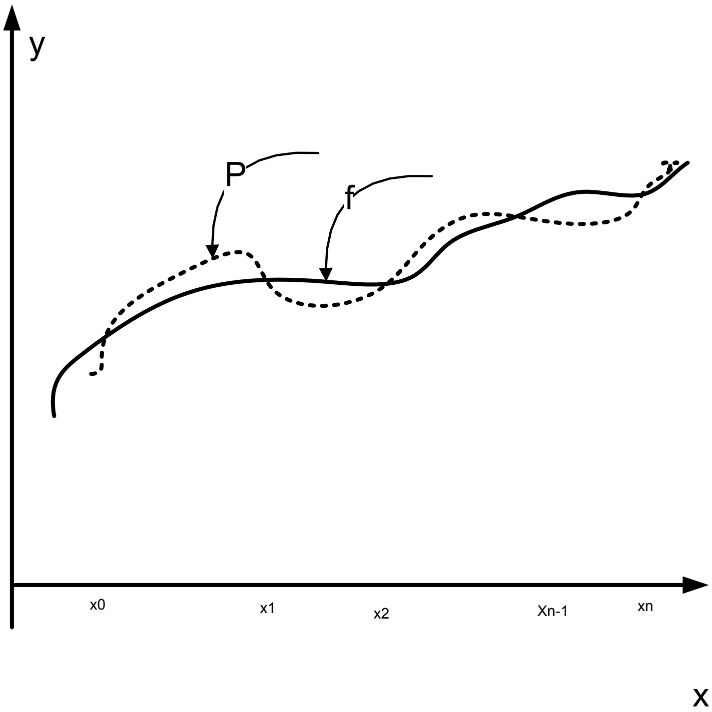
Interpolation And Lagrange Polynomial



Interpolation means to estimate a missing function value by taking a weighted average of known function values at neighboring points. One of the simplest methods for constructing a polynomial approximation to a given function f(x) is to require that the error vanishes at an appropriate number of points, i.e. that the approximation is exact at a certain number of points.

Consider the problem of determining a polynomial of degree 1 that passes through the distinct points (x_0, y_0) and (x_1, y_1).

This problem is the same as approximating a function f for which $f(x_0)=y_0$ and $f(x_1)=y_1$, by means a first degree polynomial (Linear interpolation) interpolating, or agreeing with, the values of f at given points.

The slope between (x_0, y_0) and (x_1, y_1) is

$$m = {(y_1 - y_0)} \Big/ {(x_1 - x_0)}$$

and the point slope formula for the line

$$y = m(x - x_0) + y_0$$

can be rearranged

$$y = P(x) = y_0 + (y_1 - y_0)\frac{(x - x_0)}{(x_1 - x_0)}.$$

Evaluation of P(x) at x_0 and x_1 produces y_0 and y_1, respectively:

$$P(x_0) = y_0 + (y_1 - y_0)\frac{(x_0 - x_0)}{(x_1 - x_0)} = y_0 + (y_1 - y_0)(0) = y_0$$

$$P(x_1) = y_0 + (y_1 - y_0)\frac{(x_1 - x_0)}{(x_1 - x_0)} = y_0 + (y_1 - y_0)(1) = y_1.$$

The French mathematician Joseph Louis Lagrange noticed that it could be written as

$$\begin{aligned} P_1(x) &= y_0 \frac{(x - x_1)}{(x_0 - x_1)} + y_1 \frac{(x - x_0)}{(x_1 - x_0)} \\ &= y_0 L_{1,0}(x) + y_1 L_{1,1}(x) \end{aligned}$$

The quotients are denoted by

$$L_{1,0}(x) = \frac{(x - x_1)}{(x_0 - x_1)} \quad \textbf{and} \quad L_{1,1}(x) = \frac{(x - x_0)}{(x_1 - x_0)}.$$

Example: **Consider the graph** $y = \cos(x)$ **over [0, 1.2].**

i. **Use the nodes x_0=0 and x_1=1.2 to construct a linear interpolation polynomial $P_1(x)$**

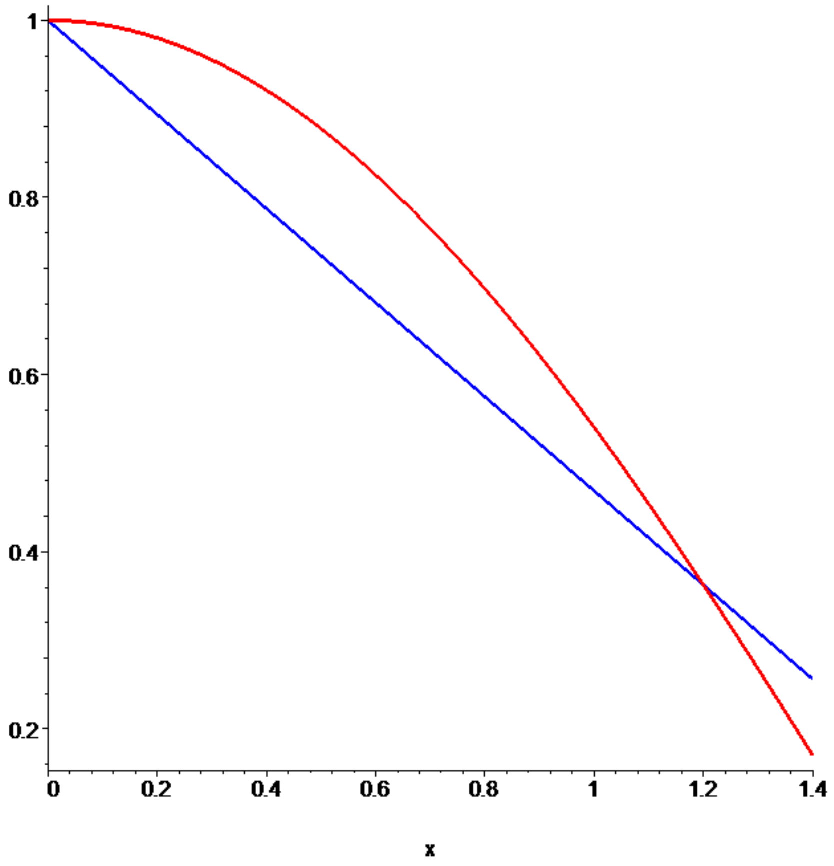
$$P_1(x) = \cos(0)\frac{(x-1.2)}{(0-1.2)} + \cos(1.2)\frac{(x-0)}{(1.2-0)}$$

$$= 1.000\frac{(x-1.2)}{(0-1.2)} + 0.362358\frac{(x-0)}{(1.2-0)}$$

$$= -0.833333(x-1.2) + 0.301965(x-0).$$

$$= -0.531335x + 0.99996$$

```
> plot([cos(x),-0.833333*(x-1.2)+0.301965*x],x=0..1.4,color=[red,blue]);
```



LAGRANGE
Interpolation Polynomial

The interpolating polynomial is easily described now that the form of $L_{n,k}$ is known. This polynomial is called the Lagrange interpolating polynomial and defined in the following theorem.

Theorem:* *If $x_0, x_1, \ldots x_n$ are (n+1) distinct numbers and f is a function whose values are given at these numbers, then there exists a unique polynomial P of degree at most n with the property that

$$f(x_k) = P(x_k)$$

For each k=0, 1, 2,…, n.

This polynomial is given by

$$P_n(x) = f(x_0)L_{n,o}(x) + \ldots + f(x_n)L_{n,n}(x)$$
$$= \sum_{k=0}^{n} f(x_k)L_{n,k}(x).$$

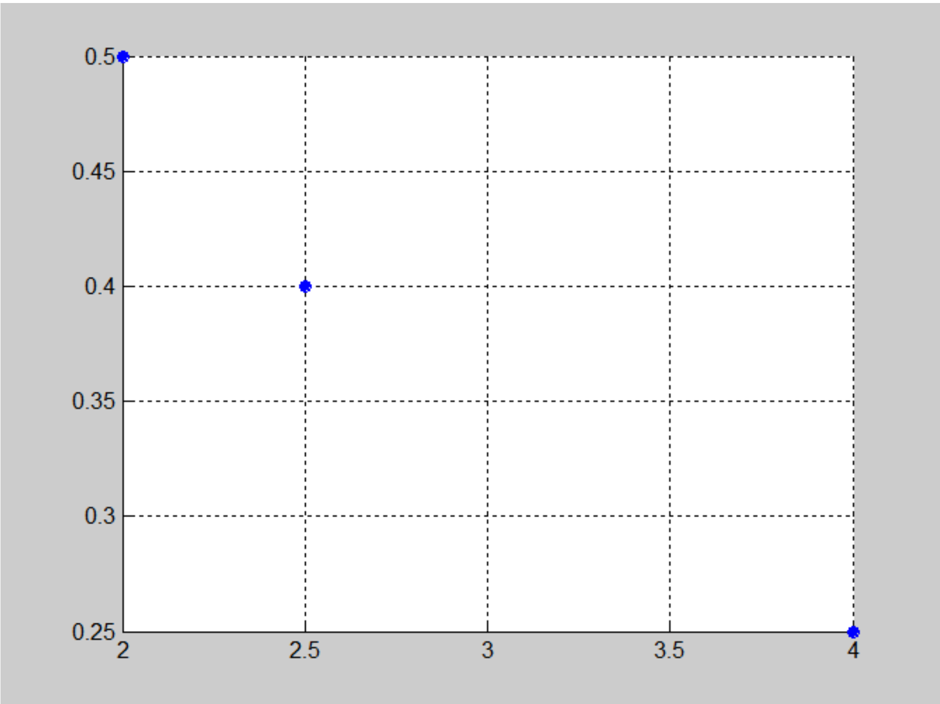
where

$$L_{n,k}(x) = \frac{(x-x_0)\ldots(x-x_{k-1})(x-x_{k+1})\ldots(x-x_n)}{(x_k-x_0)\ldots(x_k-x_{k-1})\ldots(x_k-x_n)}$$
$$= \prod_{\substack{i=0 \\ i\neq k}}^{n} \frac{(x-x_i)}{(x_k-x_i)}.$$

Example: **Suppose that we have the following data pairs – x values and f(x) values- <u>known function.</u>**

x	f(x)=1/x
2	0.50
2.5	0.40
4.0	0.25

Find the second-degree interpolating polynomial to find the interpolated value for x=3.0.



Solution:

To find the second degree interpolating polynomials for f(x)=1/x requires that we first determine the coefficient polynomials L_0, L_1, and L_2:

$$L_{2,0}(x)=\frac{(x-2.5)(x-4)}{(2-2.5)(2-4)}=x^2-6.5x+10,$$

$$L_{2,1}(x)=\frac{(x-2)(x-4)}{(2.5-2)(2.5-4)}=\frac{1}{3}(-4x^2+24x-32)$$

$$L_{2,2}(x)=\frac{(x-2)(x-2.5)}{(4-2)(4-2.5)}=\frac{1}{3}(x^2-4.5x+5).$$

Then

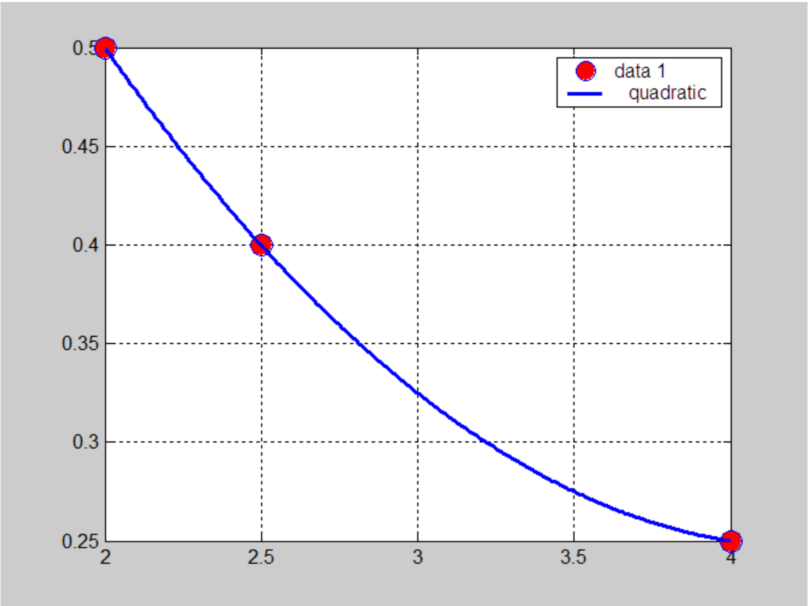
$$\begin{aligned}P_2(x)&=\sum_{k=0}^{2} f(x_k)L_{2,k}(x)\\&=0.5(x^2-6.5x+10)+\frac{0.4}{3}(-4x^2+24x-32)\\&\quad+\frac{0.25}{3}(x^2-4.5x+5)\\&=0.05x^2-0.425x+1.15.\end{aligned}$$

Resulting Polynomial

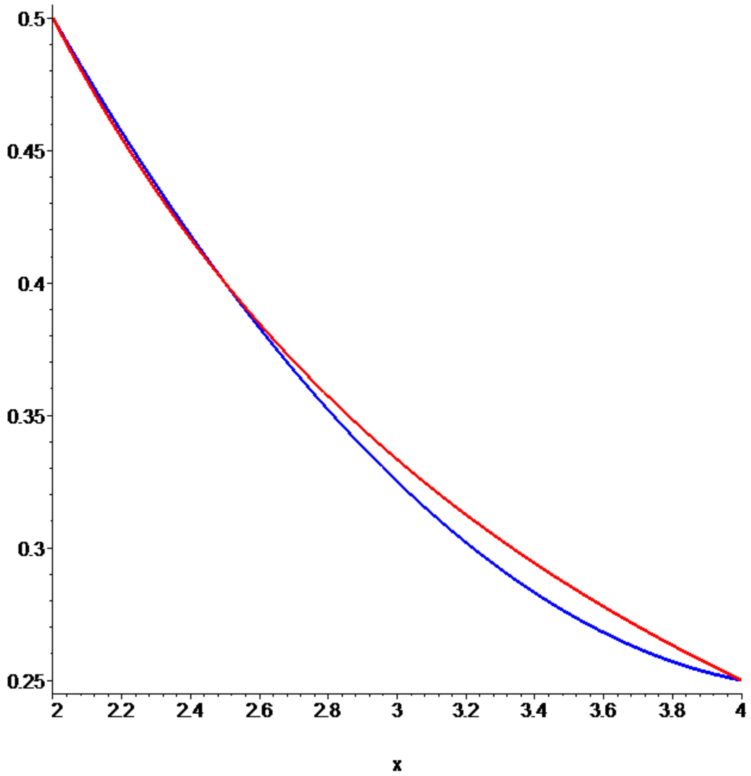
$$P_2(x)=0.05x^2-0.425x+1.15.$$

An approximation to f(3) is

$$f(3)\approx P_2(3)=0.325$$



>> plot(x,y,'bo');grid on

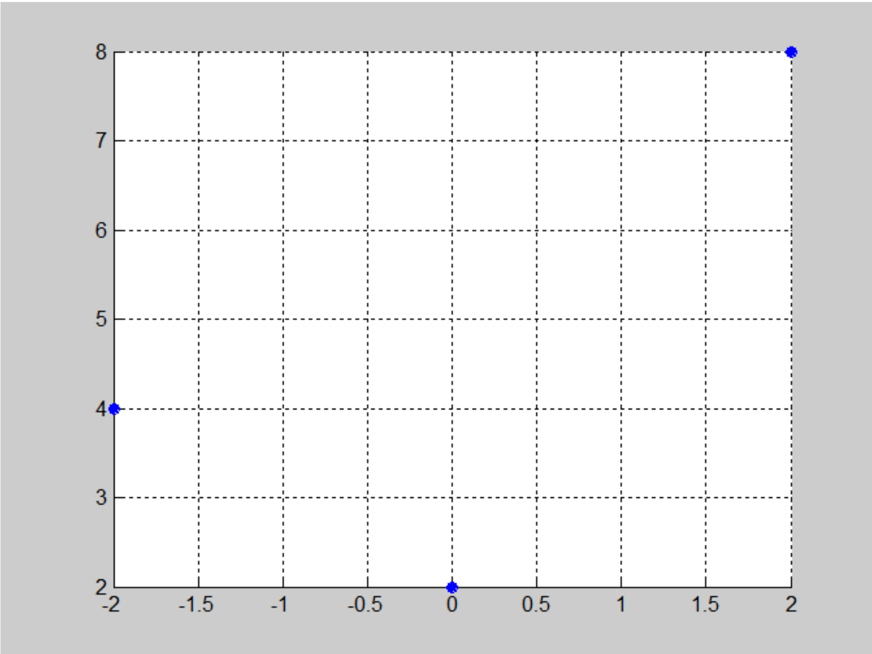



```
> plot([1/x,0.05*x^2-0.425*x+1.15],x=2..4,color=[red,blue]);
```

Example: **Suppose that we have the following data pairs – x values and f(x) values- unknown function.**

x	y
-2	**4**
0	**2**
2	**8**

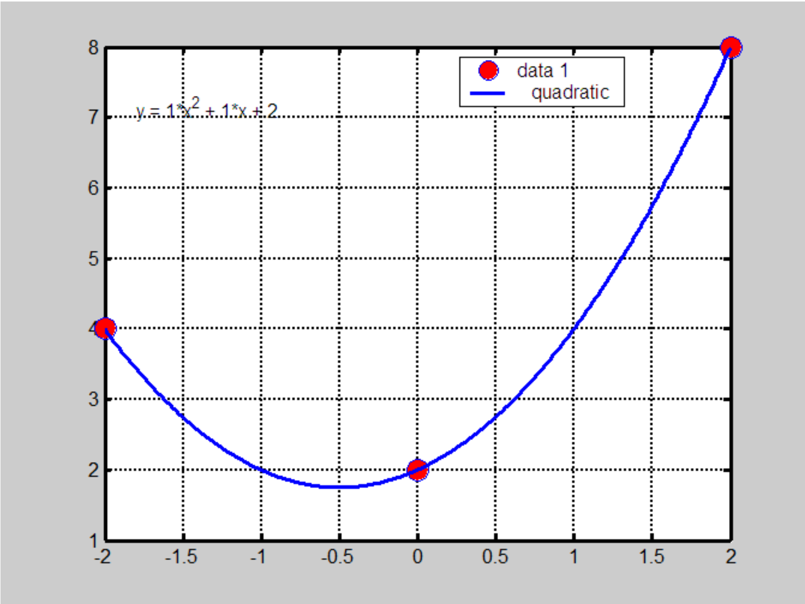
Find the second-degree interpolating polynomial to find the interpolated value for x=1.0.

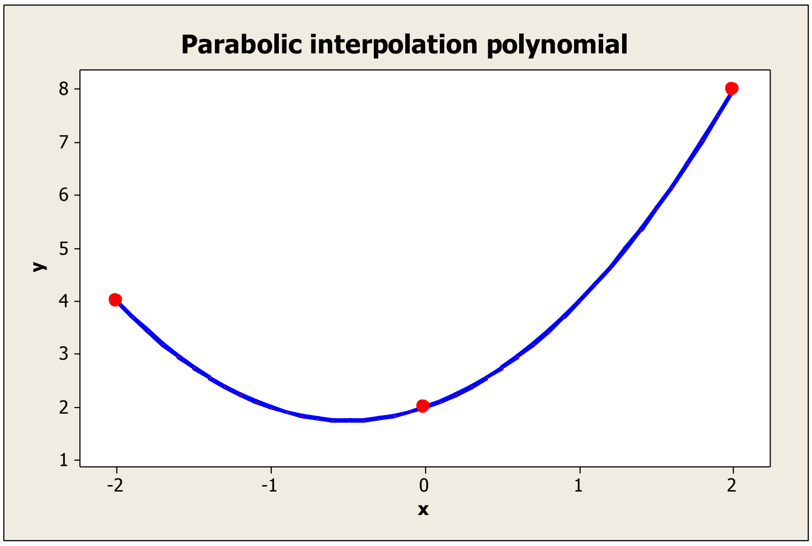


We can find a quadratic polynomial using the three given points by substituting into the general formula, which gives

$$
\begin{aligned}
p(x) &= \frac{(x-0)(x-2)}{(-2-0)(-2-2)}(4) + \frac{(x-(-2))(x-2)}{(0-(-2))(0-2)}(2) + \frac{(x-(-2))(x-0)}{(2-(-2))(2-0)}(8) \\
&= \frac{x(x-2)}{8}(4) + \frac{(x+2)(x-2)}{-4}(2) + \frac{x(x+2)}{8}(8) \\
&= x^2 + x + 2
\end{aligned}
$$

```
>> plot(x,y,'bo');grid on
```





P(1)=4

MATLAB

Calculation of coefficients of Lagrange Polynomial

```
function c = lagrangec(x ,y)
% calculate coefficients of Lagrange Polynomial
  n=length(x)

      for k=1 : n
              d(k)=1;
                 for i=1  : n
                             if i ~= k
                                  d(k)=d(k)*(x(k)-x(i));
                              end
                              c(k)=y(k)/d(k);
                          end
                      end
```

```
>> x=[-2 0 2]
x =    -2     0     2
>> y=[4 2 8]
y =     4     2     8

>> c=lagrangec(x,y)
n =     3
c =    0.5000   -0.5000    1.0000
```

$$
\begin{aligned}
p(x) &= \frac{(x-0)(x-2)}{(-2-0)(-2-2)}(4) + \frac{(x-(-2))(x-2)}{(0-(-2))(0-2)}(2) \\
&\quad + \frac{(x-(-2))(x-0)}{(2-(-2))(2-0)}(8) \\
&= \frac{x(x-2)}{8}(4) + \frac{(x+2)(x-2)}{-4}(2) + \frac{x(x+2)}{8}(8) \\
&= 0.5x(x-2) - 0.5(x+2)(x-2) + (1)x(x+2) \\
&= x^2 + x + 2
\end{aligned}
$$

Additional Data Points

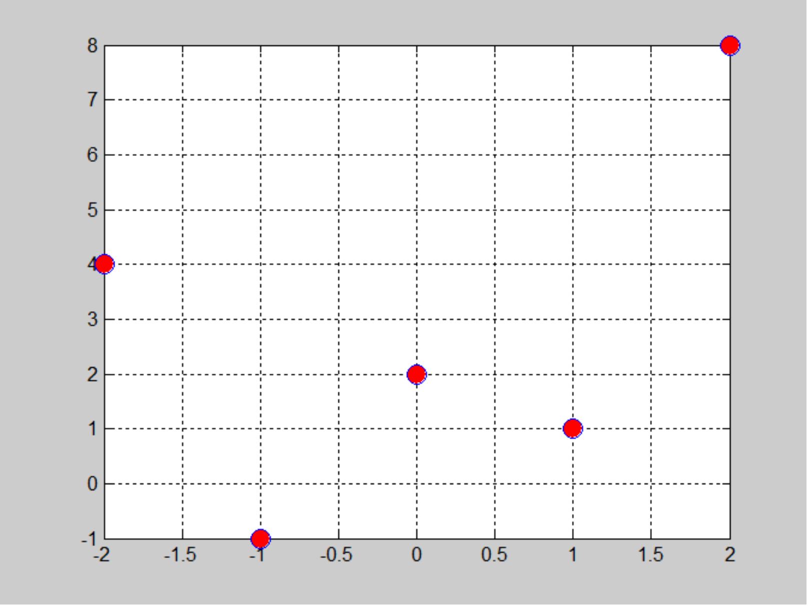
If we add two more data points to those that we previously interpolated, we must rework the problem.

Old table

x	y
-2	**4**
0	**2**
2	**8**

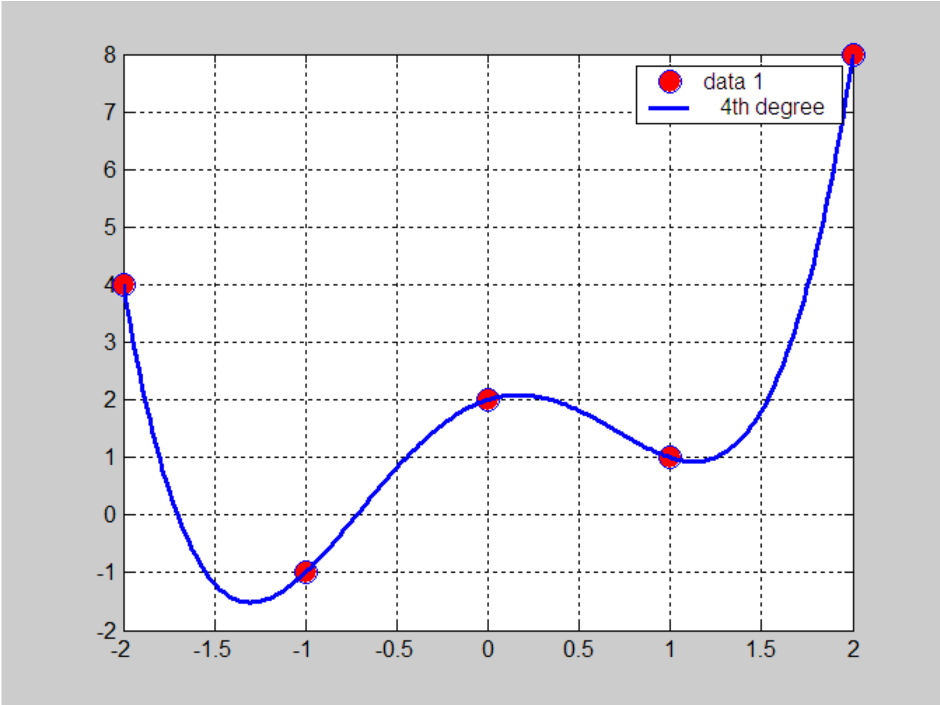
New Table

x	y
-2	**4**
0	**2**
-1	**-1**
1	**1**
2	**8**



$$p(x) = \frac{1}{6}(x)(x+1)(x-1)(x-2) + \frac{1}{2}(x+2)(x+1)(x-1)(x-2)$$
$$+\frac{1}{6}(x+2)(x)(x-1)(x-2) - \frac{1}{6}(x+2)(x)(x+1)(x-2)$$
$$+\frac{1}{3}(x+2)(x)(x+1)(x-1)$$

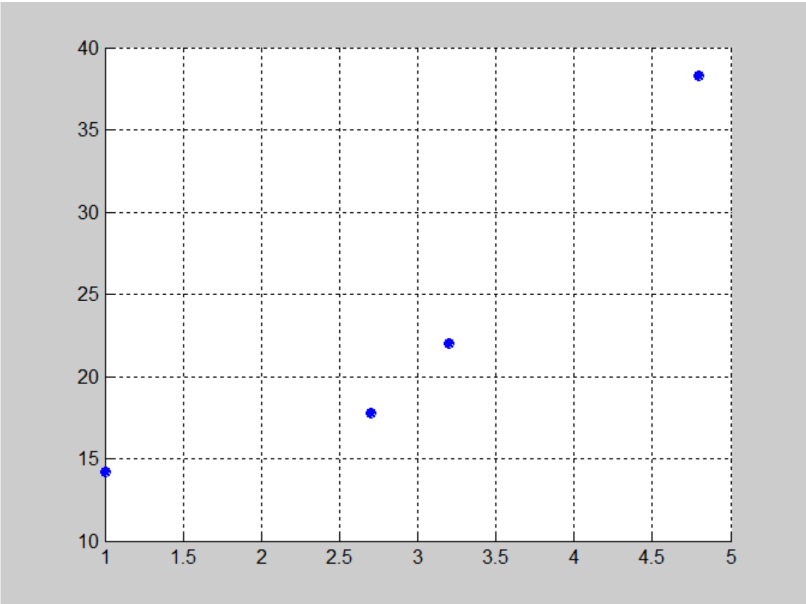
```
>> x=[-2 0 -1 1 2]
x =    -2     0    -1     1     2
>> y=[4 2 -1 1 8]
y =     4     2    -1     1     8
>> c=lagrangec(x, y)
n =     5
c =    0.1667    0.5000    0.1667   -0.1667    0.3333
```


data 1
4th degree
8
7
6
5
4
3
2
1
0
-1
-2
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2

Example: Suppose that we have the following data pairs – x values and f(x) values- where f(x) is <u>some unknown function.</u>

x	f(x)
3.2	22.0
2.7	17.8
1.0	14.2
4.8	38.3

Fit a cubic Lagrange interpolation polynomial and use it to find the interpolated value for x=3.0.



scatter(x,y,'filled');grid on

Solution:

To find the third degree interpolating polynomials for f(x) requires that we first determine the coefficient polynomials L_0, L_1, L_2, and L_3.

$$L_{3,0}(x) = \frac{(x-2.7)(x-1.0)(x-4.8)}{(3.2-2.7)(3.2-1.0)(3.2-4.8)},$$

$$L_{3,1}(x) = \frac{(x-3.2)(x-1.0)(x-4.8)}{(2.7-3.2)(2.7-1.0)(2.7-4.8)},$$

$$L_{3,2}(x) = \frac{(x-3.2)(x-2.7)(x-4.8)}{(1.0-3.2)(1.0-2.7)(1.0-4.8)},$$

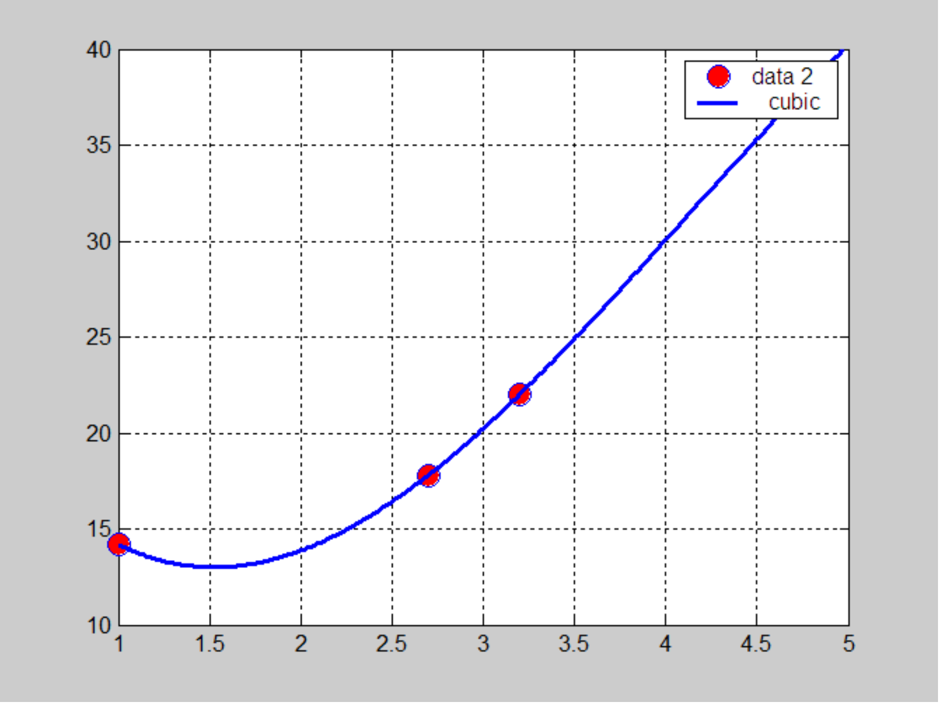
$$L_{3,3}(x) = \frac{(x-3.2)(x-2.7)(x-1.0)}{(4.8-3.2)(4.8-2.7)(4.8-1.0)}.$$

Then

$$\begin{aligned} P_3(x) = {} & \frac{(x-2.7)(x-1.0)(x-4.8)}{(3.2-2.7)(3.2-1.0)(3.2-4.8)} f_0 + \\ & \frac{(x-3.2)(x-1.0)(x-4.8)}{(2.7-3.2)(2.7-1.0)(2.7-4.8)} f_1 + \\ & \frac{(x-3.2)(x-2.7)(x-4.8)}{(1.0-3.2)(1.0-2.7)(1.0-4.8)} f_2 + \\ & \frac{(x-3.2)(x-2.7)(x-1.0)}{(4.8-3.2)(4.8-2.7)(4.8-1.0)} f_3 \\ = {} & \text{-}0.527x^3 + 6.4952x^2 - 16.1177x + 24.3499. \end{aligned}$$

Resulting Polynomial

$$P_3(x) = -0.527x^3 + 6.4952x^2 - 16.1177x + 24.3499.$$



>> x=[3.2 2.7 1.0 4.8]

>> y=[22 17.8 14.2 38.3]

>> c=lagrangec(x, y)

n = 4

c = -12.5000 9.9720 -0.9992 2.9997

To find the interpolated value for x=3.0

$$P_3(x) = -0.527(3)^3 + 6.4952(3)^2 - 16.1177(3) + 24.3499$$
$$= 20.21$$

or

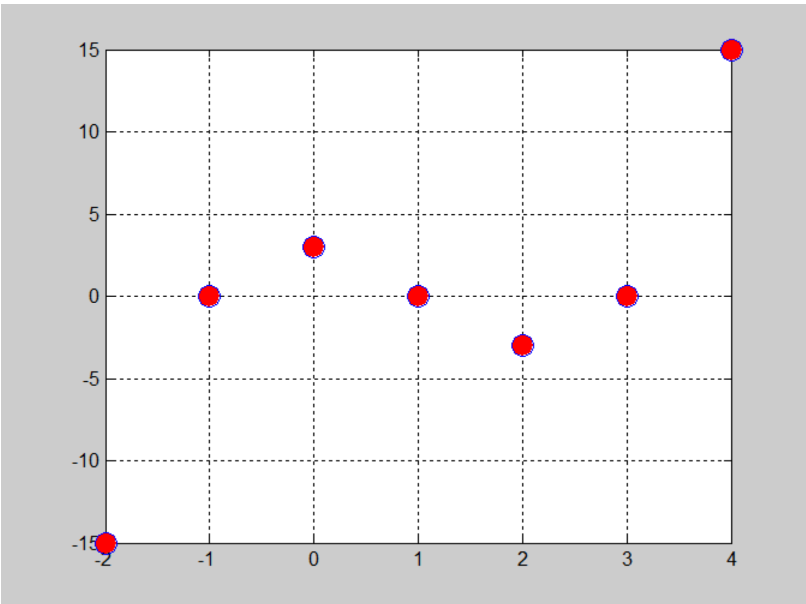
$$P_3(3.0) = \frac{(3.0-2.7)(3.0-1.0)(3.0-4.8)}{(3.2-2.7)(3.2-1.0)(3.2-4.8)}(22.0) +$$
$$\frac{(3.0-3.2)(3.0-1.0)(3.0-4.8)}{(2.7-3.2)(2.7-1.0)(2.7-4.8)}(17.8) +$$
$$\frac{(3.0-3.2)(3.0-2.7)(3.0-4.8)}{(1.0-3.2)(1.0-2.7)(1.0-4.8)}(14.2) +$$
$$\frac{(3.0-3.2)(3.0-2.7)(3.0-1.0)}{(4.8-3.2)(4.8-2.7)(4.8-1.0)}(38.3)$$
$$= 20.21$$

```
>> x=[3.2 2.7 1.0 4.8]
>> y=[22 17.8 14.2 38.3]
```

Example: (Higher Order Interpolation Polynomials)

Considering the following data:

x	f(x)
-2	-15
-1	0
0	3
1	0
2	-3
3	0
4	15



```
>> x=[-2 -1 0 1 2 3 4]
x =    -2    -1     0     1     2     3     4
>> y=[-15 0 3 0 -3 0 15]
```

$$L_{n,k}(x) = \frac{(x-x_0)\ldots(x-x_{k-1})(x-x_{k+1})\ldots(x-x_n)}{(x_k-x_0)\ldots(x_k-x_{k-1})\ldots(x_k-x_n)}$$
$$= \prod_{\substack{i=0\\ i\neq k}}^{n} \frac{(x-x_i)}{(x_k-x_i)}.$$

$$\begin{aligned}
P_6(x) = {} & \frac{(x+1)(x-0)(x-1)(x-2)(x-3)(x-4)}{(-2+1)(-2-0)(-2-1)(-2-3)(-2-4)} f_0 + \\
& \frac{(x+2)(x-0)(x-1)(x-2((x-3)(x-4)}{(-1+2)(-1-0)(-1-1)(-1-2)(-1-3)(-1-4)} f_1 + \\
& \frac{(x+2)(x+1)(x-1)(x-2)(x-3)(x-4)}{(0+2)(0+1)(0-1)(0-2)(0-3)(0-4)} f_2 + \\
& \frac{(x+2)(x+1)(x-0)(x-2)(x-3)(x-4)}{(1+2)(1+1)(1-0)(1-2)(1-3)(1-4)} f_3 + \\
& \frac{(x+2)(x+1)(x-0)(x-1)(x-3)(x-4)}{(2+2)(2+1)(2-0)(2-1)(2-3)(2-4)} f_4 + \\
& \frac{(x+2)(x+1)(x-0)(x-1)(x-2)(x-4)}{(3+2)(3+1)(3-0)(3-1)(3-2)(3-4)} f_5 + \\
& \frac{(x+2)(x+1)(x-0)(x-1)(x-2)(x-3)}{(4+2)(4+1)(4-0)(4-1)(4-2)(4-3)} f_6
\end{aligned}$$

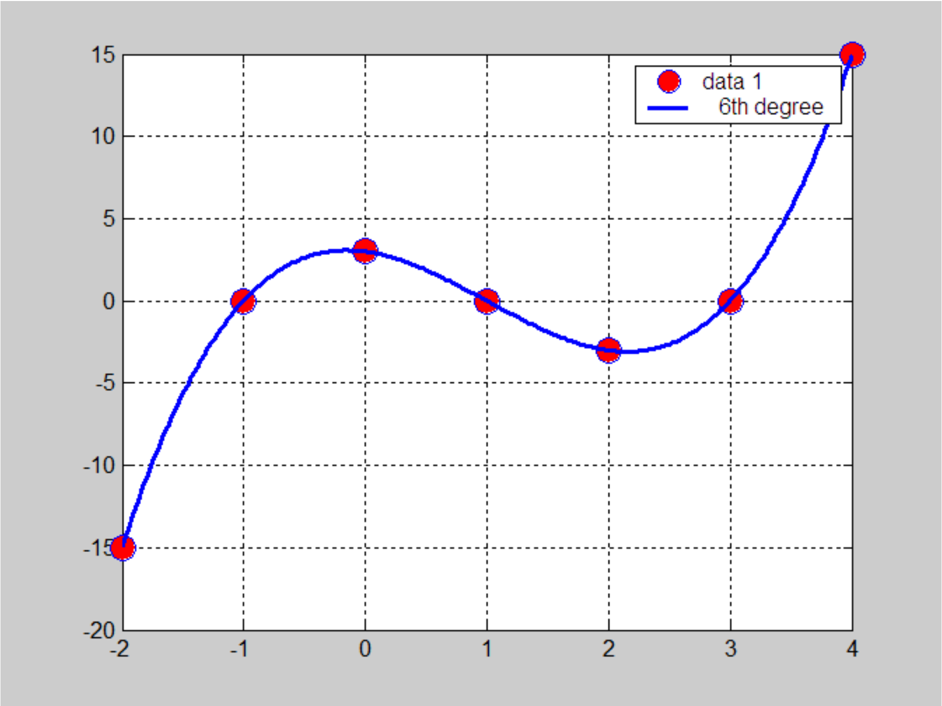
>> y=[-15 0 3 0 -3 0 15]

$$
\begin{aligned}
P_6(x) = {} & \frac{(x+1)(x-0)(x-1)(x-2)(x-3)(x-4)}{(-2+1)(-2-0)(-2-1)(-2-3)(-2-4)}(-15)+ \\
& \frac{(x+2)(x-0)(x-1)(x-2((x-3)(x-4)}{(-1+2)(-1-0)(-1-1)(-1-2)(-1-3)(-1-4)}(0)+ \\
& \frac{(x+2)(x+1)(x-1)(x-2)(x-3)(x-4)}{(0+2)(0+1)(0-1)(0-2)(0-3)(0-4)}(3)+ \\
& \frac{(x+2)(x+1)(x-0)(x-2)(x-3)(x-4)}{(1+2)(1+1)(1-0)(1-2)(1-3)(1-4)}(0)+ \\
& \frac{(x+2)(x+1)(x-0)(x-1)(x-3)(x-4)}{(2+2)(2+1)(2-0)(2-1)(2-3)(2-4)}(-3)+ \\
& \frac{(x+2)(x+1)(x-0)(x-1)(x-2)(x-4)}{(3+2)(3+1)(3-0)(3-1)(3-2)(3-4)}(0)+ \\
& \frac{(x+2)(x+1)(x-0)(x-1)(x-2)(x-3)}{(4+2)(4+1)(4-0)(4-1)(4-2)(4-3)}(15)
\end{aligned}
$$

$$
\begin{aligned}
p(x) = {} & -0.0208(x+1)(x)(x-1)(x-2)(x-3)(x-4) \\
& +0.0625(x+2)(x+1)(x-1)(x-2)(x-3)(x-4) \\
& -0.0625(x+2)(x+1)(x)(x-1)(x-3)(x-4) \\
& +0.0208(x+2)(x+1)(x)(x-1)(x-2)(x-3).
\end{aligned}
$$

```
>> x=[-2 -1 0 1 2 3 4]
x =    -2    -1     0     1     2     3     4
>> y=[-15 0 3 0 -3 0 15]

y =   -15     0     3     0    -3     0    15


>> c=lagrangec(x,y)

n =     7

c =   -0.0208         0    0.0625         0   -0.0625         0    0.0208
```

$$
\begin{aligned}
p(x) = &-0.0208(x+1)(x)(x-1)(x-2)(x-3)(x-4) \\
&+0.0625(x+2)(x+1)(x-1)(x-2)(x-3)(x-4) \\
&-0.0625(x+2)(x+1)(x)(x-1)(x-3)(x-4) \\
&+0.0208(x+2)(x+1)(x)(x-1)(x-2)(x-3).
\end{aligned}
$$



Example: Table lists values of a function (<u>The Bessel function of the first kind of order zero</u>) at various points. The approximation to f(1.5) obtained by various Lagrange polynomials will be compared.

x	f(x)
1.0	0.7651977
1.3	0.6200860
1.6	0.4554022
1.9	0.2818126
2.2	0.1103623

Since 1.5 is between 1.3 and 1.6, the linear polynomial will use x_0=1.3 and x_1=1.6.

The value of the interpolating polynomial at 1.5 is given by

$$P_1(1.5) = \frac{(1.5-1.6)}{(1.3-1.6)}(0.6200860) + \frac{(1.5-1.3)}{(1.6-1.3)}(0.4554022)$$
$$= 0.5102968$$

The polynomials of degree two could be reasonably be used by letting $x_0=1.3$, $x_1=1.6$, and $x_2=1.9$, which gives

$$
\begin{aligned}
P_2(1.5) &= \frac{(1.5-1.6)(1.5-1.9)}{(1.3-1.6)(1.3-1.9)}(0.6200860) \\
&\quad + \frac{(1.5-1.3)(1.5-1.9)}{(1.6-1.3)(1.6-1.9)}(0.4554022) \\
&\quad + \frac{(1.5-1.3)(1.5-1.6)}{(1.9-1.3)(1.9-1.6)}(0.2818186) \\
&= 0.5112857
\end{aligned}
$$

And the other by letting $x_0=1.0$, $x_1=1.3$, and $x_2=1.6$, in which case

$$\hat{P}_2(1.5) = 0.5124715$$

In the third-degree case there are also two choices for the polynomial.

One with $x_0=1.3$, $x_1=1.6$, $x_2=1.9$, and $x_3=2.2$, which gives

$$P_3(1.5) = 0.5118302.$$

The other is obtained by letting $x_0=1.0$, $x_1=1.3$, $x_2=1.6$, and $x_3=1.9$, giving

$$\hat{P}_3(1.5) = 0.5118127.$$

The forth-degree Lagrange polynomial uses all the entries in the table.

With $x_0=1.0$, $x_1=1.3$, $x_2=1.6$, $x_3=1.9$ and $x_4=2.2$, it can be shown that,

$$P_4(1.5) = 0.5118200.$$

Since $P_3(1.5) = 0.5118302$, and $\hat{P}_3(1.5) = 0.5118127$ and $P_4(1.5) = 0.5118200$ all agree to within 2×10^{-5} units, we expect $P_4(1.5) = 0.5118200$ to be the most accurate approximation and to be correct to within 2×10^{-5} units.

The actual value of f(1.5) is known to be 0.5118277, so the true accuracies of the approximations are as follows:

$$|P_1(1.5) - f(1.5)| \approx 1.53*10^{-3}$$

$$|P_2(1.5) - f(1.5)| \approx 5.42*10^{-4}$$

$$|\hat{P}_2(1.5) - f(1.5)| \approx 6.44*10^{-4}$$

$$|P_3(1.5) - f(1.5)| \approx 2.5*10^{-6}$$

$$|\hat{P}_3(1.5) - f(1.5)| \approx 1.5*10^{-5}$$

$$|P_4(1.5) - f(1.5)| \approx 7.7*10^{-6}$$

P_3 is the most accurate approximation. However, with no knowledge of the actual value of f(1.5), P_4 would be accepted as the best approximation. The error or remainder term cannot be applied here, since no knowledge of the forth derivative of f is available.

Error Terms

The error form for the Lagrange polynomial is quite similar to that for the Taylor polynomial. The Taylor polynomial of degree n about x_0 has an error term of the form

$$\frac{f^{(n+1)}(\xi(x))}{(n+1)!}(x-x_0)^{n+1}.$$

The Lagrange polynomial of degree n uses information at the distinct numbers $x_0, x_1, \ldots x_n$ and, in place of $(x-x_0)^n$, its error formula uses a product of the n+1 terms

$(x-x_0)(x-x_1)\ldots(x-x_n)$**:**

$$\frac{f^{(n+1)}(\xi(x))}{(n+1)!}(x-x_0)(x-x_1)\ldots(x-x_n).$$

With ξ on the smallest interval that contains

$\{x, x_0, x_1, \ldots x_n\}$**.**

Using MATLAB

MATLAB gets interpolating polynomials readily.

Example: Suppose that we have the following data pairs – x values and f(x) values- where f(x) is <u>some unknown function.</u>

x	f(x)
3.2	22.0
2.7	17.8
1.0	14.2
4.8	38.3

<u>Fit a cubic Lagrange interpolation polynomial using MATLAB and use it to find the interpolated value for x=3.0.</u>

>> x=[3.2 2.7 1.0 4.8]

x =

3.2000 2.7000 1.0000 4.8000

>> y=[22.0 17.8 14.2 38.3]

y =

22.0000 17.8000 14.2000 38.3000

>> P=polyfit(x, y, 3)

P = -0.5275 6.4952 -16.1177 24.3499

$$\mathrm{P(x)} = -0.527x^3 + 6.4952x^2 - 16.1177x + 24.3499.$$

>> xval=polyval(P,3.0)

xval = 20.2120

An Algorithm for Interpolation from a Lagrange Polynomial

Given a set of n+1 points $[(x_i,f_i), i=0,\ldots n]$ and a value for x at which the polynomial is to be evaluated:

```
            Set  Sum=0

        For i=1 to n Step 1 Do
            Set P=1.
               For j=1 To n Step 1 Do
          If (j≠i) Then
             Set P=P*(x-xj)/(xi-xj)
          End If
              End Do j
        Set Sum= Sum +P*fi
        End Do i
Sum is the interpolated value at x
```

MATLAB M-File (Lagrange Approximation)

```
function [C,L]=lagran(X,Y)
%Input   - X is a vector that contains a list of abscissas
%        - Y is a vector that contains a list of ordinates
%Output - C is a matrix that contains the coefficents of
%          the Lagrange interpolatory polynomial
%        - L is a matrix that contains the Lagrange
%          coefficient polynomials
w=length(X);
n=w-1;
L=zeros(w,w);
%Form the Lagrange coefficient polynomials
for k=1:n+1
   V=1;
   for j=1:n+1
      if k~=j
         V=conv(V,poly(X(j)))/(X(k)-X(j));
      end
   end
   L(k,:)=V;
end
%Determine the coefficients of the Lagrange interpolator
%polynomial
C=Y*L;
```

Example: Suppose that we have the following data pairs – x values and f(x) values- where f(x) is some unknown function.

x	**f(x)**
3.2	**22.0**
2.7	**17.8**
1.0	**14.2**
4.8	**38.3**

Fit a cubic Lagrange interpolation polynomial using MATLAB

```
>> x=[3.2 2.7 1.0 4.8]
x =    3.2000    2.7000    1.0000    4.8000
>> y=[22.0 17.8 14.2 38.3]
y =   22.0000   17.8000   14.2000   38.3000


>>[C,L]=lagran(x,y)
C =   -0.5275    6.4952  -16.1177   24.3499
```

$$\mathrm{P(x)} = -0.527x^3 + 6.4952x^2 - 16.1177x + 24.3499.$$

```
L =
   -0.5682    4.8295  -11.6250    7.3636
    0.5602   -5.0420   13.0868   -8.6050
   -0.0704    0.7529   -2.6006    2.9181
    0.0783   -0.5404    1.1388   -0.6767
```

Polyfit Command

in MATLAB

If we use Polyfit command

P=polyfit(x, y, 3)

P =   -0.5275   6.4952 -16.1177   24.3499

We obtain same result.

(So Polyfit command use Lagrange Polynomial Approximation)

Neville's Method

The trouble with the standard Lagrange Polynomial technique is that we do not know which degree of polynomial to use.

- **If the degree is too low, the interpolating polynomial does not give good estimates of f(x).**
- **If the degree is too high, undesirable oscillations in polynomial values can occur.**

Neville's method can overcome this difficulty. The usual practice is to compute the results given from various polynomials until appropriate agreement is obtained.

The Lagrange formula for linear interpolation to P(x) from two data pairs, (x_0,y_0) and (x_1,y_1), is

$$P_1(x) = \frac{(x - x_1)}{(x_0 - x_1)} y_0 + \frac{(x - x_0)}{(x_1 - x_0)} y_1$$

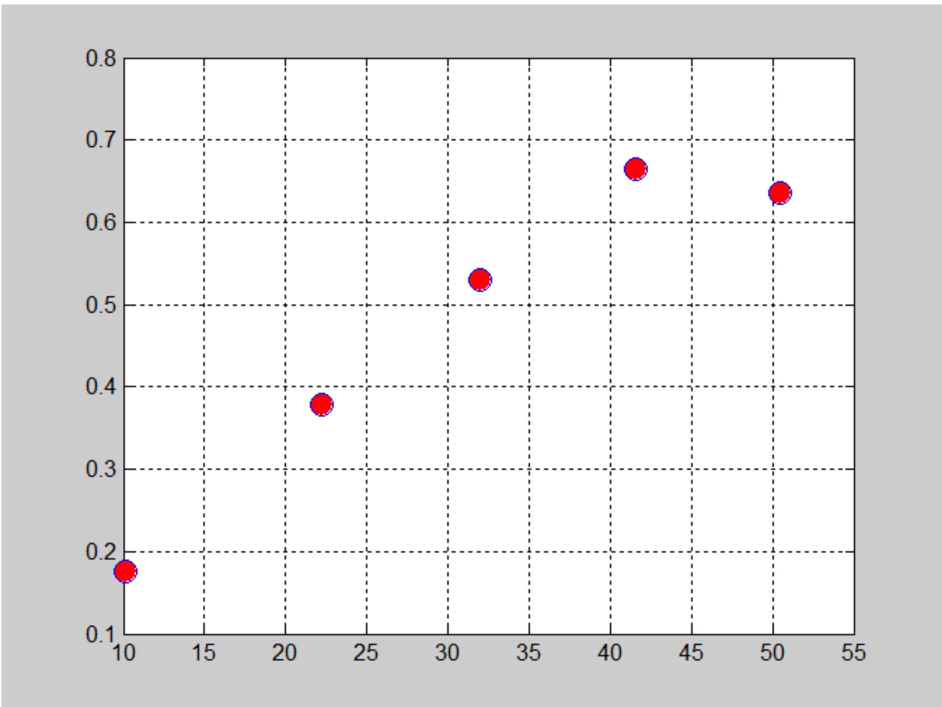
this can be written more compactly as

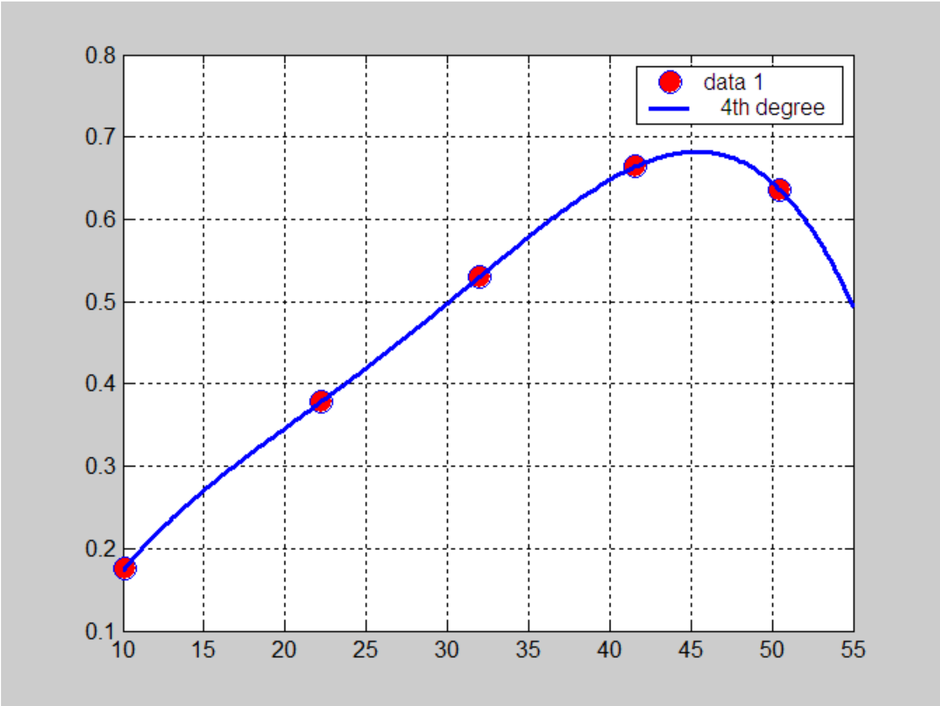
$$P_1(x) = \frac{(x - x_1) y_0 + (x_0 - x) y_1}{(x_0 - x_1)}$$

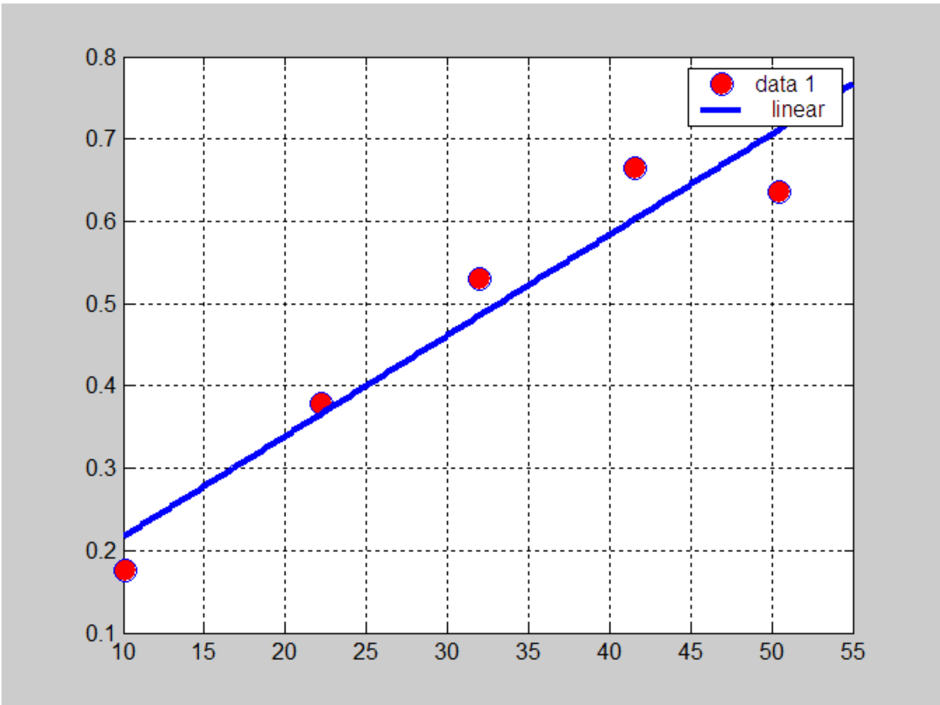
Example: Suppose we are given these data:

x	f(x)
10.1	0.17537
22.2	0.37784
32.0	0.52992
41.6	0.66393
50.5	0.63608

We want to interpolate for x=27.5.

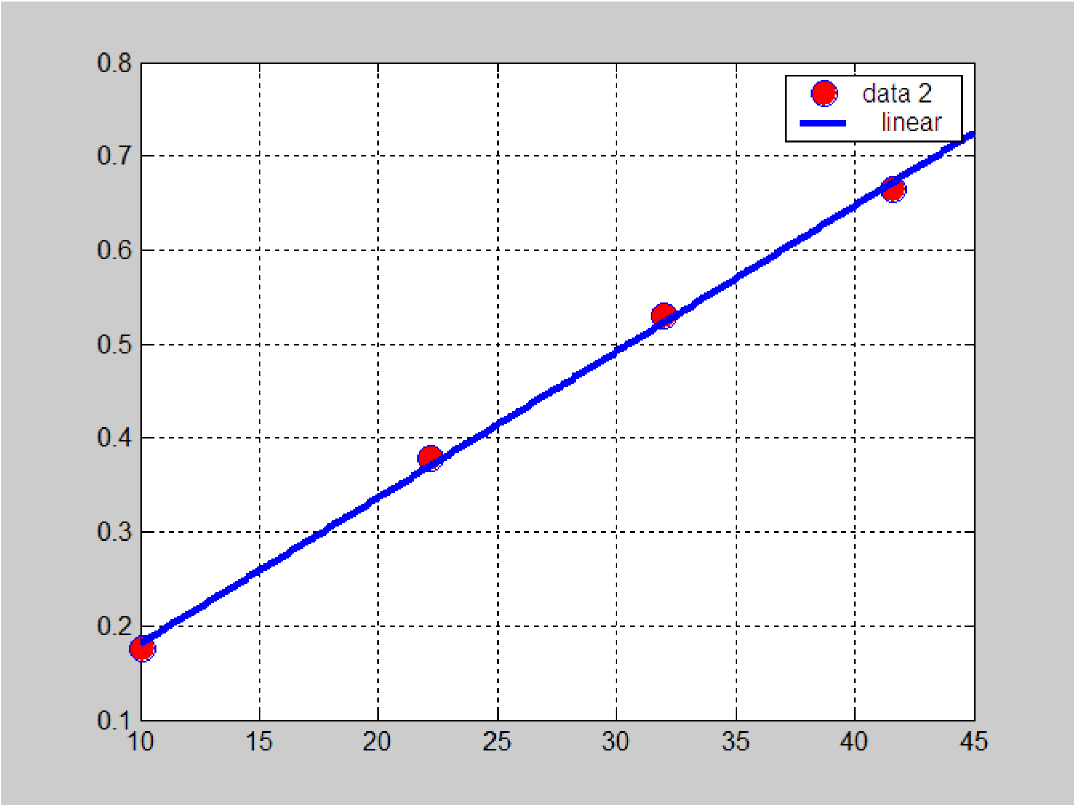






The Best Linear Model ?

If we omit the last point (nearly linear relation)



We first rearrange the data pairs in order of closeness to $x=27.5$:

i	x	f(x)	$\lvert x-x_i \rvert$
0	32.0	0.52992	4.5
1	22.2	0.37784	5.3
2	41.6	0.66393	14.1
3	10.1	0.17537	17.4
4	50.5	0.63608	23.0

Neville's method begins by renaming the f_i as P_{i0}. We build a Table by first interpolating linearly between **pairs of values for i=0,1, i=1,2, i=2,3, and so on. The values are written in a column to the right of the first P of each pair. The next column of the Table is created by linearly interpolating from the previous column for i=0,2, i=1,3, i=2,4 and so on. The next column after this uses values for i=0,3, i=1,4, and the last column uses values for i=0,4.**

Here is the Neville table for the preceding data:

i	x	f(x)	P_{i1}	P_{i2}	P_{i3}	P_{i4}
0	**32.0**	**0.52992**	**0.4609**	**0.46200**	**0.46174**	**0.45754**
1	**22.2**	**0.37784**	**0.45600**	**0.46071**	**0.47901**	
2	**41.6**	**0.66393**	**0.44524**	**0.55843**		
3	**10.1**	**0.17537**	**0.37379**			
4	**50.5**	**0.63608**				

The general formula for computing entries into the table is

$$P_{i,j} = \frac{(x - x_i)P_{i+1,j-1} + (x_{i+j} - x)P_{i,j-1}}{x_{i+j} - x_i}$$

Thus, the values of P_{01} and P_{11} are computed by

$$P_{01} = \frac{(27.5-32.0)*0.3784+(22.5-27.5)*0.52992}{22.2-32.0} = 0.46009,$$

$$P_{11} = \frac{(27.5-22.2)*0.66393+(41.6-27.5)*0.37784}{41.6-22.2} = 0.45600,$$

Once we have the column of P_{i1}'s, we compute the next column. For example,

$$P_{22} = \frac{(27.5-41.6)*0.37379+(50.5-27.5)*0.44524}{50.5-41.6} = 0.55843,$$

The remaining columns are computed similarly.
The top line of the Table represents Lagrange interpolates at x=27.5 using polynomials of degree equal to the second subscript of the P's. Each of these polynomials uses the required number of data pairs, taking them as a set starting from the top of the table.

> x=[10.1 22.2 32.0 41.6 50.5]

x = 10.1000 22.2000 32.0000 41.6000 50.5000

>> y=[0.175357 0.37784 0.52992 0.66393 0.63608]

y = 0.1754 0.3778 0.5299 0.6639 0.6361

>> P=polyfit(x,y,4)

P = -0.0000 0.0001 -0.0028 0.0603 -0.2148

xval=polyval(P,27.5)

xval = 0.45754 (Result of the 4^{th} degree polynomial but we obtain same result using Neville's Method)
The preceding data are for sines of angles in degrees and the correct value for x=27.5 is 0.46175. Observe that the top line values get better and better until the last, when it diverges.

An Algorithm for Neville's Iterated Interpolation

To evaluate the interpolating polynomial P on the n+1 distinct numbers $x_0, x_1, \ldots, x_n$ at the number x for the function f:
INPUT $x_0, x_1, \ldots, x_n$ and function values f
OUTPUT The Table Q with $P(x)=Q_{n,n}$

Step 1
For i=2 to n
For j=1.. To i

$$\text{set}\, Q_{i,j} = \frac{(x - x_{i-j})Q_{i,j-1} - (x - x_i)Q_{i-1,j-1}}{(x_i - x_{i-j})}$$

Step 2 OUTPUT (Q);
STOP