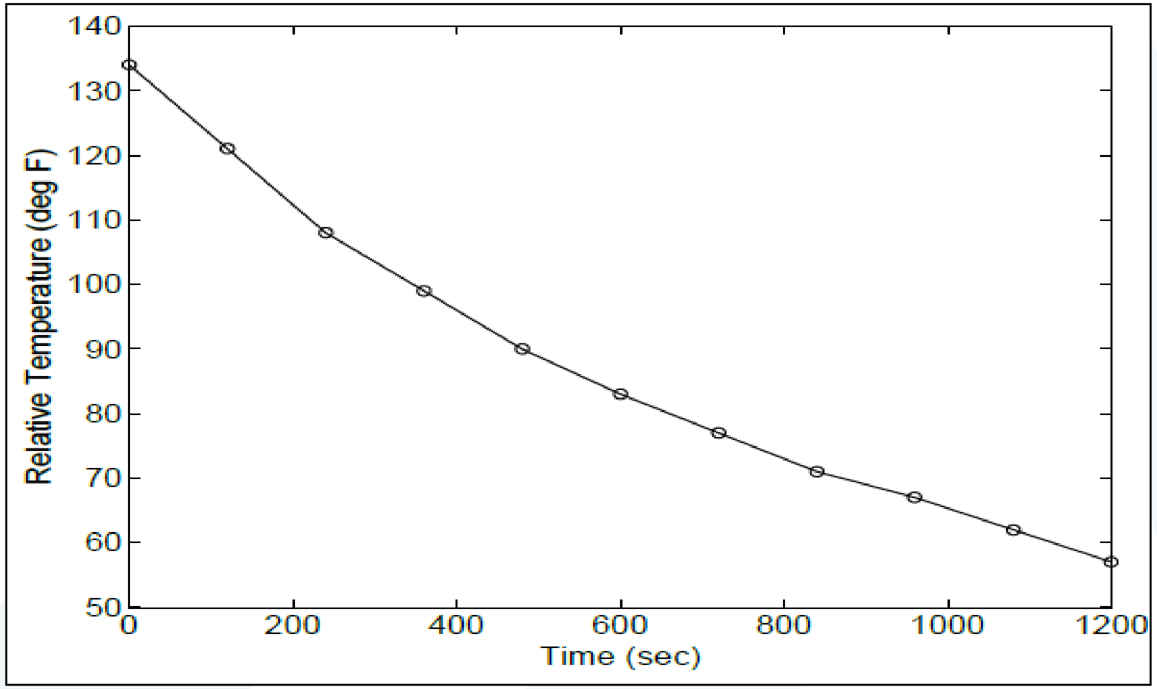
Water in a glass measuring cup was allowed to cool after being heated to 204°F. The ambient air temperature was 70°F. The measured water temperature at various times is given in the following table.

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

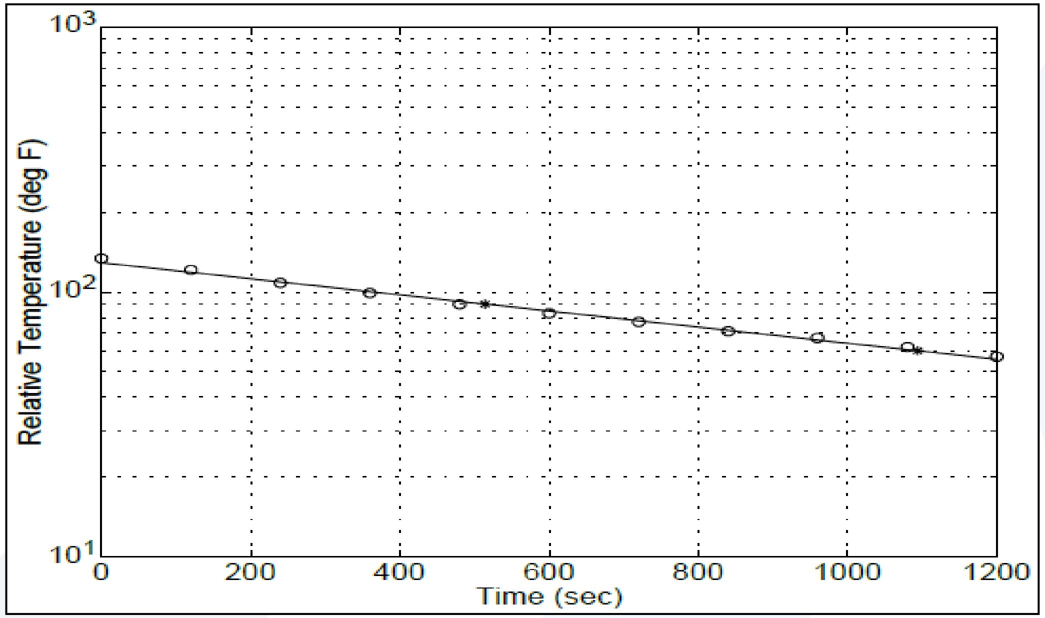
Time (sec)	720	840	960	1080	1200
Temperature (°F)	147	141	137	132	127

Obtain a functional description of the water temperature versus time.

Common sense tells us that the water temperature will eventually reach the air temperature of 70°. Thus we first subtract 70° from the temperature data T and seek to obtain a functional description of the relative temperature, $\Delta T = T - 70$. A plot of the relative temperature data is shown in Figure . We note that the plot has a distinct curvature and that it does not pass through the origin. Thus we can rule out the linear function and the power function as candidates.



To see if the data can be described by an exponential function, we plot the data on a semilog plot, which is shown in Figure . The straight line shown can be drawn by aligning a straight-edge so that it passes near most of the data points (note that this line is subjective; another person might draw a different line). The data lie close to a straight line, so we can use the exponential function to describe the relative temperature.

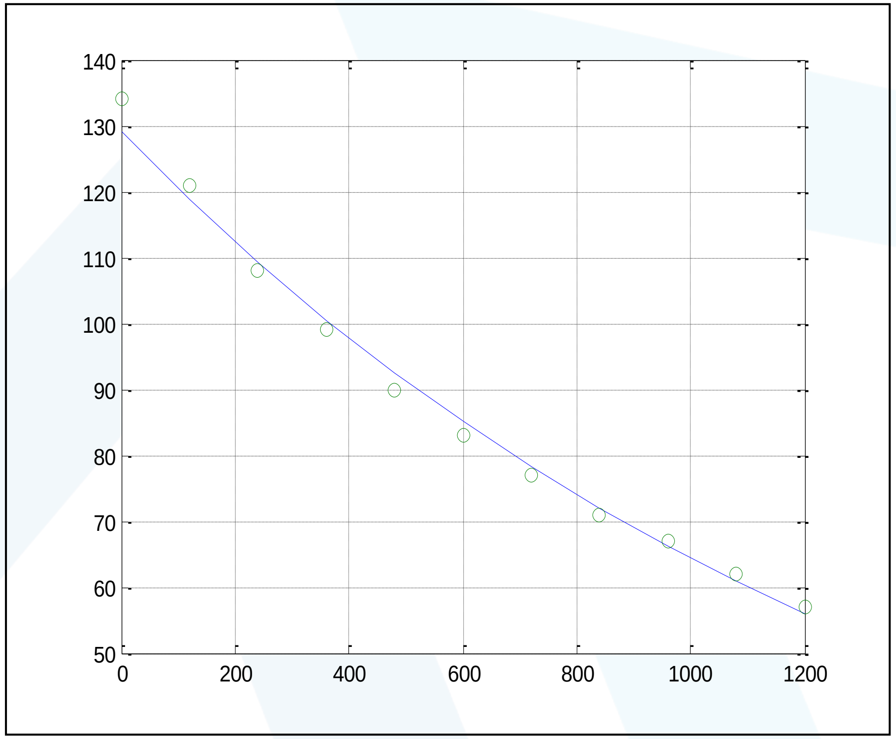


Using the second form of the exponential function, we can write $\Delta T = be^{mt}$.

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

```
time = (0:120:1200);
temp = [204,191,178,169,160,153,147,141,137,132,127];
rel_temp = temp - 70;
log_rel_temp = log(rel_temp);
p = polyfit(time,log_rel_temp,1);
m = p(1)
b = exp(p(2))
DT = b*exp(m*time);
plot(time,DT, time, rel_temp ,'o'),
grid
J = sum((DT-rel_temp).^2)
S = sum((rel_temp - mean(rel_temp)).^2)
r2 = 1 - J/S
```

```
m =
 -6.9710e-04
b =
 129.1624
J =
  47.4850
S =
  6.2429e+03
r2 =
   0.9924
```



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

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

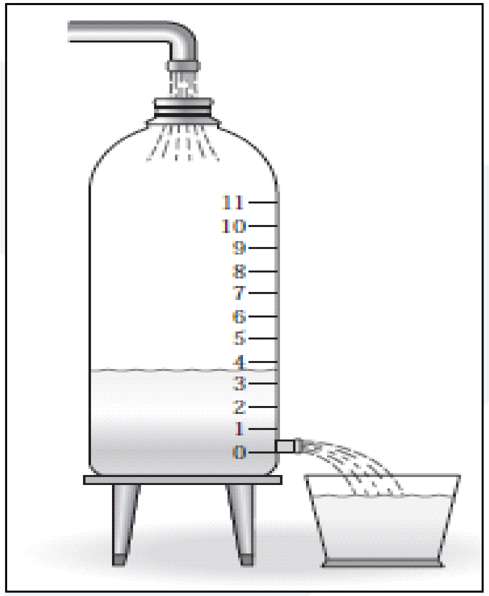
The quality-of-fit values are $J = 47.4850$, $S = 6.2429 \times 10^{3}$, and $r^2 = 0.9924$, which indicates a very good fit.

Example

A hole 6 mm in diameter was made in a translucent milk container (Figure). 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 $= 10^{-6}$ m^3). 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

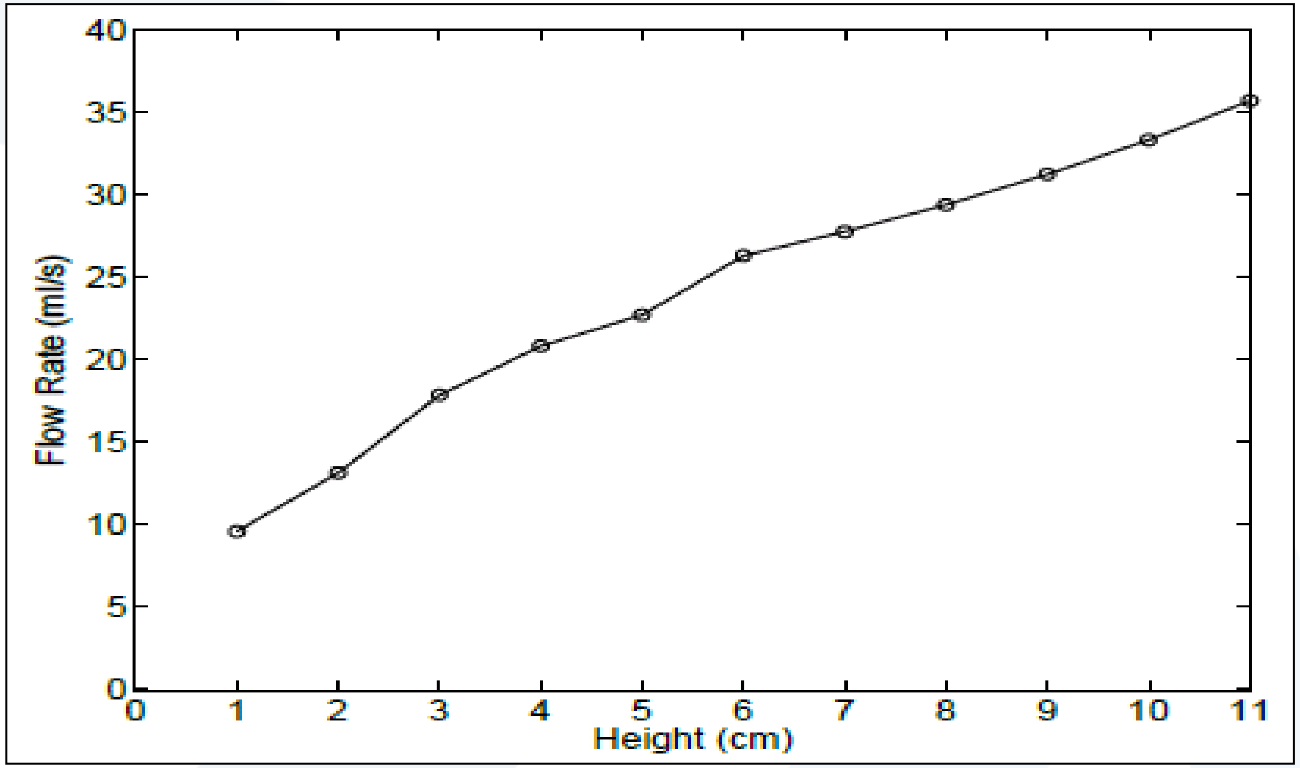
Obtain a functional description of the volume outflow rate f as a function of water height h above the hole.



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

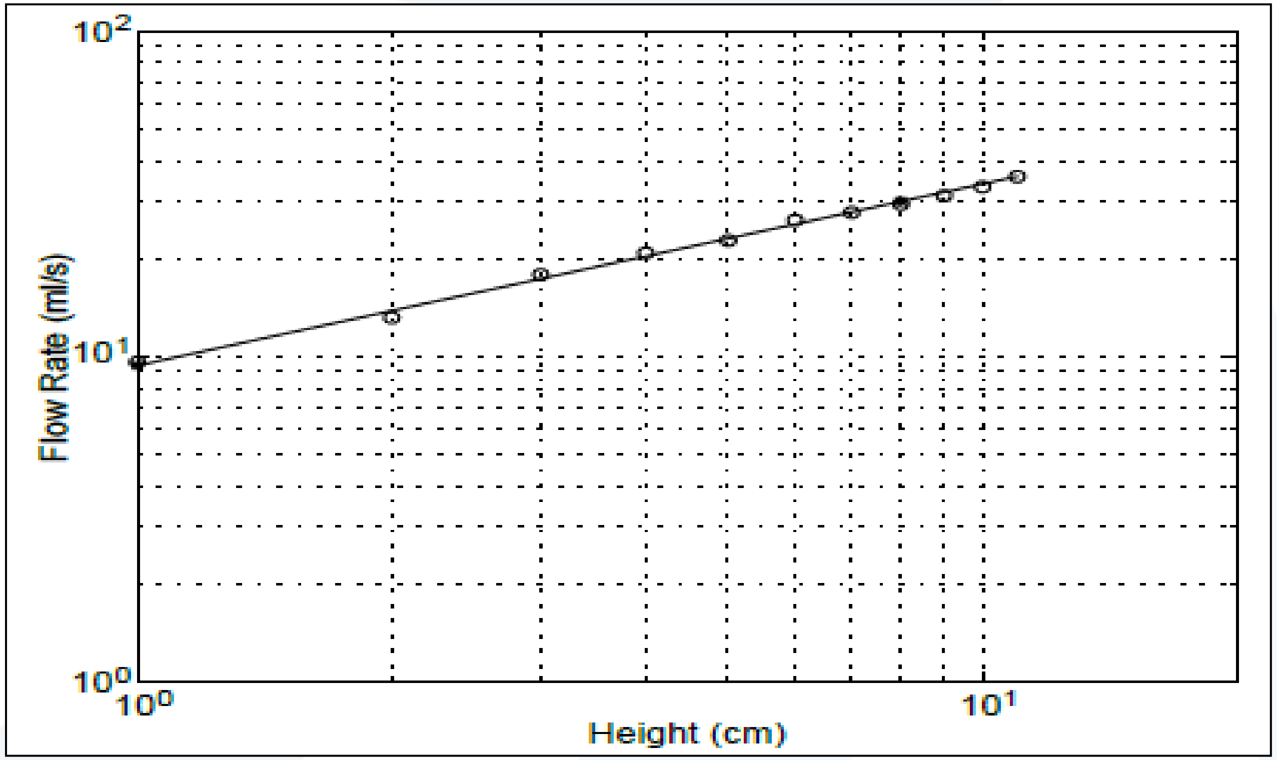
$$f = \frac{250}{t}$$

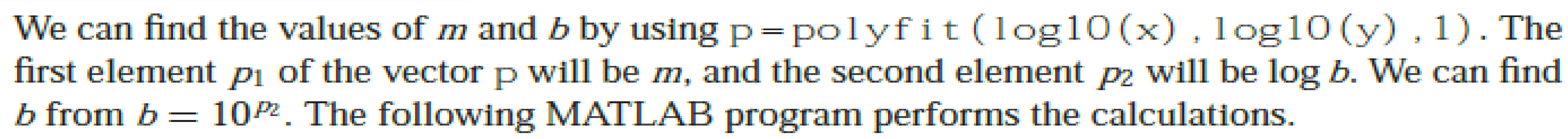
A plot of the resulting flow rate data is shown in Figure . There is some curvature in the plot, so we rule out the linear function. Common sense tells us that the outflow rate will be zero when the height is zero, so we can rule out the exponential function because it cannot pass through the origin.


Flow Rate (ml/s)
Height (cm)
0
5
10
15
20
25
30
35
40
0
1
2
3
4
5
6
7
8
9
10
11

The log-log plot shown in Figure shows that the data lie close to a straight line, so we can use the power function to describe the flow rate as a function of height. Thus we can write

$$f = bh^m$$

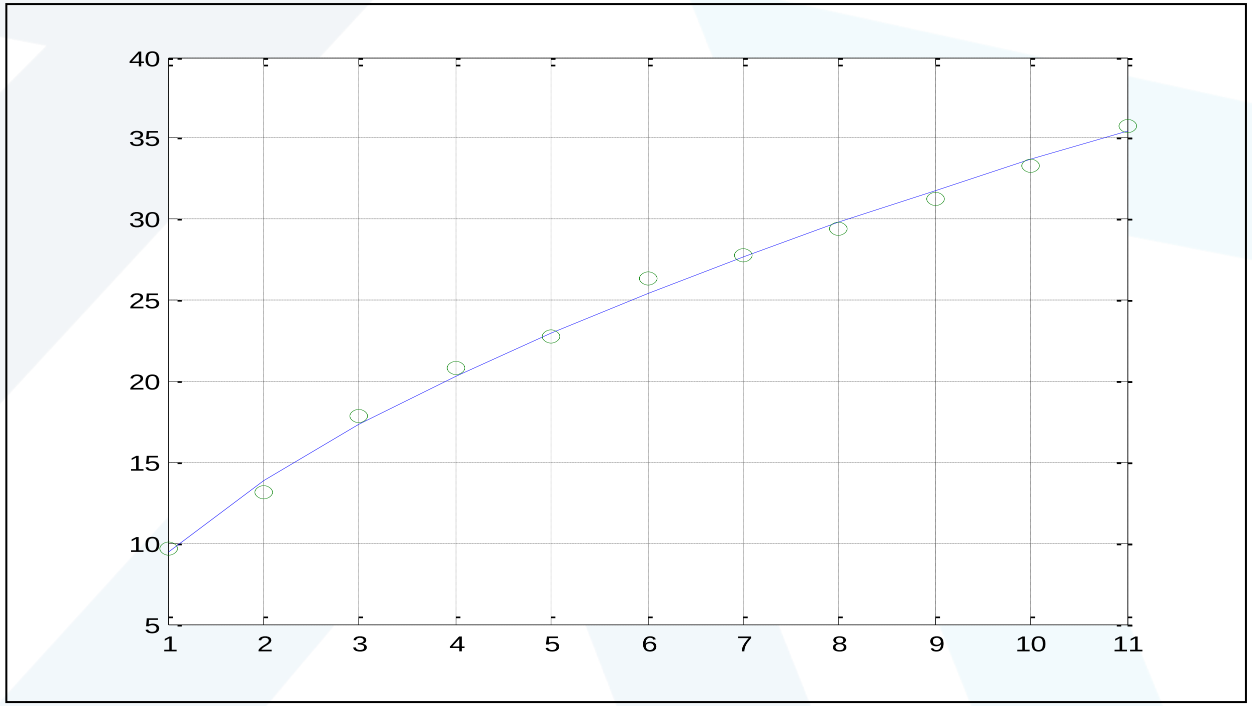


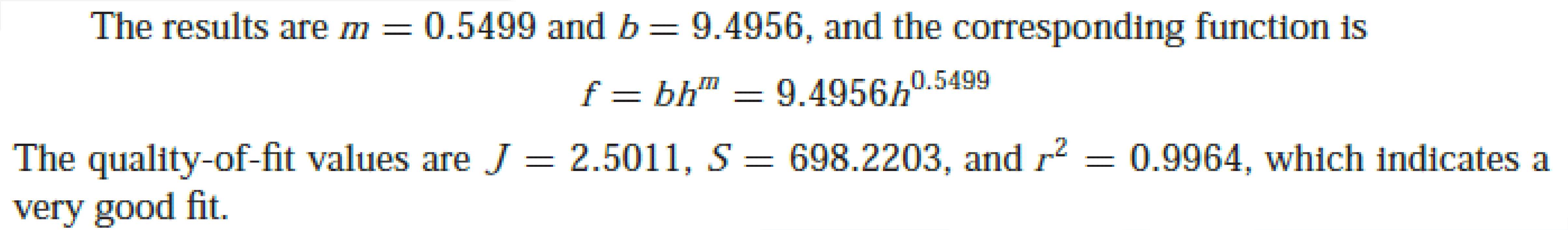


We can find the values of m and b by using p=polyfit(log10(x),log10(y),1). The first element p_1 of the vector p will be m, and the second element p_2 will be log b. We can find b from $b = 10^{p_2}$. The following MATLAB program performs the calculations.

```
h = (1:11);
time = [26, 19, 14, 12, 11, 9.5, 9, 8.5, 8, 7.5, 7];
flow = 250./time;
logflow = log10(flow);logheight = log10(h);
p = polyfit(logheight,logflow, 1);
m = p(1)
b = 10^p(2)
F = b*h.^m;
plot(h,F, h, flow ,'o'),
grid
J = sum((F - flow).^2)
S = sum((flow - mean(flow)).^2)
r2 = 1 - J/S
```

```
m =
   0.5499
b =
   9.4956
J =
   2.5011
S =
 698.2203
r2 =
   0.9964
```





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

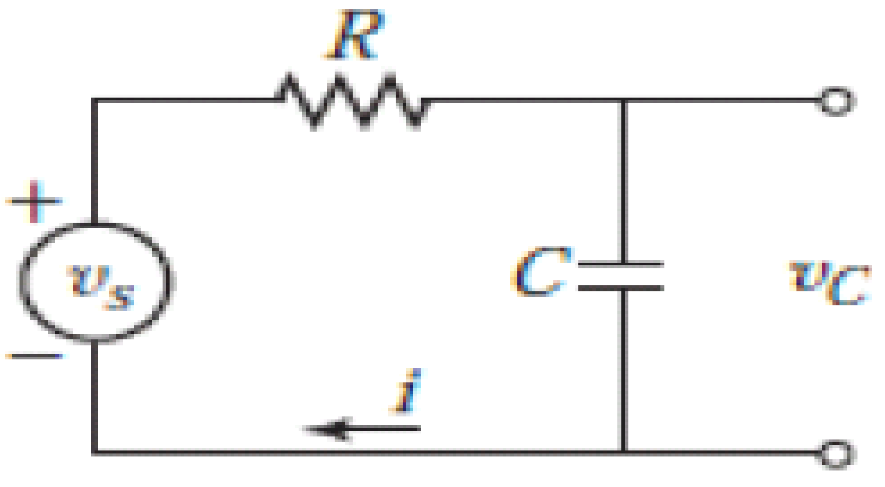
$$f = bh^m = 9.4956h^{0.5499}$$

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

Example

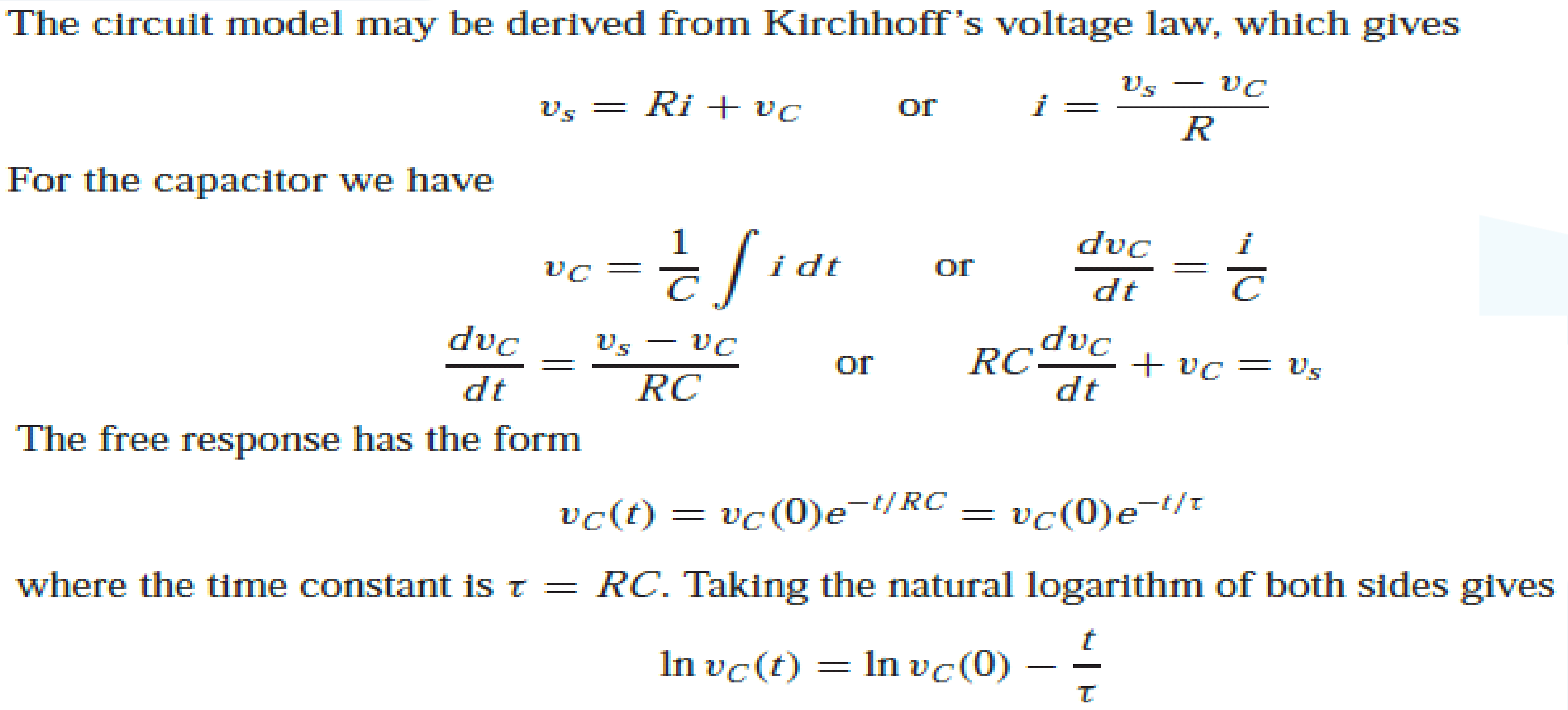
إذا علمت أن المقاومة في الدارة المبينة تبلغ $10^5\,\Omega$

و بأن الجهد طبق عندما t<0 ثم تم إيقافه فجأة عندما t=0. حيث تم عند تلك اللحظة قياس الجهد على المكثف خلال فترات زمنية لحظية كما هو مبين في الجدول



Time t (s)	Voltage v_C (V)
0	5
0.25	3.3
0.5	2.2
0.75	1.4
1	0.9
1.25	0.6
1.5	0.4
1.75	0.3
2	0.2

و المطلوب: استخدم هذه البيانات لتقدير قيمة سعة المكثف C بالاعتماد على طريقة **Least Square method**



The circuit model may be derived from Kirchhoff's voltage law, which gives

$$v_s = Ri + v_C \qquad \text{or} \qquad i = \frac{v_s - v_C}{R}$$

For the capacitor we have

$$v_C = \frac{1}{C}\int i\,dt \qquad \text{or} \qquad \frac{dv_C}{dt} = \frac{i}{C}$$

$$\frac{dv_C}{dt} = \frac{v_s - v_C}{RC} \qquad \text{or} \qquad RC\frac{dv_C}{dt} + v_C = v_s$$

The free response has the form

$$v_C(t) = v_C(0)e^{-t/RC} = v_C(0)e^{-t/\tau}$$

where the time constant is $\tau = RC$. Taking the natural logarithm of both sides gives

$$\ln v_C(t) = \ln v_C(0) - \frac{t}{\tau}$$

When the logarithmic transformation is applied to the original data, we obtain the following table.

$$m\sum_{i=1}^{n} t_i^{\ 2} + b\sum_{i=1}^{n} t_i = \sum_{i=1}^{n} \ln(vc)_i * t_i$$

$$m\sum_{i=1}^{n} t_i + b*9 = \sum_{i=1}^{n} \ln(vc)_i$$

$$\sum_{i=1}^{n} t_i^{\ 2} = 12.75 \quad \sum_{i=1}^{n} t_i = 9 \quad \sum_{i=1}^{n} \ln(vc)_i * t_i = \text{-}6.4991 \quad \sum_{i=1}^{n} \ln(vc)_i = \text{-}0.4176$$

m= -1.6217 b= 1.575

Time t (s)	$\ln v_C$
0	1.609
0.25	1.194
0.5	0.789
0.75	0.337
1	−0.105
1.25	−0.511
1.5	−0.916
1.75	−1.204
2	−1.609

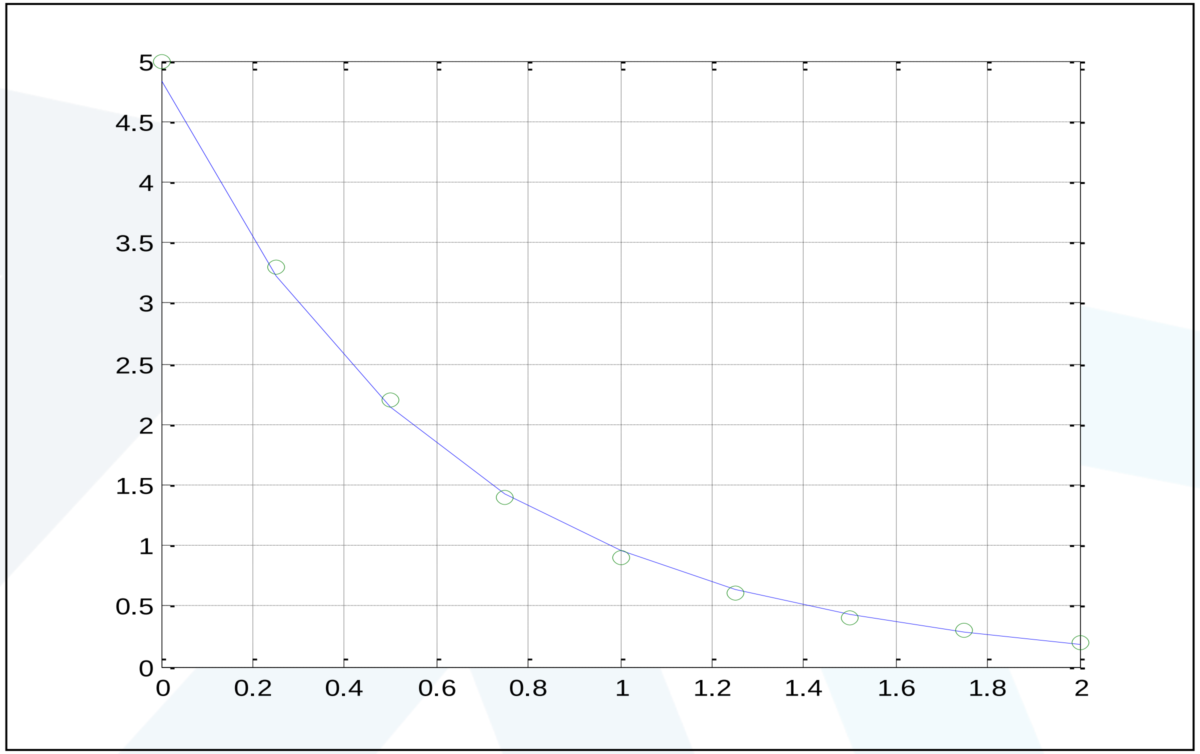
$$vc = \exp(1.575).e^{-1.6217t} = 4.832e^{-1.6127t}$$

$$\tau = 1/1.621 = 0.6166$$

Because we know that $R = 10^5\ \Omega$, we obtain $C = \tau/R = 0.617/10^5 = 6.17 \times 10^{-6}$ F.

```
time = [0:0.25:2];
voltage = [5 3.3 2.2 1.4 0.9 0.6 0.4 0.3 0.2];
logvoltage = log(voltage);
p = polyfit(time, logvoltage , 1);
m = p(1),b = exp(p(2))
T=-1/m;
C=T/(10^5)
vc = b*exp(time*m);
plot(time, vc, time , voltage ,'o'),
grid
J = sum((vc - voltage).^2)
S = sum((voltage - mean(voltage)).^2)
r2 = 1 - J/S
```

```
m =
  -1.6217
b =
   4.8323
C =
  6.1662e-06
J =
   0.0433
S =
  21.4289
r2 =
   0.9980
```



انتهت المحاضرة