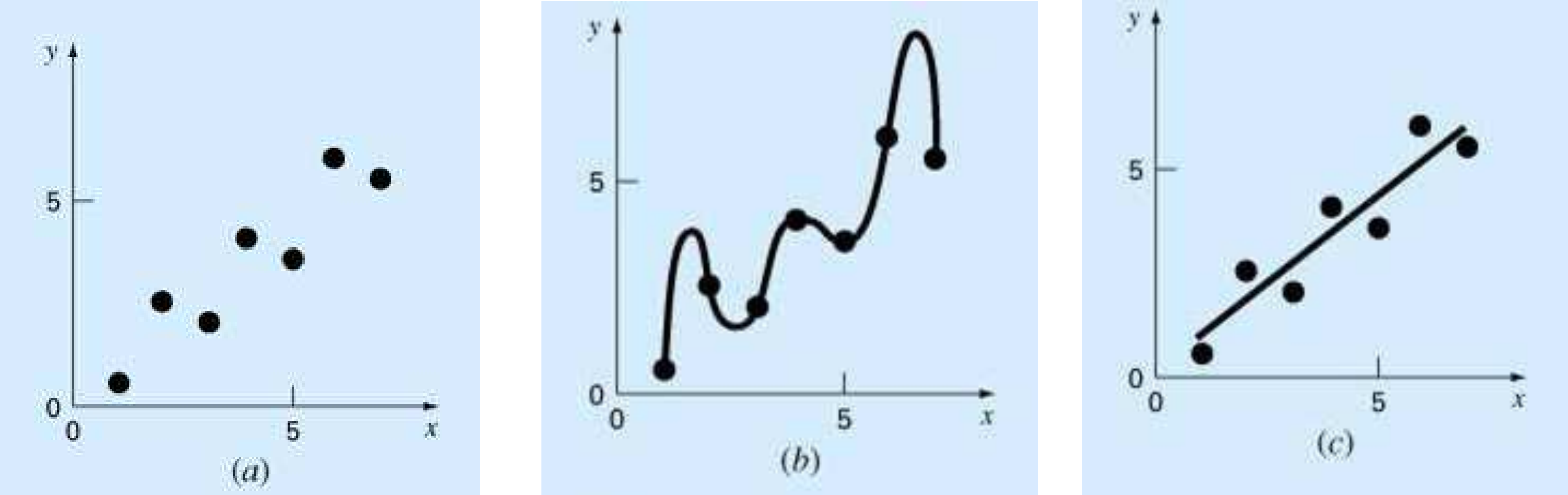
Numerical Methods

CURVE FITTING

Contents

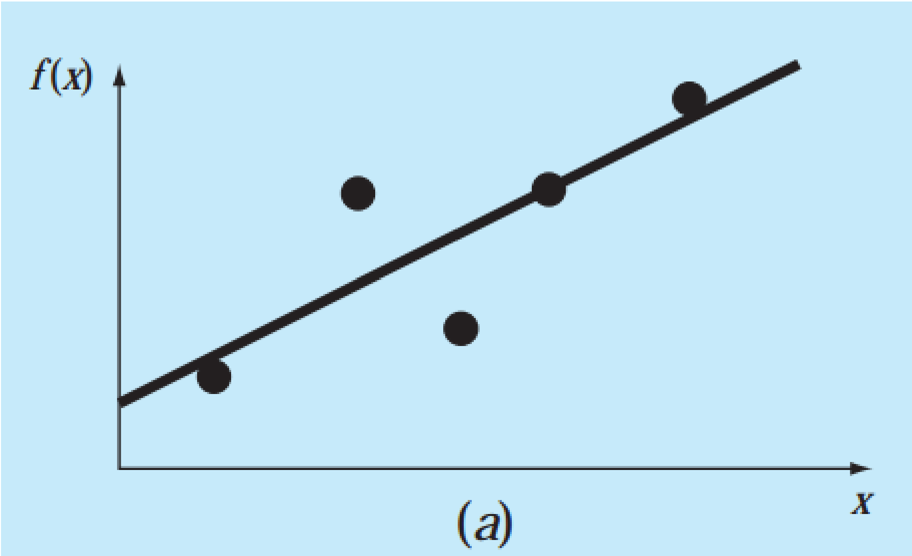
❖ Topics

➤ Curve fitting



Objectives

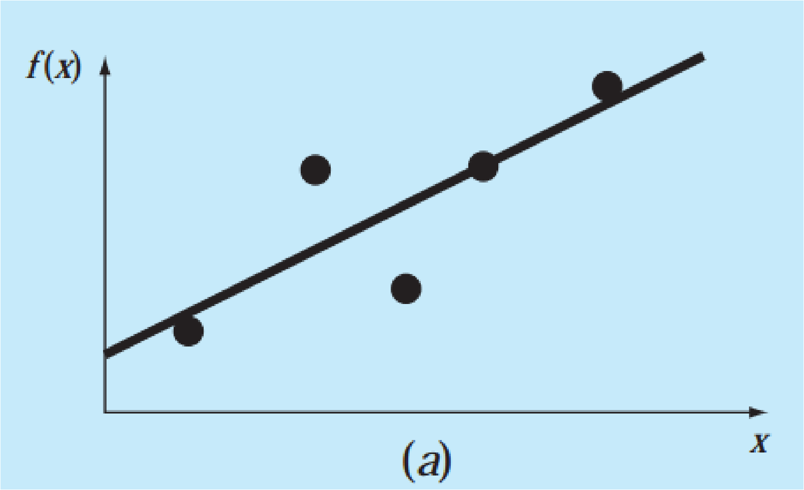
- Data is often given for discrete values along a continuum. However, you may require estimates at points between the discrete values.

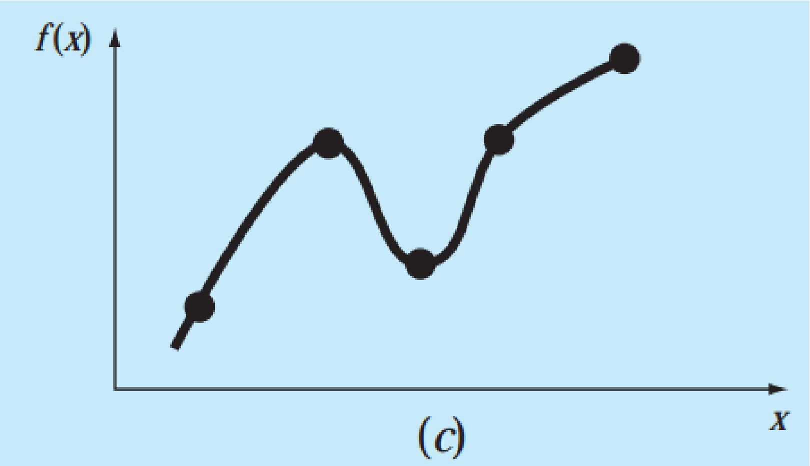


→ curve fitting

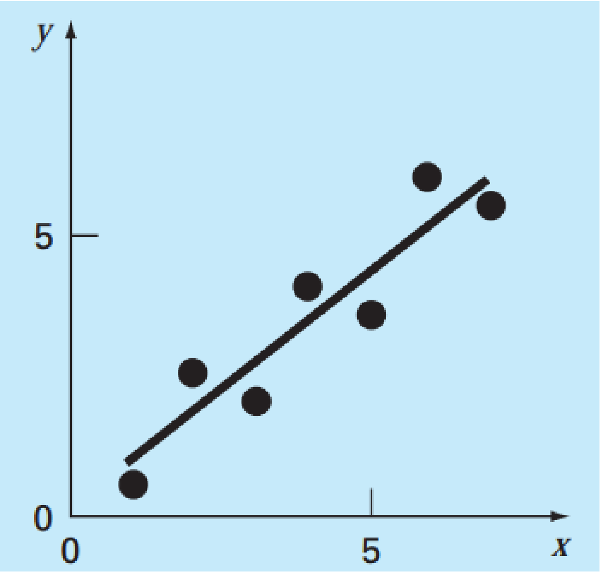
Curve fitting

- Methods
  - Least-squares regression.
  - Interpolation



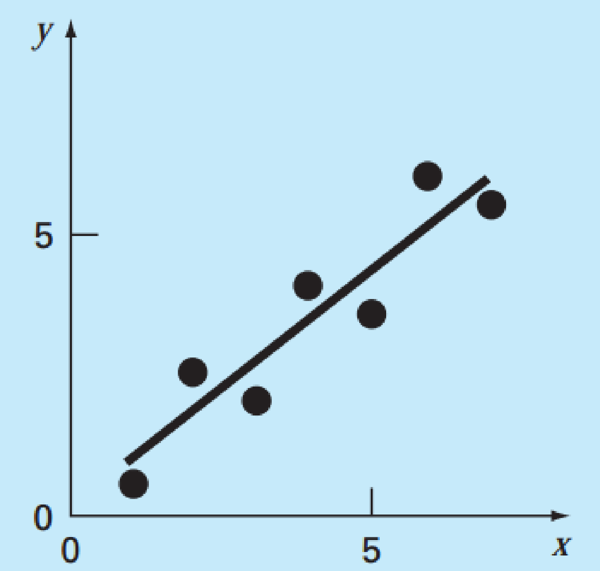


- Least-Squares Regression
  - Linear regression
  - Polynomial regression
  - Multiple linear regression



❖ Linear regression

➢ The simplest example of a least-squares approximation is fitting a straight line to a set of paired observations: (x1, y1), (x2, y2), . . . , (xn, yn). The mathematical expression for the straight line is



$$y = a_0 + a_1 x + e$$

where a_0 and a_1 are coefficients representing the intercept and the slope, respectively, and e is the error, or residual, between the model and the observations

Least-Squares Regression

$$y = a_0 + a_1 x + e \quad \Rightarrow \quad e = y - a_0 - a_1 x$$

Thus, the error, or residual, is the discrepancy between the true value of y and the approximate value, $a_0 + a_1x$, predicted by the linear equation.

- **Criteria for a "Best" Fit**

One strategy for fitting a "best" line through the data would be to minimize the sum of the residual errors for all the available data, as in:

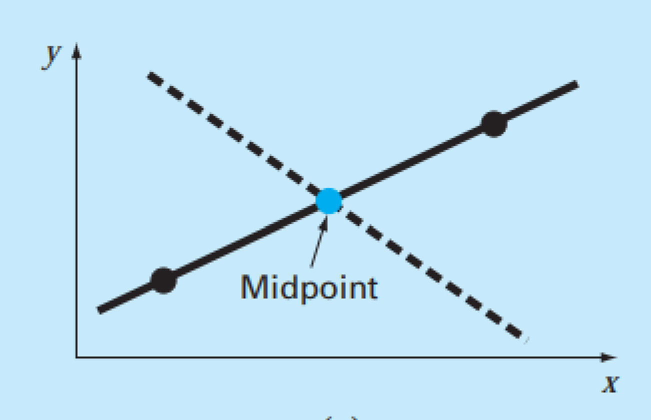
$$\sum_{i=1}^{n} e_i = \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i)$$

where n = total number of points

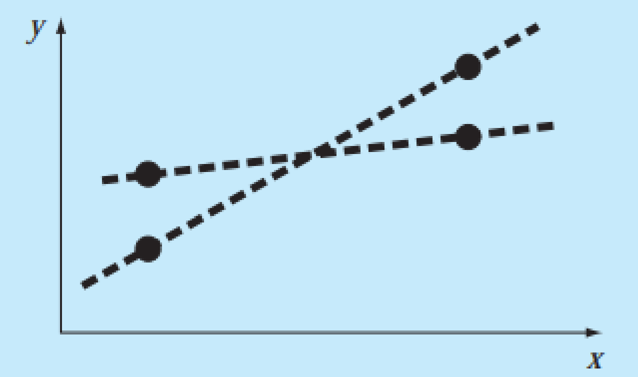
Least-Squares Regression

$$\sum_{i=1}^{n} e_i = \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i)$$

$$\sum_{i=1}^{n} |e_i| = \sum_{i=1}^{n} |y_i - a_0 - a_1 x_i|$$



(a)



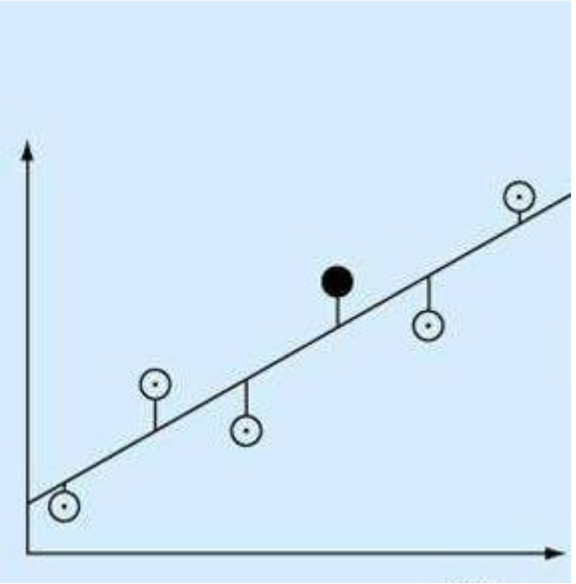
(b)

Examples of some criteria for "best fit" that are inadequate for regression:
(a) minimizes the sum of the residuals,
(b) minimizes the sum of the absolute values of the residuals

Least-Squares Regression

A strategy that overcomes the shortcomings of the aforementioned approaches is to minimize the sum of the squares of the residuals between the measured *y* and the *y* calculated with the linear model

$$S_r = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_{i,\text{measured}} - y_{i,\text{model}})^2 = \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i)^2$$



all summations are from $i = 1$ to n

This criterion, which is called least squares

Least-Squares Regression

- Least-Squares Fit of a Straight Line

To determine values for a_0 and a_1, Eq. is differentiated with respect to each coefficient:

$$\frac{\partial S_r}{\partial a_0} = -2 \sum (y_i - a_0 - a_1 x_i)$$

$$\frac{\partial S_r}{\partial a_1} = -2 \sum [(y_i - a_0 - a_1 x_i) x_i]$$

all summations are from i = 1 to *n*

Setting these derivatives equal to zero will result in a minimum Sr. If this is done, the equations can be expressed as

$$0 = \sum y_i - \sum a_0 - \sum a_1 x_i$$

$$0 = \sum y_i x_i - \sum a_0 x_i - \sum a_1 x_i^2$$

Least-Squares Regression

- Least-Squares Fit of a Straight Line

Now, realizing that $a_0 = na_0$, we can express the equations as a set of two simultaneous linear equations with two unknowns (a 0 and a1):

$$na_0 + \left(\sum x_i\right) a_1 = \sum y_i$$

$$\left(\sum x_i\right) a_0 + \left(\sum x_i^2\right) a_1 = \sum x_i y_i$$

These are called the normal equations. They can be solved simultaneously

$$a_1 = \frac{n\Sigma x_i y_i - \Sigma x_i \Sigma y_i}{n\Sigma x_i^2 - (\Sigma x_i)^2}$$

$$a_0 = \bar{y} - a_1 \bar{x}$$

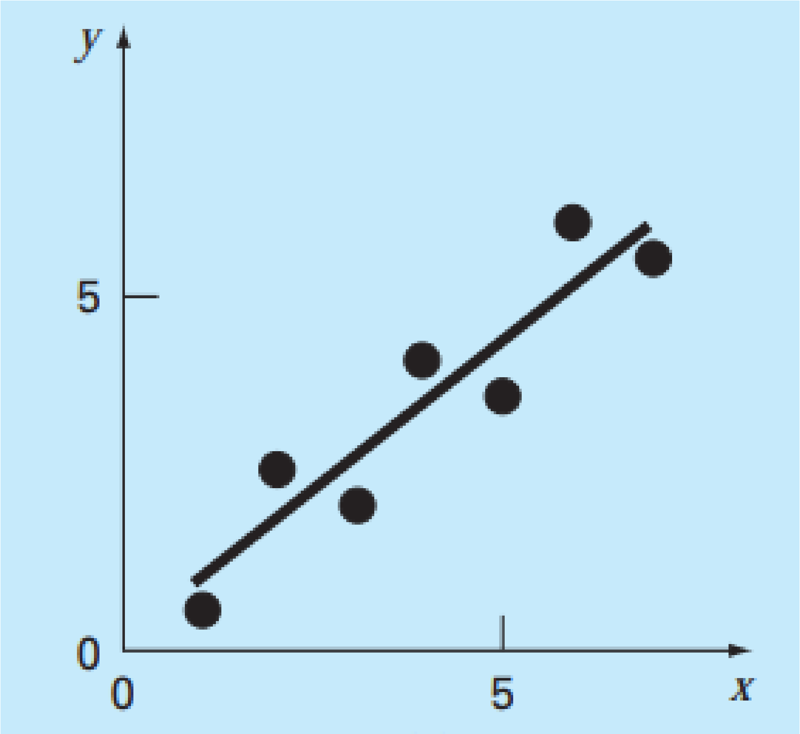
where $\bar{y}$ and $\bar{x}$ are the means of y and x, respectively.

$$\bar{y} = \frac{\sum y_i}{n}$$

Least-Squares Regression

Problem Statement. Fit a straight line to the x and y values in the first two columns of Table 17.1.

x_i	y_i
1	0.5
2	2.5
3	2.0
4	4.0
5	3.5
6	6.0
7	5.5
Σ	24.0



Solution. The following quantities can be computed:

$$n = 7 \qquad \sum x_i y_i = 119.5 \qquad \sum x_i^2 = 140$$

$$\sum x_i = 28 \qquad \bar{x} = \frac{28}{7} = 4$$

$$\sum y_i = 24 \qquad \bar{y} = \frac{24}{7} = 3.428571$$

TABLE 17.1 Computations for an error analysis of the linear fit.

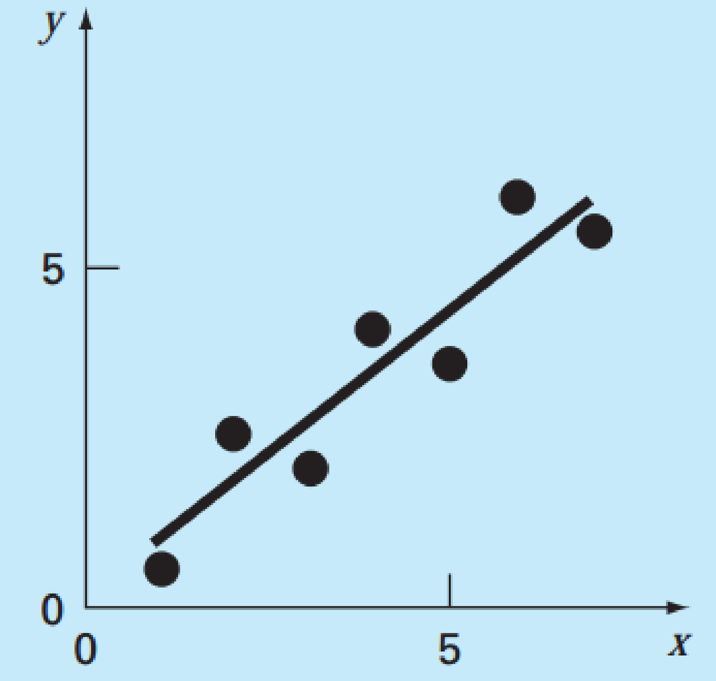
x_i	y_i	$(y_i - \bar{y})^2$	$(y_i - a_0 - a_1 x_i)^2$
1	0.5	8.5765	0.1687
2	2.5	0.8622	0.5625
3	2.0	2.0408	0.3473
4	4.0	0.3265	0.3265
5	3.5	0.0051	0.5896
6	6.0	6.6122	0.7972
7	5.5	4.2908	0.1993
Σ	24.0	22.7143	2.9911

Least-Squares Regression

Using Eqs. (17.6) and (17.7),

$$a_1 = \frac{7(119.5) - 28(24)}{7(140) - (28)^2} = 0.8392857$$

$$a_0 = 3.428571 - 0.8392857(4) = 0.07142857$$



(c)

Therefore, the least-squares fit is

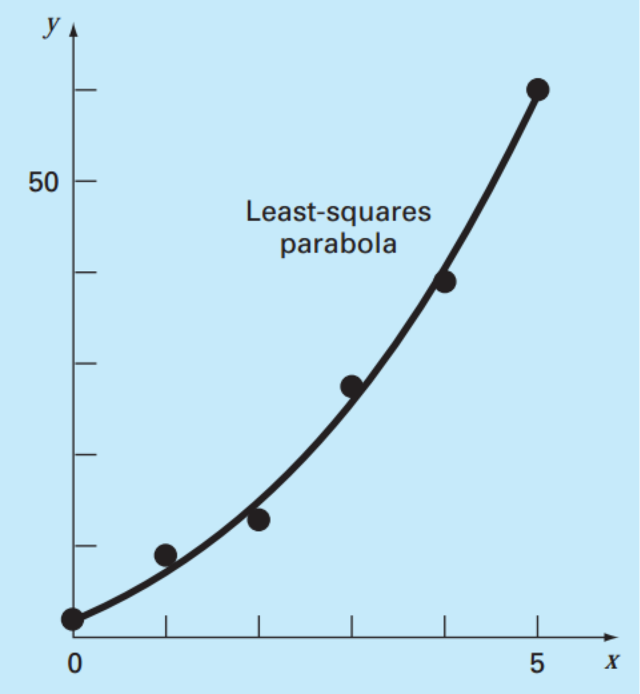
$$y = 0.07142857 + 0.8392857x$$

The line, along with the data, is shown in Fig. 17.1*c*.

Polynomial regression

- The least-squares procedure can be readily extended to fit the data to a higher-order polynomial. For example, suppose that we fit a second-order polynomial or quadratic:

$$y = a_0 + a_1 x + a_2 x^2 + e$$



For this case the sum of the squares of the residuals is

$$S_r = \sum_{i=1}^{n} \left(y_i - a_0 - a_1 x_i - a_2 x_i^2\right)^2$$

❖ Polynomial regression

➤ Following the procedure of the previous section, we take the derivative of Eq. (17.18) with respect to each of the unknown coefficients of the polynomial, as in:

$$\frac{\partial S_r}{\partial a_0} = -2\sum \left(y_i - a_0 - a_1 x_i - a_2 x_i^2\right)$$

$$\frac{\partial S_r}{\partial a_1} = -2\sum x_i \left(y_i - a_0 - a_1 x_i - a_2 x_i^2\right)$$

$$\frac{\partial S_r}{\partial a_2} = -2\sum x_i^2 \left(y_i - a_0 - a_1 x_i - a_2 x_i^2\right)$$

Polynomial regression

- These equations can be set equal to zero and rearranged to develop the following set of normal equations:

$$(n)a_0 + \left(\sum x_i\right)a_1 + \left(\sum x_i^2\right)a_2 = \sum y_i$$
$$\left(\sum x_i\right)a_0 + \left(\sum x_i^2\right)a_1 + \left(\sum x_i^3\right)a_2 = \sum x_i y_i$$
$$\left(\sum x_i^2\right)a_0 + \left(\sum x_i^3\right)a_1 + \left(\sum x_i^4\right)a_2 = \sum x_i^2 y_i$$

Note that the above three equations are linear and have three unknowns: a_0, a_1, and a_2. The coefficients of the unknowns can be calculated directly from the observed data.

Polynomial regression

- The two-dimensional case can be easily extended to an mth-order polynomial as

$$y = a_0 + a_1 x + a_2 x^2 + \cdots + a_m x^m + e$$

the standard error is formulated as

$$s_{y/x} = \sqrt{\frac{S_r}{n - (m+1)}}$$

Problem Statement. Fit a second-order polynomial to the data in the first two columns of Table 17.4.

TABLE 17.4 Computations for an error analysis of the quadratic least-squares fit.

x_i	y_i	$(y_i - \bar{y})^2$	$(y_i - a_0 - a_1 x_i - a_2 x_i^2)^2$
0	2.1	544.44	0.14332
1	7.7	314.47	1.00286
2	13.6	140.03	1.08158
3	27.2	3.12	0.80491
4	40.9	239.22	0.61951
5	61.1	1272.11	0.09439
Σ	152.6	2513.39	3.74657

Least-Squares Regression

Solution. From the given data,

$$
\begin{array}{lll}
m = 2 & \sum x_i = 15 & \sum x_i^4 = 979 \\
n = 6 & \sum y_i = 152.6 & \sum x_i y_i = 585.6 \\
\bar{x} = 2.5 & \sum x_i^2 = 55 & \sum x_i^2 y_i = 2488.8 \\
\bar{y} = 25.433 & \sum x_i^3 = 225 &
\end{array}
$$

Therefore, the simultaneous linear equations are

$$
\begin{bmatrix} 6 & 15 & 55 \\ 15 & 55 & 225 \\ 55 & 225 & 979 \end{bmatrix} \begin{Bmatrix} a_0 \\ a_1 \\ a_2 \end{Bmatrix} = \begin{Bmatrix} 152.6 \\ 585.6 \\ 2488.8 \end{Bmatrix}
$$

Least-Squares Regression

Solving these equations through a technique such as Gauss elimination gives $a_0 = 2.47857$, $a_1 = 2.35929$, and $a_2 = 1.86071$. Therefore, the least-squares quadratic equation for this case is

$$y = 2.47857 + 2.35929x + 1.86071x^2$$

The standard error of the estimate based on the regression polynomial is [Eq. (17.20)]

$$s_{y/x} = \sqrt{\frac{3.74657}{6-3}} = 1.12$$

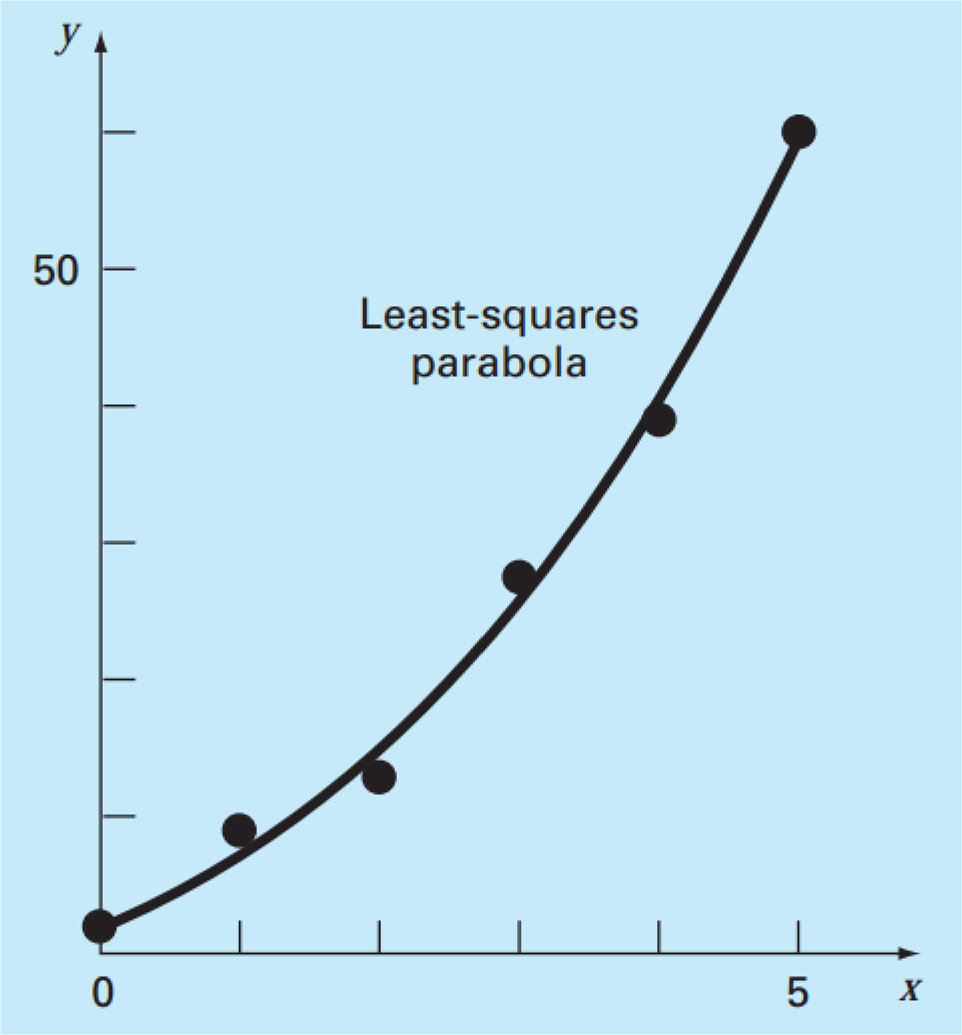
The coefficient of determination is

$$r^2 = \frac{2513.39 - 3.74657}{2513.39} = 0.99851$$

and the correlation coefficient is $r = 0.99925$.

These results indicate that 99.851 percent of the original uncertainty has been explained by the model. This result supports the conclusion that the quadratic equation represents an excellent fit, as is also evident from Fig. 17.11.

Least-Squares Regression



Algorithm for Polynomial Regression

Algorithm for implementation of polynomial and multiple linear regression.

Step 1: Input order of polynomial to be fit, m.
Step 2: Input number of data points, n.
Step 3: If $n < m + 1$, print out an error message that regression is impossible and terminate the process. If $n \geq m + 1$, continue.
Step 4: Compute the elements of the normal equation in the form of an augmented matrix.
Step 5: Solve the augmented matrix for the coefficients $a_0, a_1, a_2, \ldots, a_m$, using an elimination method.
Step 6: Print out the coefficients.

Algorithm for Polynomial Regression

Pseudocode to assemble the elements of the normal equations for polynomial regression.

```
DOFOR i = 1, order + 1
  DOFOR j = 1, i
    k = i + j − 2
    sum = 0
    DOFOR ℓ = 1, n
      sum = sum + x_ℓ^k
    END DO
    a_i,j = sum
    a_j,i = sum
  END DO
  sum = 0
  DOFOR ℓ = 1, n
    sum = sum + y_ℓ · x_ℓ^(i−1)
  END DO
  a_i,order+2 = sum
END DO
```

Curve fitting with MATLAB

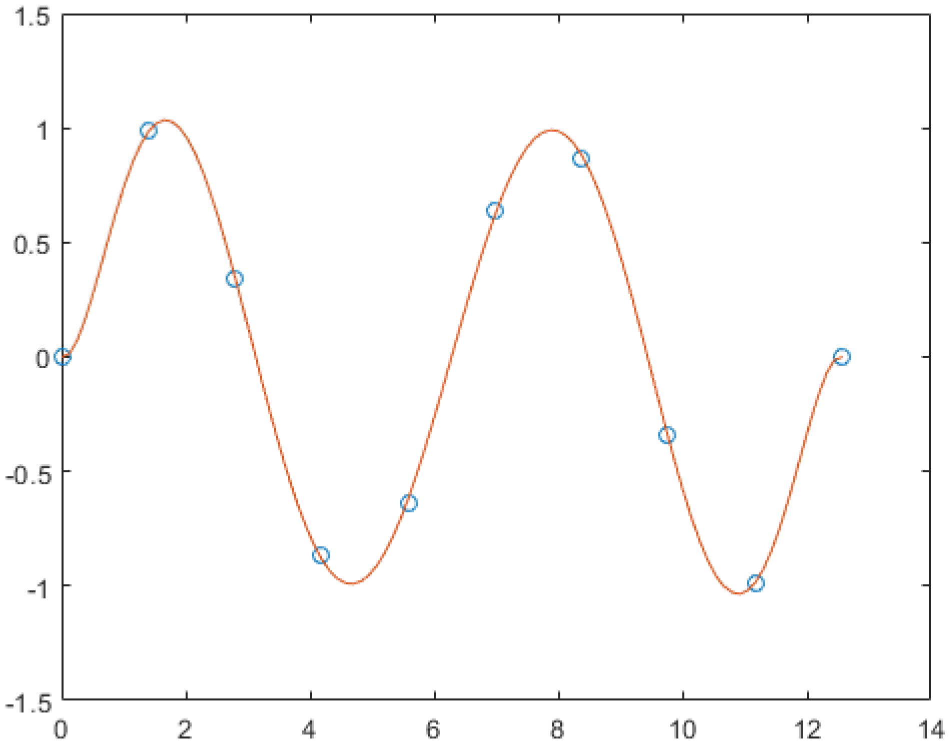
Some of the MATLAB functions to implement interpolation, regression, splines, and the FFT

Function	Description
polyfit	Fit polynomial to data
interp1	1-D interpolation (1-D table lookup)
interp2	2-D interpolation (2-D table lookup)
spline	Cubic spline data interpolation
fft	Discrete Fourier transform

Polyfit

- **p = polyfit(x,y,n)** returns the coefficients for a polynomial *p(x)* of degree *n* that is a best fit (in a least-squares sense) for the data in y.
- The coefficients in *p* are in descending powers, and the length of *p* is *n*+1

$$p(x) = p_1 x^n + p_2 x^{n-1} + \ldots + p_n x + p_{n+1}.$$



Syntax

```
p = polyfit(x,y,n)
[p,S] = polyfit(x,y,n)
[p,S,mu] = polyfit(x,y,n)
```

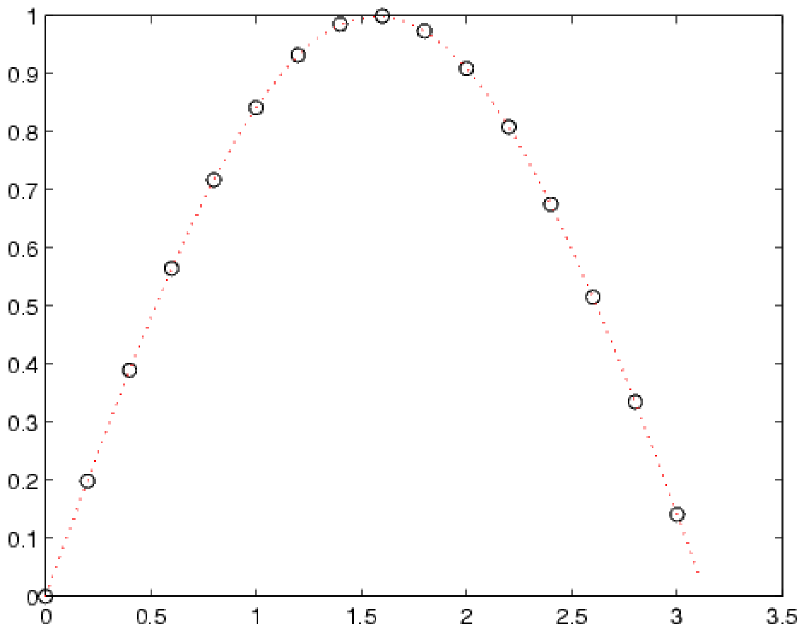
Matlab

interp1

- **vq = interp1(x,v,xq)** returns interpolated values of a 1-D function at specific query points using linear interpolation.
- Vector x contains the sample points, and v contains the corresponding values, v(x). Vector xq contains the coordinates of the query points.

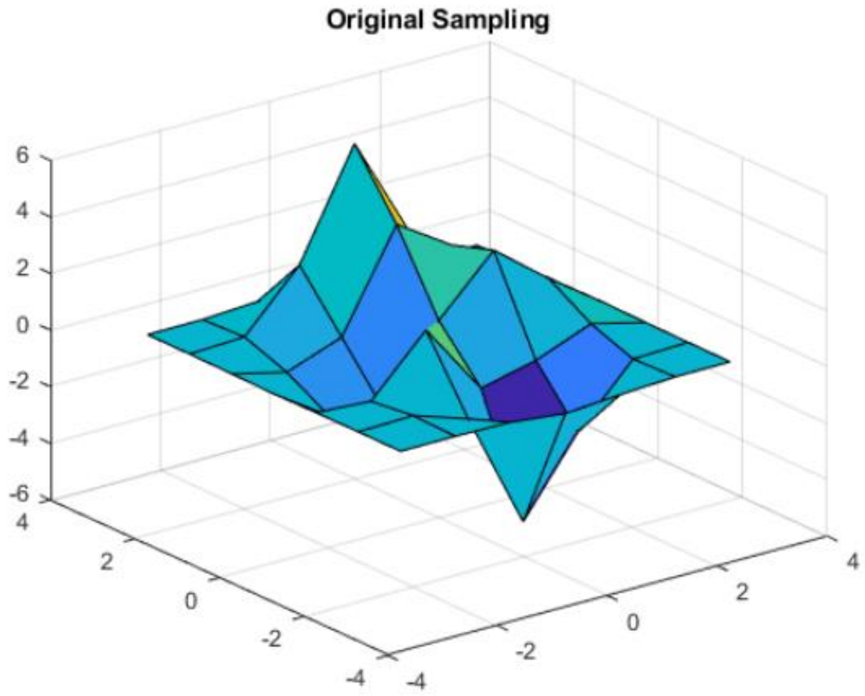
Syntax

```
vq = interp1(x,v,xq)
vq = interp1(x,v,xq,method)
vq = interp1(x,v,xq,method,extrapolation)
```



interp2

- **Vq = interp2(X,Y,V,Xq,Yq)** returns interpolated values of a function of two variables at specific query points using linear interpolation.
- The results always pass through the original sampling of the function. X and Y contain the coordinates of the sample points.
- V contains the corresponding function values at each sample point. Xq and Yq contain the coordinates of the query points.

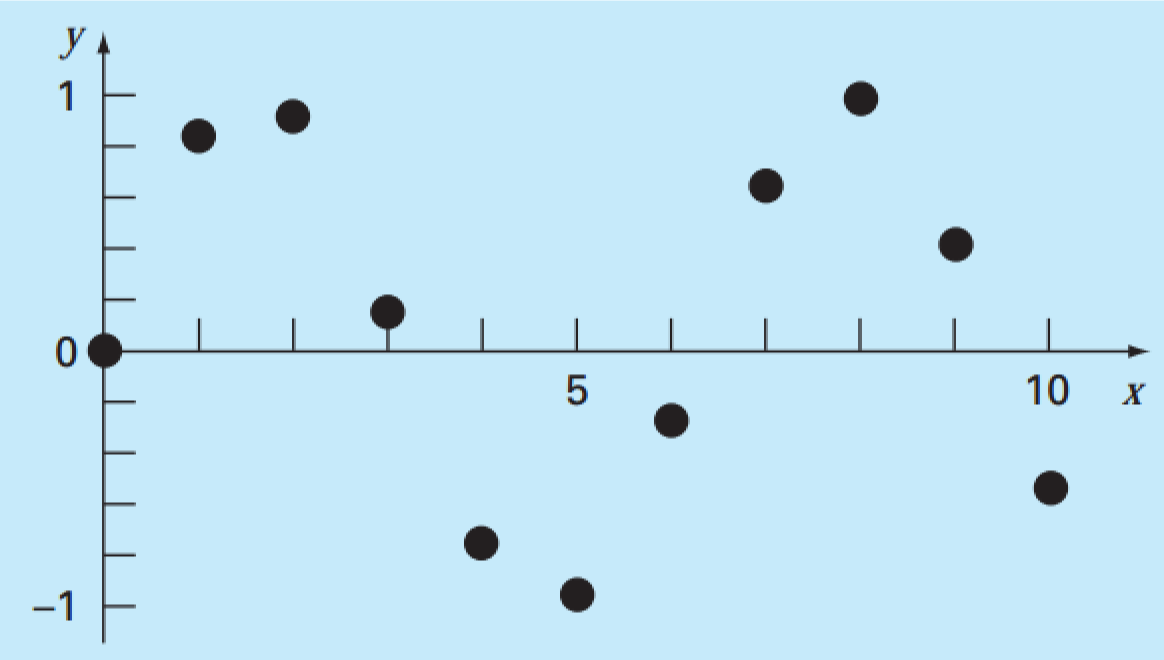


Syntax

```
Vq = interp2(X,Y,V,Xq,Yq)
Vq = interp2(V,Xq,Yq)
Vq = interp2(V)
Vq = interp2(V,k)
```

Matlab

Problem Statement. Explore how MATLAB can be employed to fit curves to data. To do this, use the sine function to generate equally spaced $f(x)$ values from 0 to 10. Employ a step size of 1 so that the resulting characterization of the sine wave is sparse (Fig. 19.22). Then, fit it with (*a*) linear interpolation, (*b*) a fifth-order polynomial, and (*c*) a cubic spline.



Eleven points sampled from a sinusoid

Matlab

Solution.

(a) The values for the independent and the dependent variables can be entered into vectors by

```
>> x=0:10;
>> y=sin(x);
```

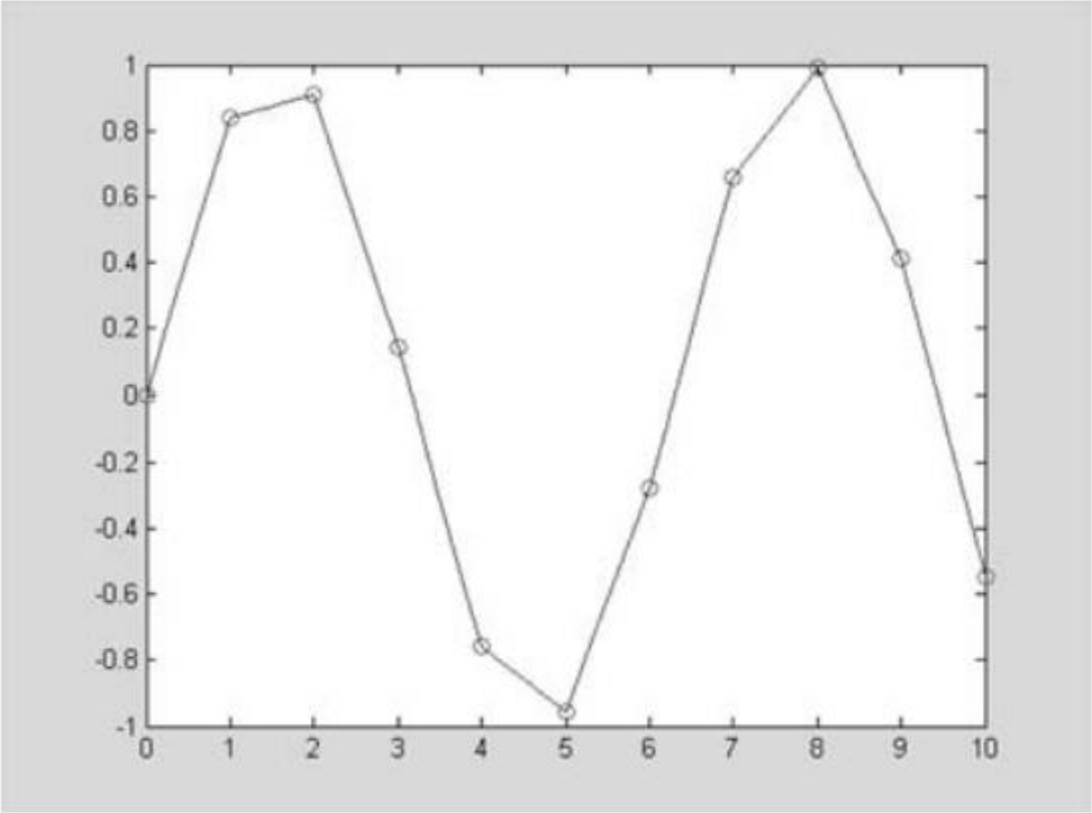
A new, more finely spaced vector of independent variable values can be generated and stored in the vector **xi,**

```
>> xi=0:.25:10;
```

The MATLAB function `interp1` can then be used to generate dependent variable values *yi* for all the *xi* values using linear interpolation. Both the original data (x, y) along with the linearly interpolated values can be plotted together,

Matlab

```
>> yi=interp1(x, y, xi);
>> plot(x, y, 'o', xi, yi)
```



20.6 A reactor is thermally stratified as in the following table:

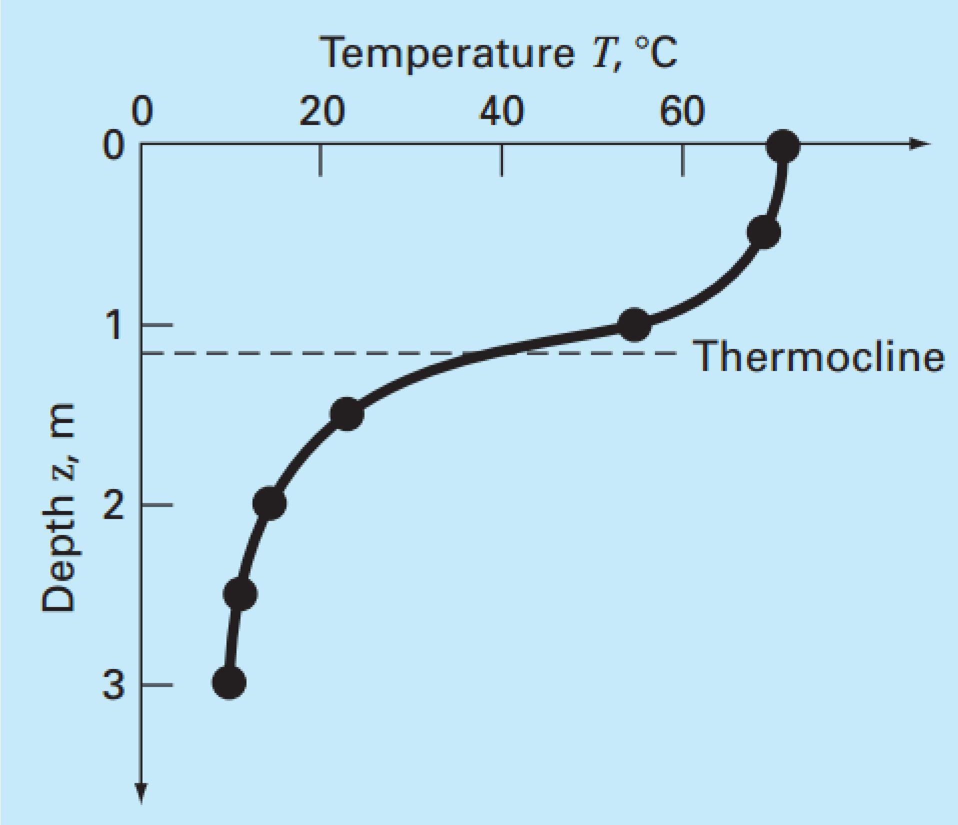
Depth, m	0	0.5	1.0	1.5	2.0	2.5	3.0
Temperature, °C	70	68	55	22	13	11	10

As depicted in Fig. P20.6, the tank can be idealized as two zones separated by a strong temperature gradient or thermocline. The depth of this gradient can be defined as the inflection point of the temperature-depth curve—that is, the point at which $d^2T/dz^2 = 0$.

At this depth, the heat flux from the surface to the bottom layer can be computed with Fourier's law,

$$J = -k\frac{dT}{dz}$$

Use a cubic spline fit of this data to determine the thermocline depth. If $k = 0.02$ cal/(s · cm · °C) compute the flux across this interface.



20.49 *Hooke's law,* which holds when a spring is not stretched too far, signifies that the extension of the spring and the applied force are linearly related. The proportionality is parameterized by the spring constant *k*. A value for this parameter can be established experimentally by placing known weights onto the spring and measuring the resulting compression. Such data were contained in Table P20.49 and plotted in Fig. P20.49. Notice that above a weight of 40×10^4 N, the linear relationship between the force and displacement breaks down. This sort of behavior is typical of what is termed a "hardening spring." Employ linear regression to determine a value of *k* for the linear portion of this system. In addition, fit a nonlinear relationship to the nonlinear portion.

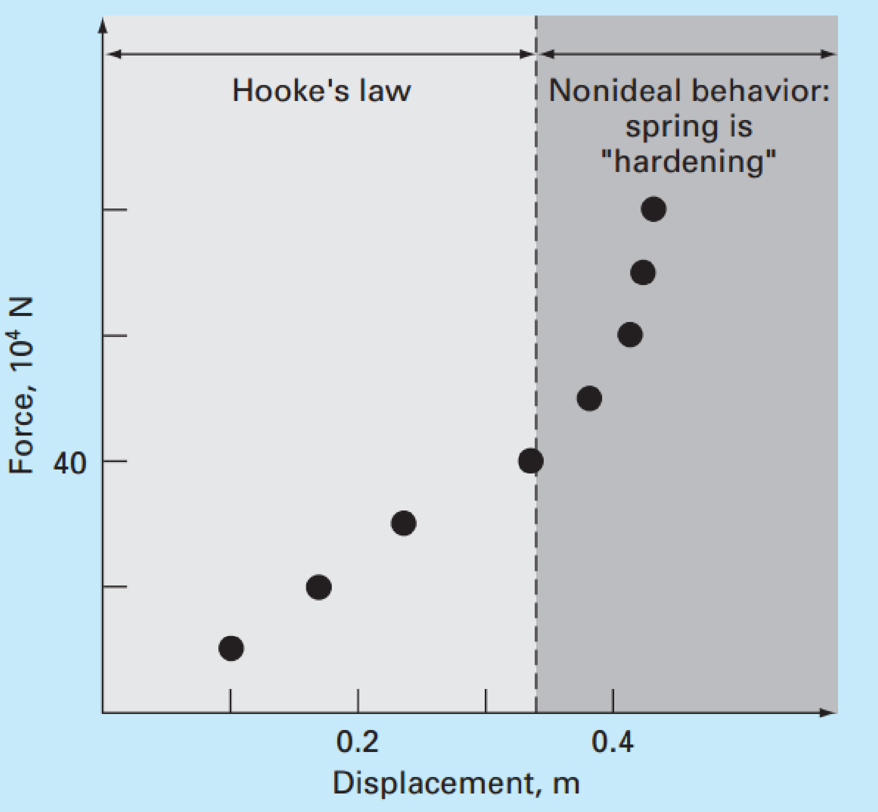


Table P20.49 Experimental values for elongation x and force F for the spring on an automobile suspension system.

Displacement, m	0.10	0.17	0.27	0.35	0.39	0.42	0.43	0.44
Force, 10^4 N	10	20	30	40	50	60	70	80

20.20 A transportation engineering study was conducted to determine the proper design of bike lanes. Data were gathered on bike-lane widths and average distance between bikes and passing cars. The data from nine streets are

Distance, m	2.4	1.5	2.4	1.8	1.8	2.9	1.2	3	1.2
Lane width, m	2.9	2.1	2.3	2.1	1.8	2.7	1.5	2.9	1.5

(a) Plot the data.
(b) Fit a straight line to the data with linear regression. Add this line to the plot.
(c) If the minimum safe average distance between bikes and passing cars is considered to be 2 m, determine the corresponding minimum lane width.

20.36 You measure the voltage drop V across a resistor for a number of different values of current i. The results are

i	0.25	0.75	1.25	1.5	2.0
V	−0.45	−0.6	0.70	1.88	6.0

Use first- through fourth-order polynomial interpolation to estimate the voltage drop for $i = 1.15$. Interpret your results.