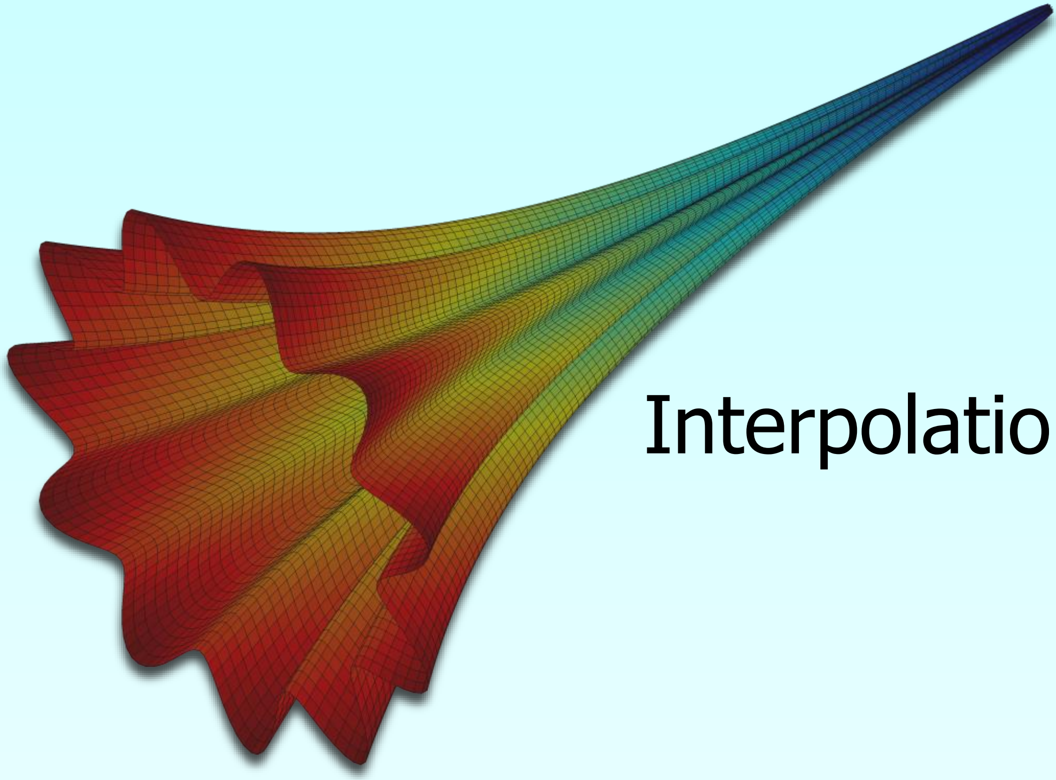
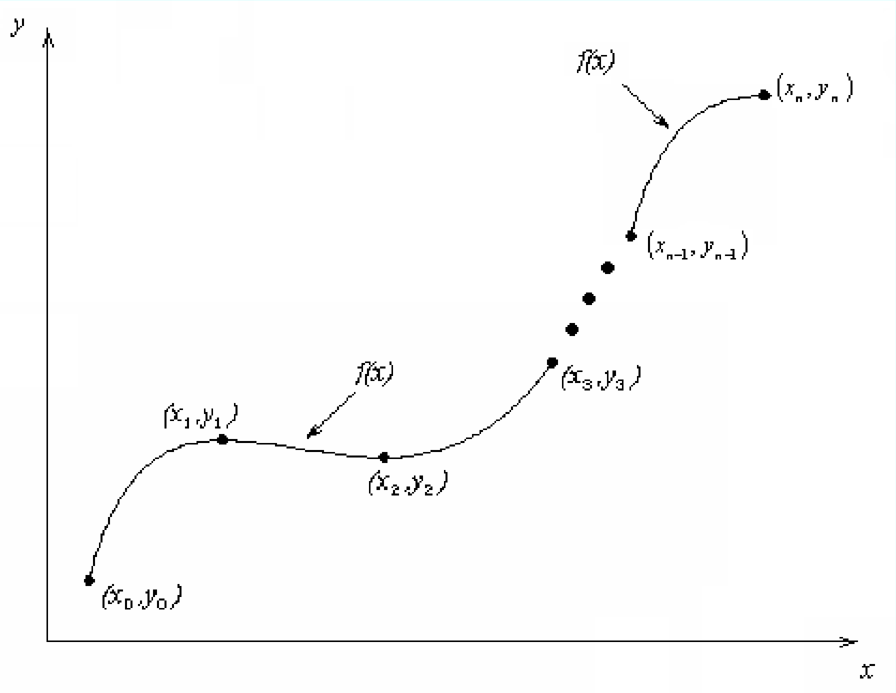


Interpolation

What is Interpolation ?

Given (x_0,y_0), (x_1,y_1), …… (x_n,y_n), find the value of 'y' at a value of 'x' that is not given.



14.1 Interpolation Problem Statement

- Assume we have a data set consisting of independent data values, *xi* , and dependent data values, *yi* , where *i* = 1*, . . . , n*. We would like to find an estimation function $\hat{y}(x)$ such that $\hat{y}(xi) = yi$ for every point in our data set. This means the estimation function goes through our data points.
- Given a new $x*$, we can **interpolate** its function value using $\hat{y}(x*)$. In this context, $\hat{y}(x)$ is called an **interpolation function**. Figure 14.1 shows the interpolation problem statement.
- For example, our data may consist of *(x, y)* coordinates of a car over time. Since motion is restricted to the maneuvering physics of the car, we can expect that the points between the *(x, y)* coordinates in our set will be “smooth” rather than jagged.

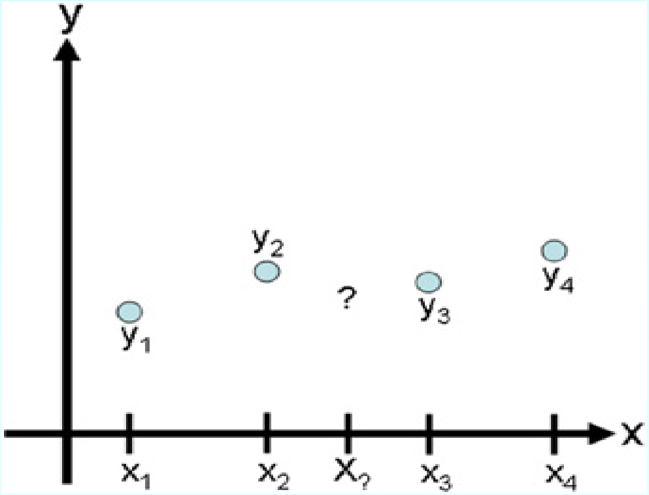


FIGURE 14.1

Illustration of the interpolation problem: estimate the value of a function in between data points.

Polynomial Interpolation

- You will frequently have occasions to estimate intermediate values between precise data points.
- The function you use to interpolate must pass through the actual data points - this makes interpolation more restrictive than fitting.
- The most common method for this purpose is polynomial interpolation, where an (n-1)th order polynomial is solved that passes through n data points:

$$f(x) = a_1 + a_2 x + a_3 x^2 + \cdots + a_n x^{n-1}$$

MATLAB version :

$$f(x) = p_1 x^{n-1} + p_2 x^{n-2} + \cdots + p_{n-1} x + p_n$$

14.2 Linear Interpolation

In **linear interpolation**, the estimated point is assumed to lie on the line joining the nearest points to the left and right. Assume, without loss of generality, that the x-data points are in ascending order; that is, $x_i < x_{i+1}$, and let x be a point such that $x_i < x < x_{i+1}$. Then the linear interpolation at x is

$$\hat{y}(x) = y_i + \frac{(y_{i+1} - y_i)(x - x_i)}{(x_{i+1} - x_i)}.$$

TRY IT! Find the linear interpolation at `x = 1.5` based on the data `x = [0 1 2]`, `y = [1 3 2]`. Verify the result using MATLAB's function `interp1`. (See Figure 14.2.)

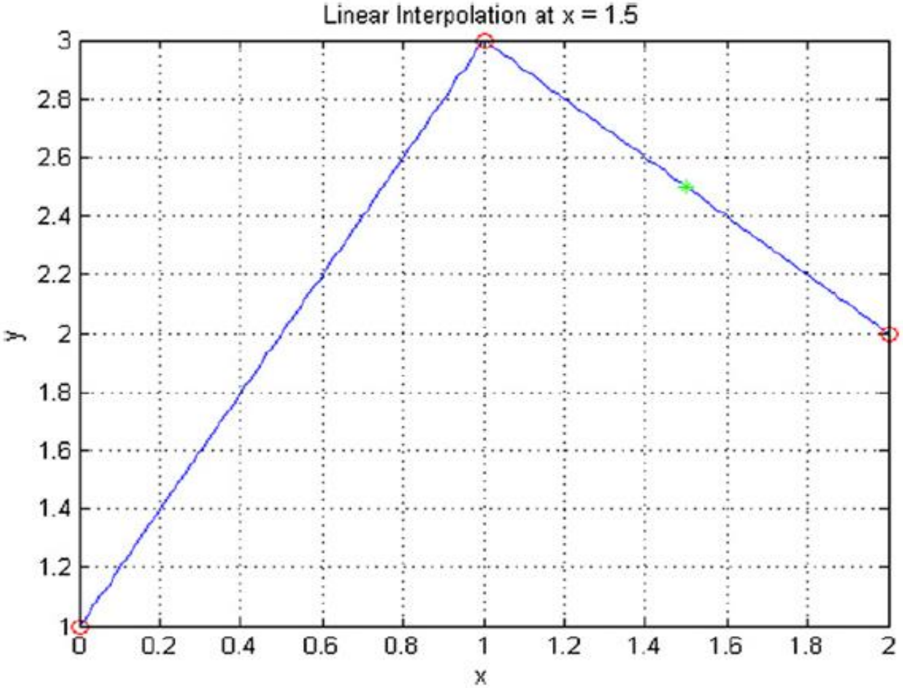
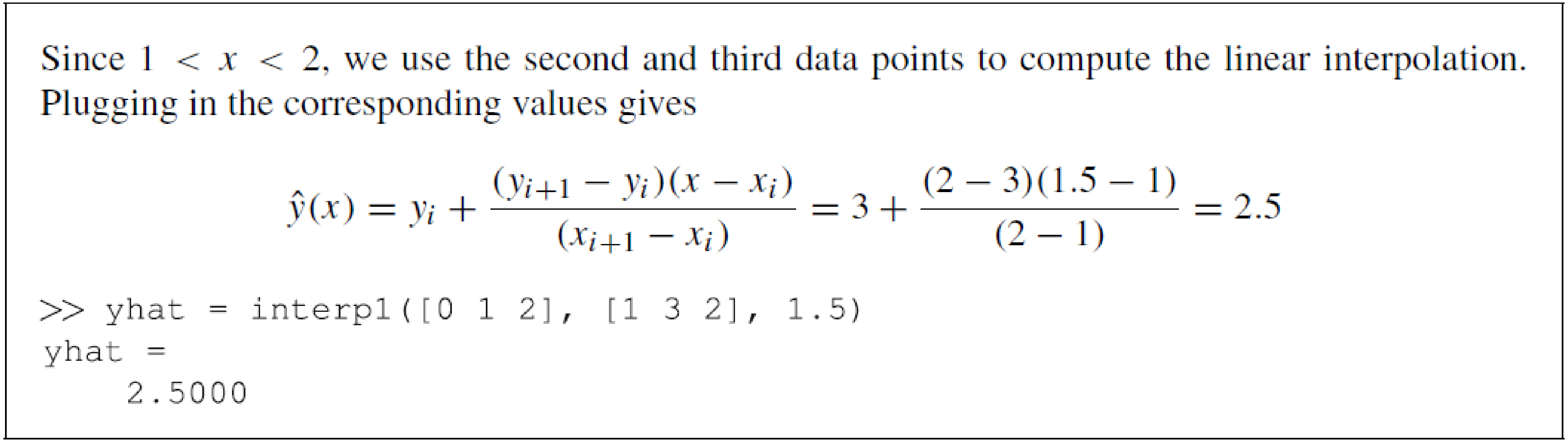


FIGURE 14.2

Linear interpolation of the points $x = (0, 1, 2)$ and $y = (1, 3, 2)$.



Since $1 < x < 2$, we use the second and third data points to compute the linear interpolation. Plugging in the corresponding values gives

$$\hat{y}(x) = y_i + \frac{(y_{i+1} - y_i)(x - x_i)}{(x_{i+1} - x_i)} = 3 + \frac{(2-3)(1.5-1)}{(2-1)} = 2.5$$

```
>> yhat = interp1([0 1 2], [1 3 2], 1.5)
yhat =
    2.5000
```

Determining Coefficients

- Since polynomial interpolation provides as many basis functions as there are data points (*n*), the polynomial coefficients can be found exactly using linear algebra.

- MATLAB’s built in polyfit and polyval commands can also be used - all that is required is making sure the order of the fit for *n* data points is *n*-1.

Direct Method

Given ‘n+1’ data points (x_0,y_0), (x_1,y_1),………….. (x_n,y_n), pass a polynomial of order ‘n’ through the data as given below:

$$y = a_0 + a_1 x + \dots\dots\dots\dots\dots\dots\dots + a_n x^n .$$

where a_0, a_1,………………. a_n are real constants.

- Set up ‘n+1’ equations to find ‘n+1’ constants.
- To find the value ‘y’ at a given value of ‘x’, simply substitute the value of ‘x’ in the above polynomial.

Example 1

The upward velocity of a rocket is given as a function of time in Table 1.

Find the velocity at t=16 seconds using the direct method for linear interpolation.

Table 1 Velocity as a function of time.

$t, (\text{s})$	$v(t), (\text{m/s})$
0	0
10	227.04
15	362.78
20	517.35
22.5	602.97
30	901.67

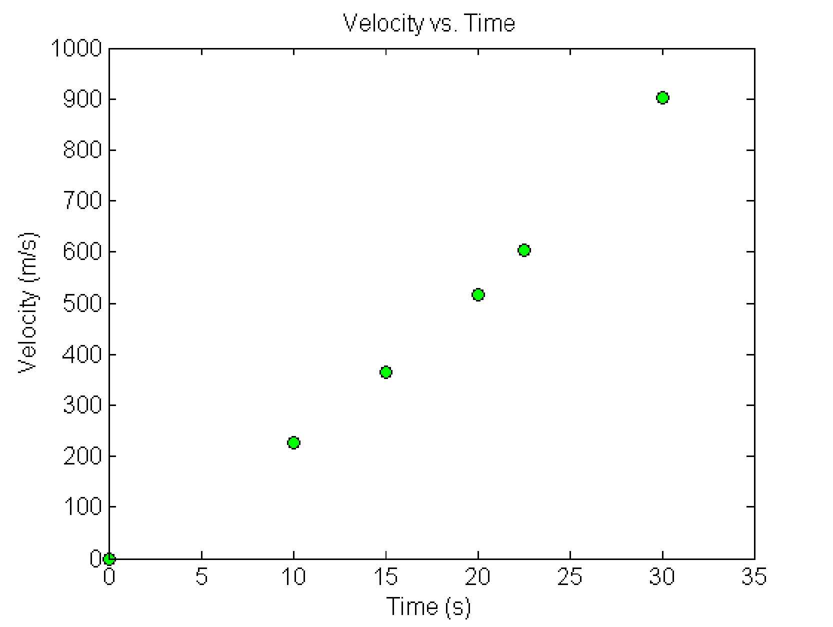


Figure 2 Velocity vs. time data for the rocket example

Linear Interpolation

$$v(t) = a_0 + a_1 t$$

$$v(15) = a_0 + a_1(15) = 362.78$$

$$v(20) = a_0 + a_1(20) = 517.35$$

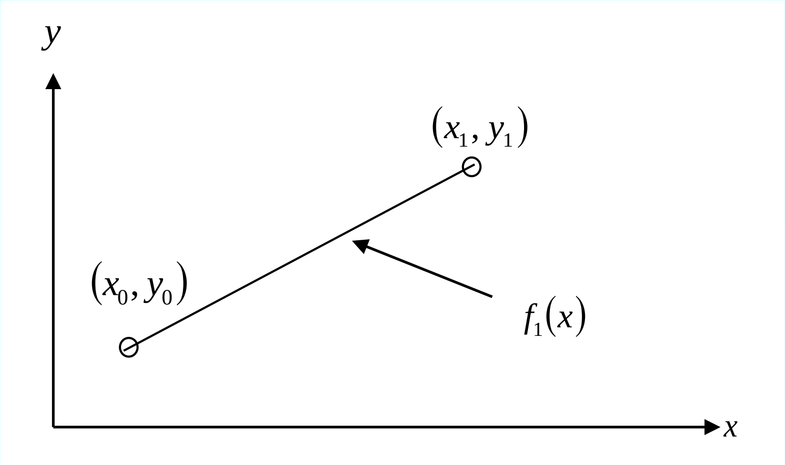


Figure 3 Linear interpolation.

Solving the above two equations gives,

$$a_0 = -100.93 \quad a_1 = 30.914$$

Hence

$$v(t) = -100.93 + 30.914t,\ 15 \le t \le 20.$$

$$v(16) = -100.93 + 30.914(16) = 393.7 \text{ m/s}$$

Newton Interpolating Polynomials

- Another way to express a polynomial interpolation is to use *Newton's interpolating polynomial*.

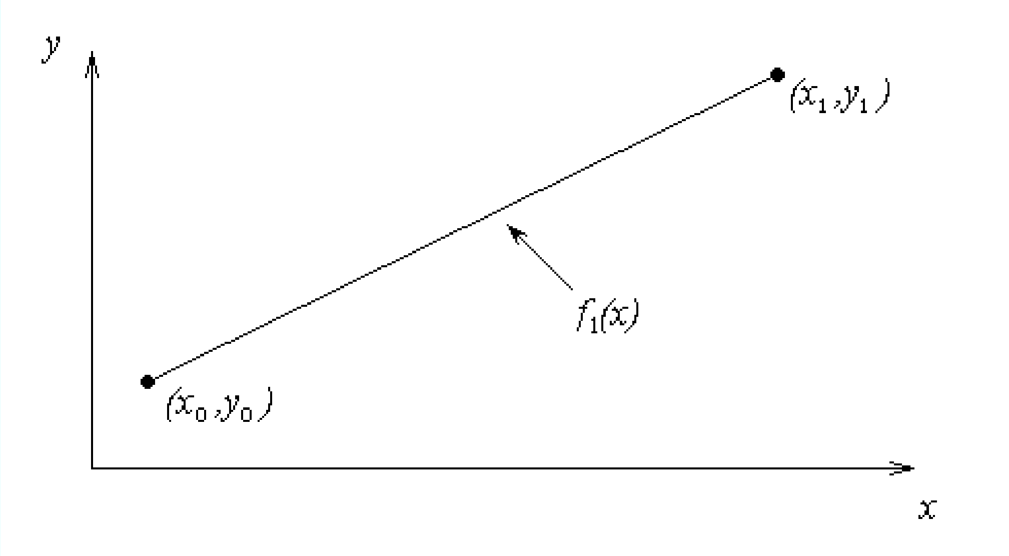
- The differences between a simple polynomial and Newton's interpolating polynomial for first and second order interpolations are:

Order	Simple	Newton
$1st$	$f_1(x) = a_1 + a_2 x$	$f_1(x) = b_1 + b_2(x - x_1)$
$2nd$	$f_2(x) = a_1 + a_2 x + a_3 x^2$	$f_2(x) = b_1 + b_2(x - x_1) + b3(x - x_1)(x - x_2)$

Newton's Divided Difference Method

Linear interpolation: Given $(x_0, y_0), (x_1, y_1),$ pass a linear interpolant through the data

$$f_1(x) = b_0 + b_1(x - x_0)$$



where

$$b_0 = f(x_0)$$

$$b_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

Example

The upward velocity of a rocket is given as a function of time in Table 1. Find the velocity at t=16 seconds using the Newton Divided Difference method for linear interpolation.

Table. Velocity as a function of time

t (s)	$v(t)$ (m/s)
0	0
10	227.04
15	362.78
20	517.35
22.5	602.97
30	901.67

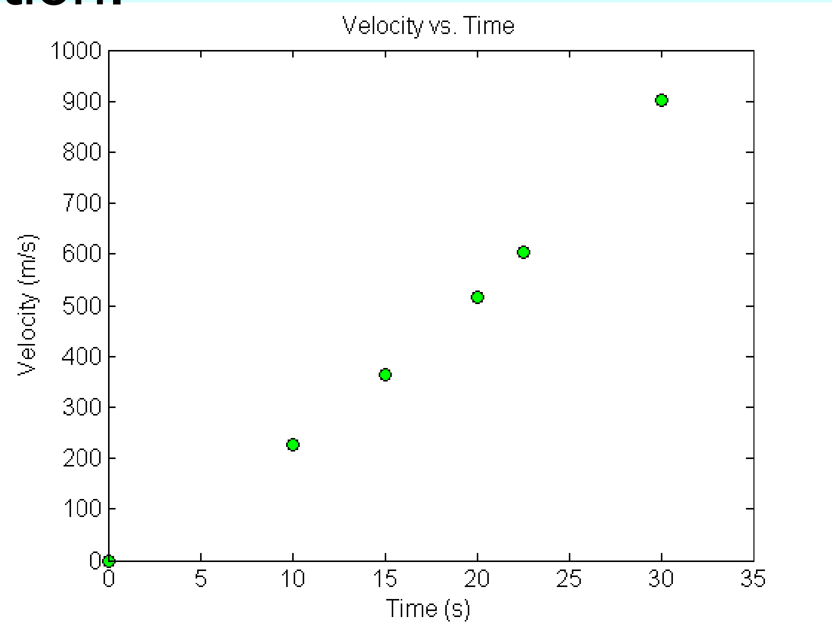


Figure. Velocity vs. time data for the rocket example

Linear Interpolation

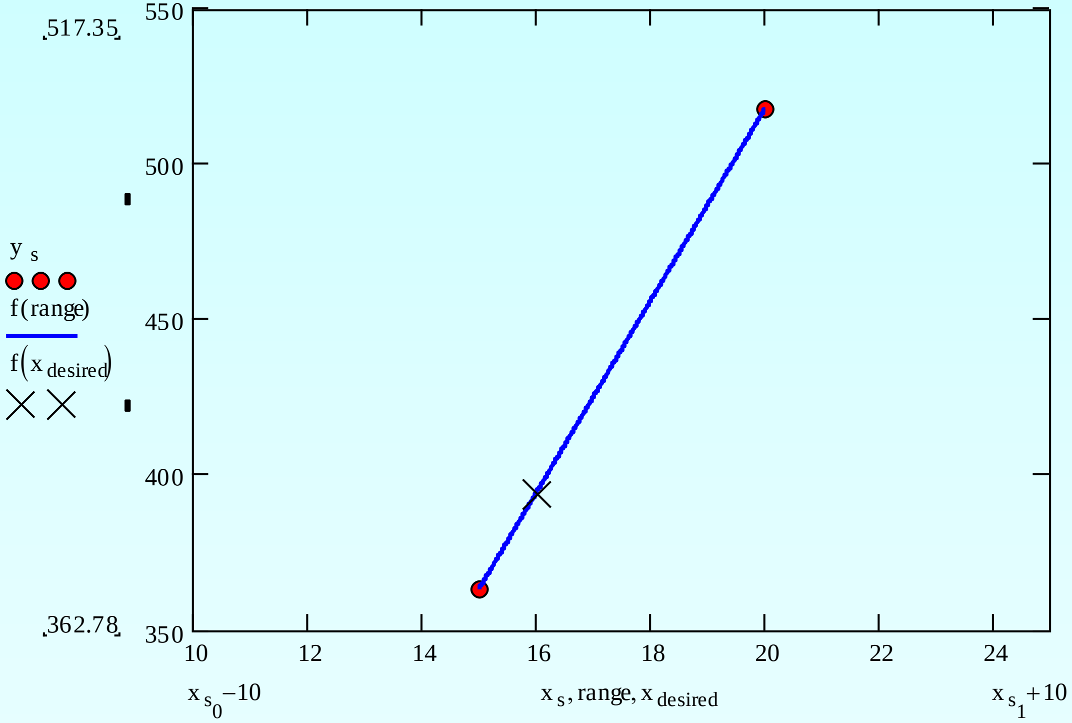
$$v(t) = b_0 + b_1(t - t_0)$$

$$t_0 = 15,\ v(t_0) = 362.78$$

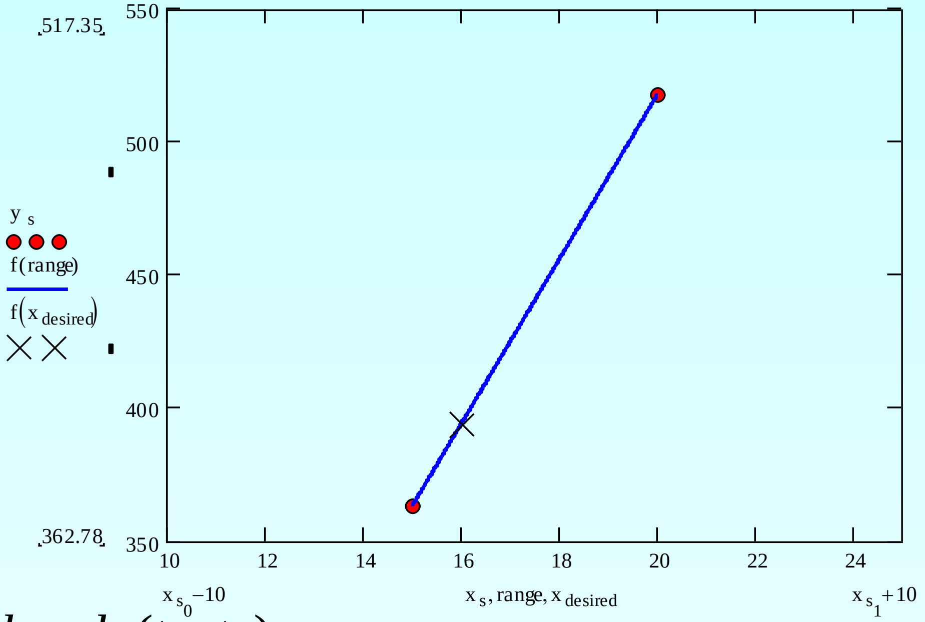
$$t_1 = 20,\ v(t_1) = 517.35$$

$$b_0 = v(t_0) = 362.78$$

$$b_1 = \frac{v(t_1) - v(t_0)}{t_1 - t_0} = 30.914$$



Linear Interpolation (contd)



$$v(t) = b_0 + b_1(t - t_0)$$

$$= 362.78 + 30.914(t - 15),\ 15 \le t \le 20$$

At $t = 16$

$$v(16) = 362.78 + 30.914(16 - 15)$$

$$= 393.69 \text{ m/s}$$

Lagrangian Interpolation

Lagrangian interpolating polynomial is given by

$$f_n(x) = \sum_{i=0}^{n} L_i(x) f(x_i)$$

where 'n' in $f_n(x)$ stands for the n^{th} order polynomial that approximates the function $y = f(x)$ given at $(n+1)$ data points as $(x_0, y_0), (x_1, y_1), \ldots\ldots, (x_{n-1}, y_{n-1}), (x_n, y_n)$, and

$$L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j}$$

$L_i(x)$ is a weighting function that includes a product of $(n-1)$ terms with terms of $j = i$ omitted.

Example

The upward velocity of a rocket is given as a function of time in Table 1. Find the velocity at t=16 seconds using the Lagrangian method for linear interpolation.

Table Velocity as a function of time

t (s)	$v(t)$ (m/s)
0	0
10	227.04
15	362.78
20	517.35
22.5	602.97
30	901.67

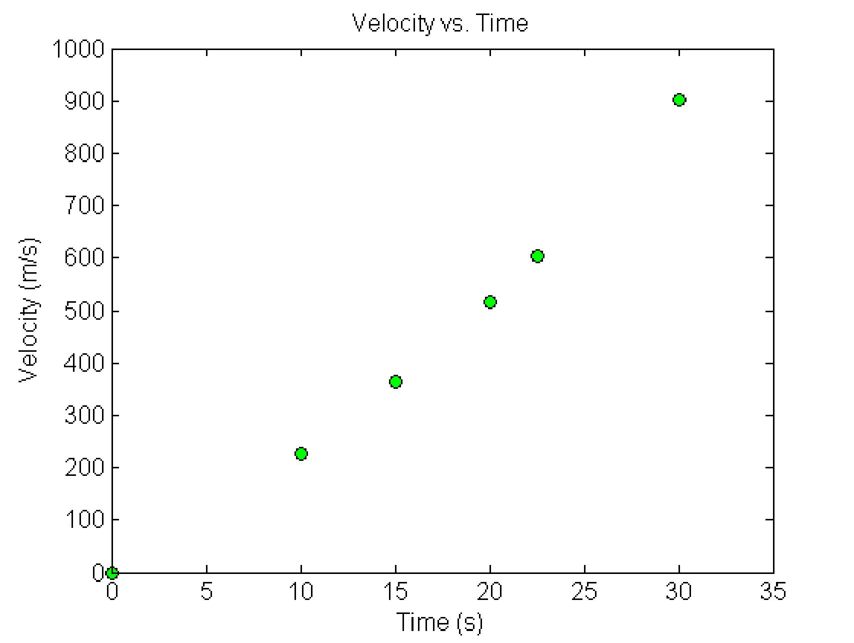


Figure. Velocity vs. time data for the rocket example

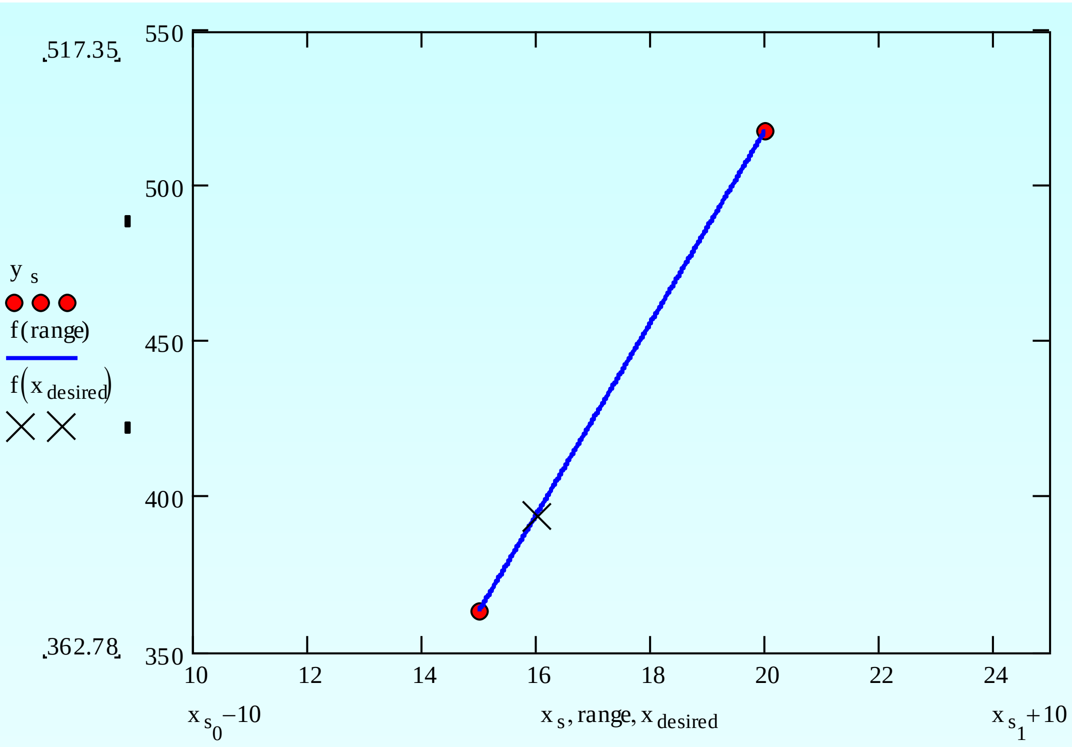
Linear Interpolation

$$v(t) = \sum_{i=0}^{1} L_i(t) v(t_i)$$

$$= L_0(t) v(t_0) + L_1(t) v(t_1)$$

$$t_0 = 15, v(t_0) = 362.78$$

$$t_1 = 20, v(t_1) = 517.35$$



Linear Interpolation (contd)

$$L_0(t) = \prod_{\substack{j=0 \\ j\neq 0}}^{1} \frac{t - t_j}{t_0 - t_j} = \frac{t - t_1}{t_0 - t_1}$$

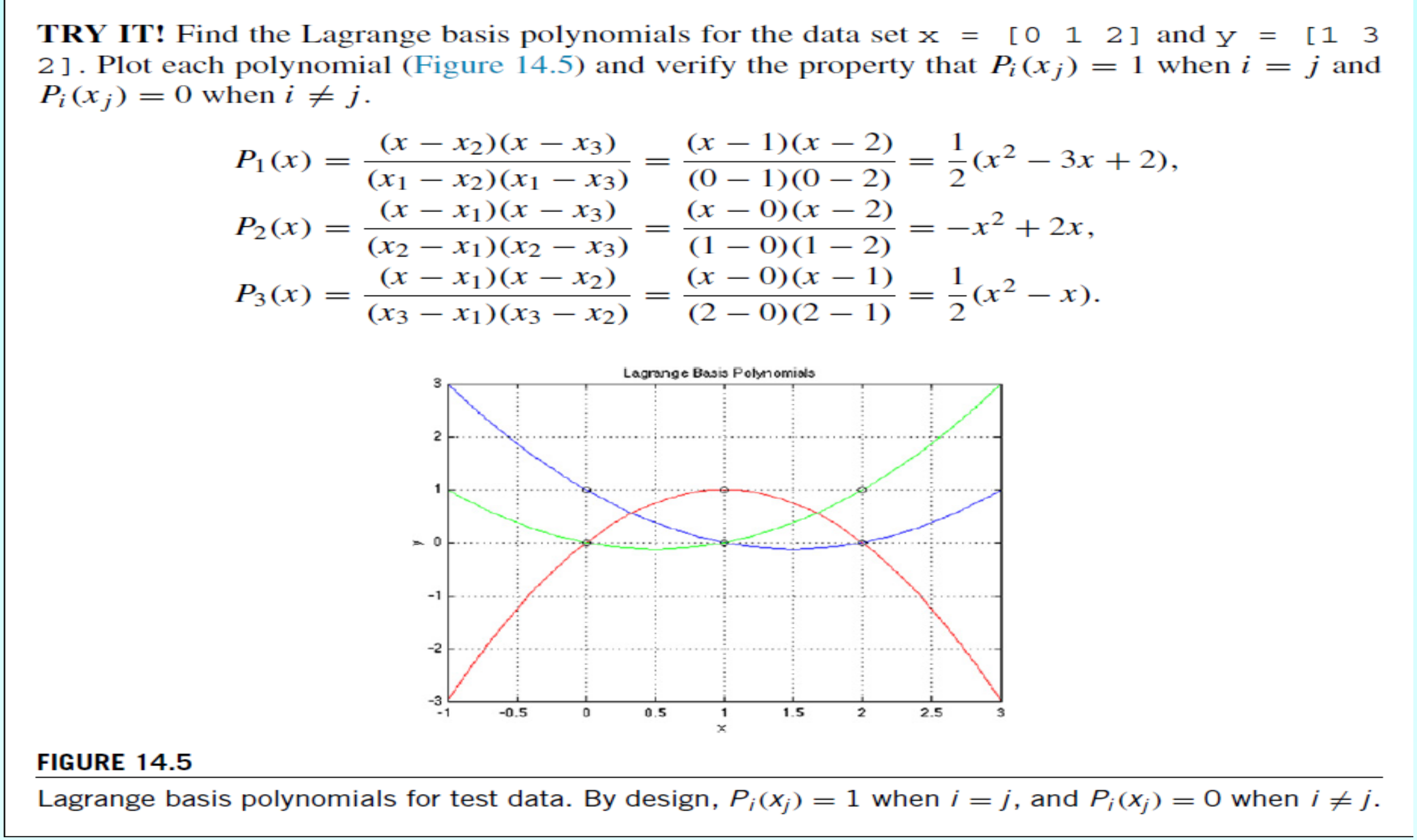
$$L_1(t) = \prod_{\substack{j=0 \\ j\neq 1}}^{1} \frac{t - t_j}{t_1 - t_j} = \frac{t - t_0}{t_1 - t_0}$$

$$v(t) = \frac{t - t_1}{t_0 - t_1} v(t_0) + \frac{t - t_0}{t_1 - t_0} v(t_1) = \frac{t - 20}{15 - 20}(362.78) + \frac{t - 15}{20 - 15}(517.35)$$

$$v(16) = \frac{16 - 20}{15 - 20}(362.78) + \frac{16 - 15}{20 - 15}(517.35)$$

$$= 0.8(362.78) + 0.2(517.35)$$

$$= 393.7 \text{ m/s.}$$



TRY IT! Find the Lagrange basis polynomials for the data set `x = [0 1 2]` and `y = [1 3 2]`. Plot each polynomial (Figure 14.5) and verify the property that $P_i(x_j) = 1$ when $i = j$ and $P_i(x_j) = 0$ when $i \neq j$.

$$P_1(x) = \frac{(x - x_2)(x - x_3)}{(x_1 - x_2)(x_1 - x_3)} = \frac{(x - 1)(x - 2)}{(0 - 1)(0 - 2)} = \frac{1}{2}(x^2 - 3x + 2),$$

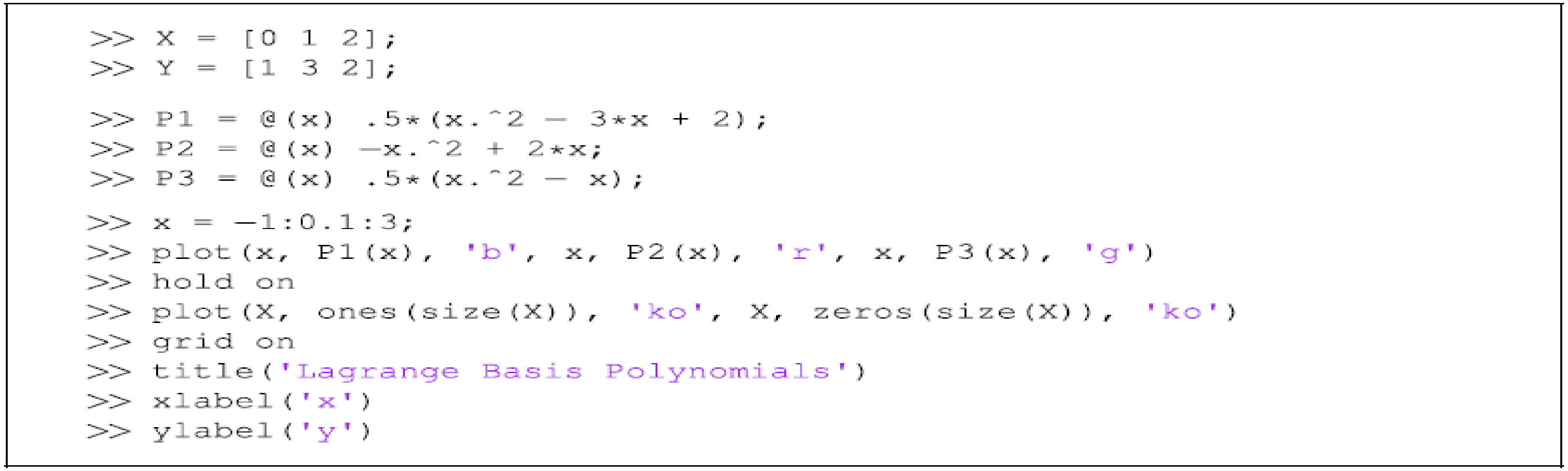
$$P_2(x) = \frac{(x - x_1)(x - x_3)}{(x_2 - x_1)(x_2 - x_3)} = \frac{(x - 0)(x - 2)}{(1 - 0)(1 - 2)} = -x^2 + 2x,$$

$$P_3(x) = \frac{(x - x_1)(x - x_2)}{(x_3 - x_1)(x_3 - x_2)} = \frac{(x - 0)(x - 1)}{(2 - 0)(2 - 1)} = \frac{1}{2}(x^2 - x).$$

Lagrange Basis Polynomials
3
2
1
0
-1
-2
-3
y
-1
-0.5
0
0.5
1
1.5
2
2.5
3
x

FIGURE 14.5

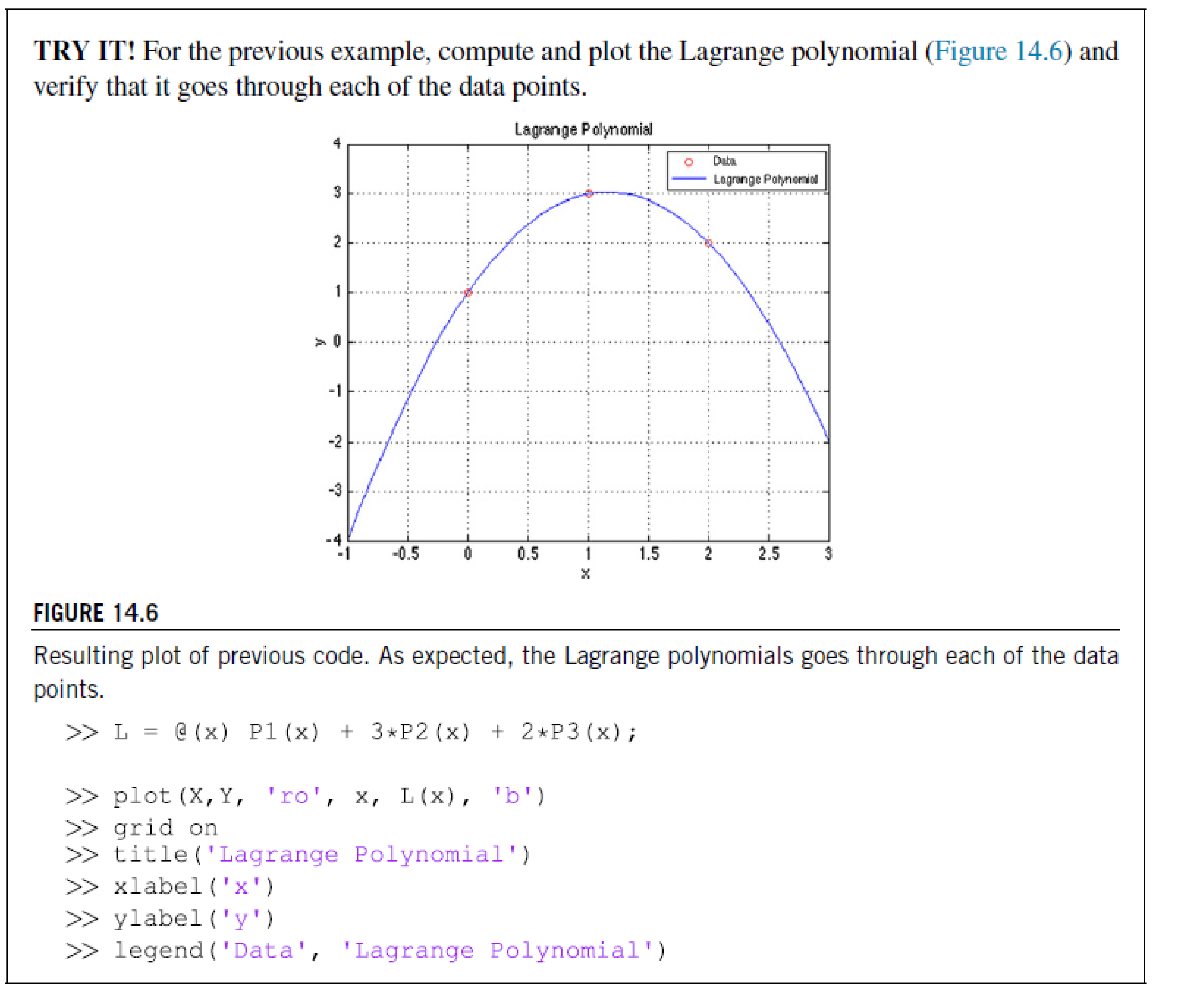
Lagrange basis polynomials for test data. By design, $P_i(x_j) = 1$ when $i = j$, and $P_i(x_j) = 0$ when $i \neq j$.



```
>> X = [0 1 2];
>> Y = [1 3 2];

>> P1 = @(x) .5*(x.^2 - 3*x + 2);
>> P2 = @(x) -x.^2 + 2*x;
>> P3 = @(x) .5*(x.^2 - x);

>> x = -1:0.1:3;
>> plot(x, P1(x), 'b', x, P2(x), 'r', x, P3(x), 'g')
>> hold on
>> plot(X, ones(size(X)), 'ko', X, zeros(size(X)), 'ko')
>> grid on
>> title('Lagrange Basis Polynomials')
>> xlabel('x')
>> ylabel('y')
```



TRY IT! For the previous example, compute and plot the Lagrange polynomial (Figure 14.6) and verify that it goes through each of the data points.

FIGURE 14.6

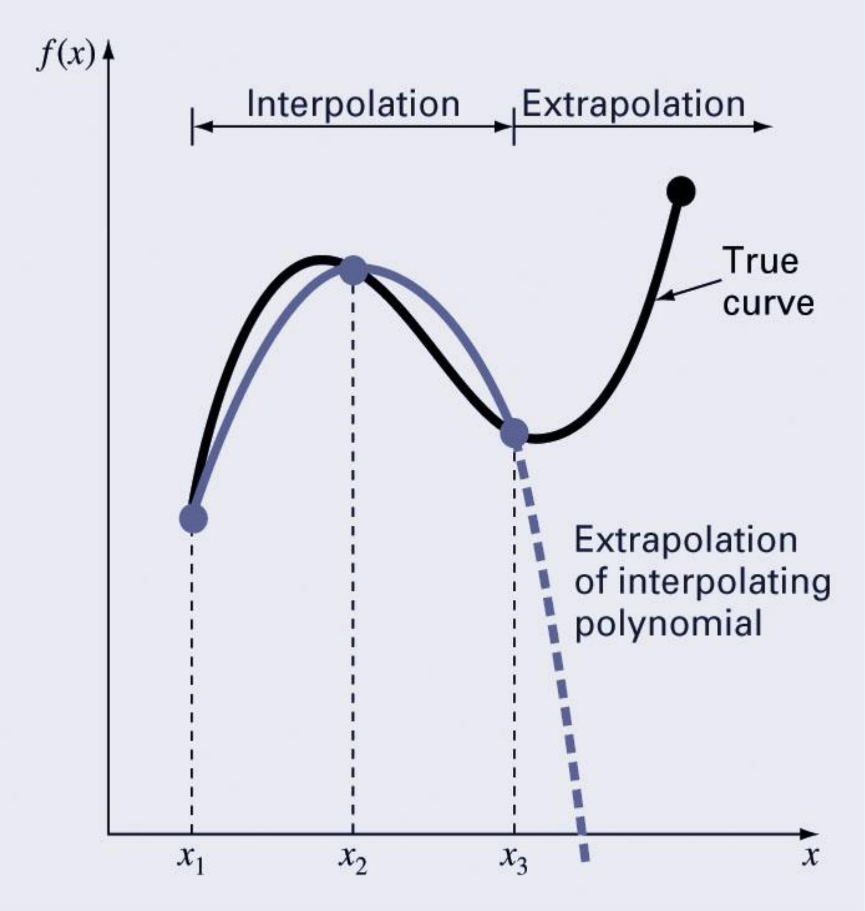
Resulting plot of previous code. As expected, the Lagrange polynomials goes through each of the data points.

```
>> L = @(x) P1(x) + 3*P2(x) + 2*P3(x);

>> plot(X,Y, 'ro', x, L(x), 'b')
>> grid on
>> title('Lagrange Polynomial')
>> xlabel('x')
>> ylabel('y')
>> legend('Data', 'Lagrange Polynomial')
```

Extrapolation

- *Extrapolation* is the process of estimating a value of $f(x)$ that lies outside the range of the known base points x_1, x_2, …, x_n.
- Extrapolation represents a step into the unknown, and extreme care should be exercised when extrapolating!


$f(x)$
Interpolation
Extrapolation
True curve
Extrapolation of interpolating polynomial
x_1
x_2
x_3
x