with the function `polyval` to create a vector `yp` with the value of the polynomial for each element of `xp`.

When the script file is executed, the following vector `p` is displayed in the Command Window.

```
p =
    0.0220   -0.4005    2.6138   -1.4158
```

This means that the polynomial of the third degree in Figure 8-2 has the form $0.022x^3 - 0.4005x^2 + 2.6138x - 1.4148$.

8.2.2 Curve Fitting with Functions Other than Polynomials

Many situations in science and engineering require fitting functions that are not polynomials to given data. Theoretically, any function can be used to model data within some range. For a particular data set, however, some functions provide a better fit than others. In addition, determining the best-fitting coefficients can be more difficult for some functions than for others. This section covers curve fitting with power, exponential, logarithmic, and reciprocal functions, which are commonly used. The forms of these functions are:

$y = bx^m$ (power function)

$y = be^{mx}$ or $y = b10^{mx}$ (exponential function)

$y = m\ln(x) + b$ or $y = m\log(x) + b$ (logarithmic function)

$y = \frac{1}{mx+b}$ (reciprocal function)

All of these functions can easily be fitted to given data with the `polyfit` function. This is done by rewriting the functions in a form that can be fitted with a linear polynomial ($n = 1$), which is

$$y = mx + b$$

The logarithmic function is already in this form, and the power, exponential, and reciprocal equations can be rewritten as:

$\ln(y) = m\ln(x) + \ln b$ (power function)

$\ln(y) = mx + \ln(b)$ or $\log(y) = mx + \log(b)$ (exponential function)

$\frac{1}{y} = mx + b$ (reciprocal function)

These equations describe a linear relationship between $\ln(y)$ and $\ln(x)$ for the power function, between $\ln(y)$ and x for the exponential function, between y and $\ln(x)$ or $\log(x)$ for the logarithmic function, and between $1/y$ and x for the reciprocal function. This means that the `polyfit(x,y,1)` function can be used to determine the best-fit constants m and b for best fit if, instead of `x` and `y`, the following arguments are used.

Function		`polyfit` **function form**
power	$y = bx^m$	`p=polyfit(log(x),log(y),1)`
exponential	$y = be^{mx}$ or $y = b10^{mx}$	`p=polyfit(x,log(y),1)` or `p=polyfit(x,log10(y),1)`
logarithmic	$y = m\ln(x) + b$ or $y = m\log(x) + b$	`p=polyfit(log(x),y,1)` or `p=polyfit(log10(x),y,1)`
reciprocal	$y = \frac{1}{mx+b}$	`p=polyfit(x,1./y,1)`

The result of the `polyfit` function is assigned to `p`, which is a two-element vector. The first element, `p(1)`, is the constant `m`, and the second element, `p(2)`, is `b` for the logarithmic and reciprocal functions, $\ln(b)$ or $\log(b)$ for the exponential function, and $\ln(b)$ for the power function ($b = e^{p(2)}$ or $b = 10^{p(2)}$ for the exponential function, and $b = e^{p(2)}$ for the power function).

For given data it is possible to estimate, to some extent, which of the functions has the potential for providing a good fit. This is done by plotting the data using different combinations of linear and logarithmic axes. If the data points in one of the plots appear to fit a straight line, the corresponding function can provide a good fit according to the list below.

***x* axis**	***y* axis**	**Function**
linear	linear	linear $y = mx + b$
logarithmic	logarithmic	power $y = bx^m$
linear	logarithmic	exponential $y = be^{mx}$ or $y = b10^{mx}$
logarithmic	linear	logarithmic $y = m\ln(x) + b$ or $y = m\log(x) + b$
linear	linear (plot $1/y$)	reciprocal $y = \frac{1}{mx+b}$

Other considerations in choosing a function:

- Exponential functions cannot pass through the origin.
- Exponential functions can fit only data with all positive y's or all negative y's.
- Logarithmic functions cannot model $x = 0$ or negative values of x.
- For the power function $y = 0$ when $x = 0$.
- The reciprocal equation cannot model $y = 0$.

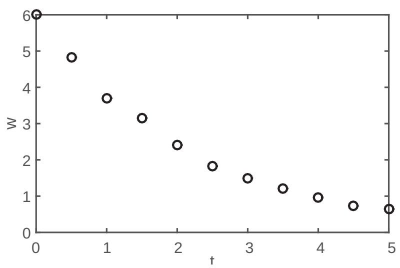
The following example illustrates the process of fitting a function to a set of data points.

Sample Problem 8-2: Fitting an equation to data points

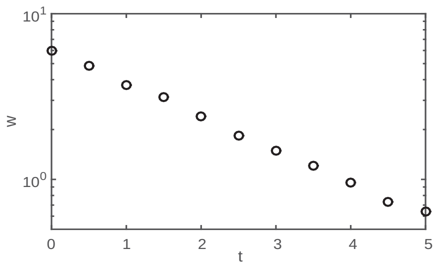
The following data points are given. Determine a function $w = f(t)$ (t is the independent variable, w is the dependent variable) with a form discussed in this section that best fits the data.

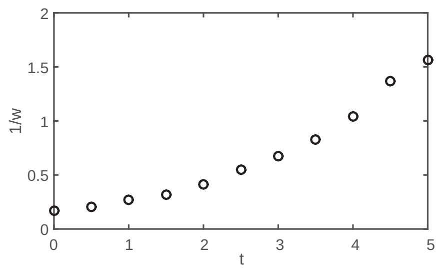
t	0.0	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0
w	6.00	4.83	3.70	3.15	2.41	1.83	1.49	1.21	0.96	0.73	0.64

Solution



The data is first plotted with linear scales on both axes. The figure indicates that a linear function will not give the best fit since the points do not appear to line up along a straight line. From the other possible functions, the logarithmic function is excluded since for the first point $t = 0$, and the power function is excluded since at $t = 0$, $w \neq 0$. To check if the other two functions (exponential and reciprocal) might give a better fit, two additional plots, shown below, are made. The plot on the left has a log scale on the vertical axis and linear horizontal axis. In the plot on the right, both axes have linear scales, and the quantity $1/w$ is plotted on the vertical axis.





In the left figure, the data points appear to line up along a straight line. This indicates that an exponential function of the form $y = be^{mx}$ can give a good fit to the data. A program in a script file that determines the constants b and m, and that plots the data points and the function is given below.

```
t=0:0.5:5;
```
Create vectors `t` and `w` with the coordinates of the data points.

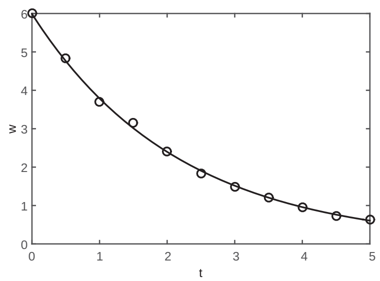
```
w=[6 4.83 3.7 3.15 2.41 1.83 1.49 1.21 0.96 0.73 0.64];
p=polyfit(t,log(w),1);
```
Use the `polyfit` function with `t` and `log(w)`.

```
m=p(1)
b=exp(p(2))                Determine the coefficient b.
tm=0:0.1:5;                Create a vector tm to be used for plotting the polynomial.
wm=b*exp(m*tm);            Calculate the function value at each element of tm.
plot(t,w,'o',tm,wm)        Plot the data points and the function.
```

When the program is executed, the values of the constants m and b are displayed in the Command Window.

```
m =
   -0.4580
b =
    5.9889
```

The plot generated by the program, which shows the data points and the function (with axis labels added with the Plot Editor) is



It should be pointed out here that in addition to the power, exponential, logarithmic, and reciprocal functions that are discussed in this section, many other functions can be written in a form suitable for curve fitting with the `polyfit` function. One example where a function of the form $y = e^{(a_2x^2 + a_1x + a_0)}$ is fitted to data points using the `polyfit` function with a third-order polynomial is described in Sample Problem 8-7.

8.3 *INTERPOLATION*

Interpolation is the estimation of values between data points. MATLAB has interpolation functions that are based on polynomials, which are described in this section, and on Fourier transformation, which is outside the scope of this book. In one-dimensional interpolation, each point has one independent variable (x) and one dependent variable (y). In two-dimensional interpolation, each point has two independent variables (x and y) and one dependent variable (z).