Topic : Newton Divided Difference Method - Interpolation
Simulation : Graphical Simulation of the Method
Language : Matlab r12
Authors : Nathan Collier, Autar Kaw
Date : 12 October
Abstract : This simulation illustrates the Newton Divided Difference method of interpolation. Given n data points of y versus x, you are then required to find the value of y at a particular value of x using first, second, and third order interpolation. So one has to first pick the needed data points, and then use those to interpolate data.

```
clear all
% INPUTS: Enter the following

% Array of x-data
  x=[10 0 20 15 30 22.5];

% Array of y-data
  y=[227.04 0 517.35 362.78 901.67 602.97];

% Value of x at which y is desired
  xdesired = 16;

% SOLUTION

% This calculates window size to be used in figures
set(0,'Units','pixels')
scnsize = get(0,'ScreenSize');
wid = round(scnsize(3));
hei = round(0.95*scnsize(4));
wind = [1, 1, wid, hei];

% The following considers the x and y data and selects the two closest points to xdesired
% that also bracket it.
n = numel(x);
comp = abs(x-xdesired);
c=min(comp);
for i=1:n
    if comp(i)==c;
    ci=i;
    end
end
if x(ci) < xdesired
    q=1;
    for i=1:n
        if x(i) > xdesired
```

```
            ne(q)=x(i);
            q=q+1;
        end
    end
    b=min(ne);
    for i=1:n
        if x(i)==b
            bi=i;
        end
    end
end
if x(ci) > xdesired
    q=1;
    for i=1:n
        if x(i) < xdesired
            ne(q)=x(i);
            q=q+1;
        end
    end
    b=max(ne);
    for i=1:n
        if x(i)==b
            bi=i;
        end
    end
end
% If more than two values are needed, the following selects the subsequent values and
puts
% them into a matrix, preserving the orginal data order.
for i = 1:n
    A(i,2)=i;
    A(i,1)=comp(i);
end
A=sortrows(A,1);
for i=1:n
    A(i,3)=i;
end
A=sortrows(A,2);
d=A(1:n,3);
if d(bi)~=2
    temp=d(bi);
    d(bi)=1;
    for i=1:n
        if i ~= bi & i ~= ci & d(i) <= temp
            d(i)=d(i)+1;
        end
```

```
        d(ci)=1;
    end
end

%%%%%%%%% LINEAR INTERPOLATION %%%%%%%%%

% Pick two data points
datapoints=2;
p=1;
for i=1:n
    if d(i) <= datapoints
        xdata(p)=x(i);
        ydata(p)=y(i);
        p=p+1;
    end
end

% Calculating coefficients of Newton's Divided difference polynomial
b0=sym('b0');
b1=sym('b1');
z=sym('z');
b0=ydata(1);
b1=(ydata(2)-ydata(1))/(xdata(2)-xdata(1));
fl=b0 +b1*(z-xdata(1));
fxdesired=subs(fl,z,xdesired);
fprev=fxdesired;

% Writing to figure windows
figure('Position',wind)
s=0.04;
title('Linear interpolation','Fontweight','bold','FontSize',14)
text(0,1,'Selected Data','Fontweight','bold')
axis off
p=1;
text(0,p-s ,['      ',num2str(xdata(1))])
text(0,p-1.5*s,['x = '])
text(0,p-2*s,['      ',num2str(xdata(2))])
text(0.1,p-s ,['      ',num2str(ydata(1))])
text(0.1,p-1.5*s,['y = '])
text(0.1,p-2*s,['      ',num2str(ydata(2))])

p=p-2*s-2*s;
text(0,p,'Calculating coefficients of Newton^,s Divided difference
polynomial','Fontweight','bold')
text(0,p-s,'b0 = y(1)')
text(0,p-2*s,['b0 = ',num2str(b0)])
```

```
text(0,p-3*s,'b1 = [y(2)-y(1)] / [x(2)-x(1)]')
text(0,p-4*s,['b1 = ',num2str(b1)])

p=p-6*s;
text(0,p,'Newton^,s divided difference formula for linear
interpolation','Fontweight','bold')
text(0,p-s,'f(x) = b0 + b1 * (x-x(1))')

p=p-3*s;
text(0,p,'Calculating value at desired point','Fontweight','bold')
text(0,p-s,['f(xdesired) = f(',num2str(xdesired),') = ',num2str(fxdesired)])

figure('Position',wind)
axis on
subplot(2,1,1), ezplot(fl,[min(xdata),max(xdata)])
title('Linear interpolation','Fontweight','bold')
hold on
plot(xdata,ydata,'ro','MarkerSize',10','MarkerFaceColor',[1,0,0])
plot(xdesired,fxdesired,'kx','Linewidth',2,'MarkerSize',12')

axis on
subplot(2,1,2), ezplot(fl,[min(xdata),max(xdata)])
title('  ')
hold on
plot(x,y,'ro','MarkerSize',10','MarkerFaceColor',[1,0,0])
plot(xdesired,fxdesired,'kx','Linewidth',2,'MarkerSize',12')
xlim([min(x) max(x)])
ylim([min(y) max(y)])


%%%%%%%%% QUADRATIC INTERPOLATION %%%%%%%%%

% Pick three data points
datapoints=3;
p=1;
for i=1:n
    if d(i) <= datapoints
        xdata(p)=x(i);
        ydata(p)=y(i);
        p=p+1;
    end
end

% Calculating coefficients of Newton's Divided difference polynomial
b0=sym('b0');
b1=sym('b1');
```

```
b2=sym('b2');
z=sym('z');
b0=ydata(1);
b1=(ydata(2)-ydata(1))/(xdata(2)-xdata(1));
b2=((ydata(3)-ydata(2))/(xdata(3)-xdata(2))-(ydata(2)-ydata(1))/(xdata(2)-
xdata(1)))/(xdata(3)-xdata(1));
fq=b0 +b1*(z-xdata(1)) + b2*(z-xdata(1))*(z-xdata(2));
fxdesired=subs(fq,z,xdesired);

fnew=fxdesired;
ea=abs((fnew-fprev)/fnew*100);
if ea >= 5
   sd=0;
else
   sd=floor(2-log10(abs(ea)/0.5));
end

% Writing to figure windows
figure('Position',wind)
s=0.04;
title('Quadratic interpolation','Fontweight','bold','FontSize',14)
text(0,1,'Selected Data','Fontweight','bold')
axis off
p=1;
text(0,p-s ,['      ',num2str(xdata(1))])
text(0,p-1.5*s,['x = '])
text(0,p-2*s,['      ',num2str(xdata(2))])
text(0,p-3*s,['      ',num2str(xdata(3))])
text(0.1,p-s ,['      ',num2str(ydata(1))])
text(0.1,p-1.5*s,['y = '])
text(0.1,p-2*s,['      ',num2str(ydata(2))])
text(0.1,p-3*s,['      ',num2str(ydata(3))])

p=p-3*s-2*s;
text(0,p,'Calculating coefficients of Newton^,s Divided difference
polynomial','Fontweight','bold')
text(0,p-s,'b0 = y(1)')
text(0,p-2*s,['b0 = ',num2str(b0)])
text(0,p-3*s,'b1 = [y(2)-y(1)] / [x(2)-x(1)]')
text(0,p-4*s,['b1 = ',num2str(b1)])
text(0,p-5*s,'b2 = ([y(3)-y(2)]/[x(3)-x(2)]-[y(2)-y(1)]/[x(2)-x(1)])/[x(3)-x(1)]')
text(0,p-6*s,['b2 = ',num2str(b2)])

p=p-8*s;
text(0,p,'Newton^,s divided difference formula for linear
interpolation','Fontweight','bold')
```

```
text(0,p-s,'f(x) = b0 + b1 * (x-x(1)) + b2 * (x-x(1)) * (x-x(2))')

p=p-3*s;
text(0,p,'Calculating value at desired point','Fontweight','bold')
text(0,p-s,['f(xdesired) = f(',num2str(xdesired),') = ',num2str(fxdesired)])

p=p-3*s;
text(0,p,'Absolute relative approximate error and significant digits','Fontweight','bold')
text(0,p-s,['ea = abs(( ',num2str(fnew),' - ',num2str(fprev),' )/ ',num2str(fnew),')*100 =
',num2str(ea), ' %'])
text(0,p-2*s,['sigdig = ',num2str(sd)])

figure('Position',wind)
axis on
subplot(2,1,1), ezplot(fq,[min(xdata),max(xdata)])
title('Quadratic interpolation','Fontweight','bold')
hold on
plot(xdata,ydata,'ro','MarkerSize',10','MarkerFaceColor',[1,0,0])
plot(xdesired,fxdesired,'kx','Linewidth',2,'MarkerSize',12')

axis on
subplot(2,1,2), ezplot(fq,[min(xdata),max(xdata)])
title('  ')
hold on
plot(x,y,'ro','MarkerSize',10','MarkerFaceColor',[1,0,0])
plot(xdesired,fxdesired,'kx','Linewidth',2,'MarkerSize',12')
xlim([min(x) max(x)])
ylim([min(y) max(y)])
fprev=fnew;


%%%%%%%% CUBIC INTERPOLATION %%%%%%%%

% Pick four data points
datapoints=4;
p=1;
for i=1:n
    if d(i) <= datapoints
        xdata(p)=x(i);
        ydata(p)=y(i);
        p=p+1;
    end
end

% Calculating coefficients of Newton's Divided difference polynomial
b0=sym('b0');
```

```
b1=sym('b1');
b2=sym('b2');
b3=sym('b3');
z=sym('z');
b0=ydata(1);
b1=(ydata(2)-ydata(1))/(xdata(2)-xdata(1));
b2=((ydata(3)-ydata(2))/(xdata(3)-xdata(2))-(ydata(2)-ydata(1))/(xdata(2)-
xdata(1)))/(xdata(3)-xdata(1));
b3=(((ydata(4)-ydata(3))/(xdata(4)-xdata(3))-(ydata(3)-ydata(2))/(xdata(3)-
xdata(2)))/(xdata(4)-xdata(2))-b2)/(xdata(4)-xdata(1));
fc=b0 +b1*(z-xdata(1)) + b2*(z-xdata(1))*(z-xdata(2)) + b3*(z-xdata(1))*(z-
xdata(2))*(z-xdata(3));
fxdesired=subs(fc,z,xdesired);

fnew=fxdesired;
ea=abs((fnew-fprev)/fnew*100);
if ea >= 5
   sd=0;
else
   sd=floor(2-log10(abs(ea)/0.5));
end

figure('Position',wind)
title('Cubic interpolation','Fontweight','bold','FontSize',14)
text(0,1,'Selected Data','Fontweight','bold')
axis off
p=1;
text(0,p-s ,['      ',num2str(xdata(1))])
text(0,p-1.5*s,['x = '])
text(0,p-2*s,['      ',num2str(xdata(2))])
text(0,p-3*s,['      ',num2str(xdata(3))])
text(0,p-4*s,['      ',num2str(xdata(4))])
text(0.1,p-s ,['      ',num2str(ydata(1))])
text(0.1,p-1.5*s,['y = '])
text(0.1,p-2*s,['      ',num2str(ydata(2))])
text(0.1,p-3*s,['      ',num2str(ydata(3))])
text(0.1,p-4*s,['      ',num2str(ydata(4))])

p=p-4*s-2*s;
text(0,p,'Calculating coefficients of Newton^,s Divided difference
polynomial','Fontweight','bold')
text(0,p-s,'b0 = y(1)')
text(0,p-2*s,['b0 = ',num2str(b0)])
text(0,p-3*s,'b1 = [y(2)-y(1)] / [x(2)-x(1)]')
text(0,p-4*s,['b1 = ',num2str(b1)])
text(0,p-5*s,'b2 = ([y(3)-y(2)]/[x(3)-x(2)]-[y(2)-y(1)]/[x(2)-x(1)])/[x(3)-x(1)]')
```

```
text(0,p-6*s,['b2 = ',num2str(b2)])
text(0,p-7*s,'b3 = (((y(4)-y(3))/(x(4)-x(3))-(y(3)-y(2))/(x(3)-x(2)))/(x(4)-x(2))-b2)/(x(4)-
x(1))')
text(0,p-8*s,['b3 = ',num2str(b3)])

p=p-10*s;
text(0,p,'Newton^,s divided difference formula for linear
interpolation','Fontweight','bold')
text(0,p-s,'f(x) = b0 + b1 * (x-x(1)) + b2 * (x-x(1)) * (x-x(2)) + b3 * (x-x(1)) * (x-x(2)) *
(x-x(3))')

p=p-3*s;
text(0,p,'Calculating value at desired point','Fontweight','bold')
text(0,p-s,['f(xdesired) = f(',num2str(xdesired),') = ',num2str(fxdesired)])

p=p-3*s;
text(0,p,'Absolute relative approximate error and significant digits','Fontweight','bold')
text(0,p-s,['ea = abs(( ',num2str(fnew),' - ',num2str(fprev),' )/ ',num2str(fnew),')*100 =
',num2str(ea), ' %'])
text(0,p-2*s,['sigdig = ',num2str(sd)])

figure('Position',wind)
axis on
subplot(2,1,1), ezplot(fc,[min(xdata),max(xdata)])
title('Cubic interpolation','Fontweight','bold')
hold on
plot(xdata,ydata,'ro','MarkerSize',10','MarkerFaceColor',[1,0,0])
plot(xdesired,fxdesired,'kx','Linewidth',2,'MarkerSize',12')

axis on
subplot(2,1,2), ezplot(fc,[min(xdata),max(xdata)])
title('  ')
hold on
plot(x,y,'ro','MarkerSize',10','MarkerFaceColor',[1,0,0])
plot(xdesired,fxdesired,'kx','Linewidth',2,'MarkerSize',12')
xlim([min(x) max(x)])
ylim([min(y) max(y)])
```

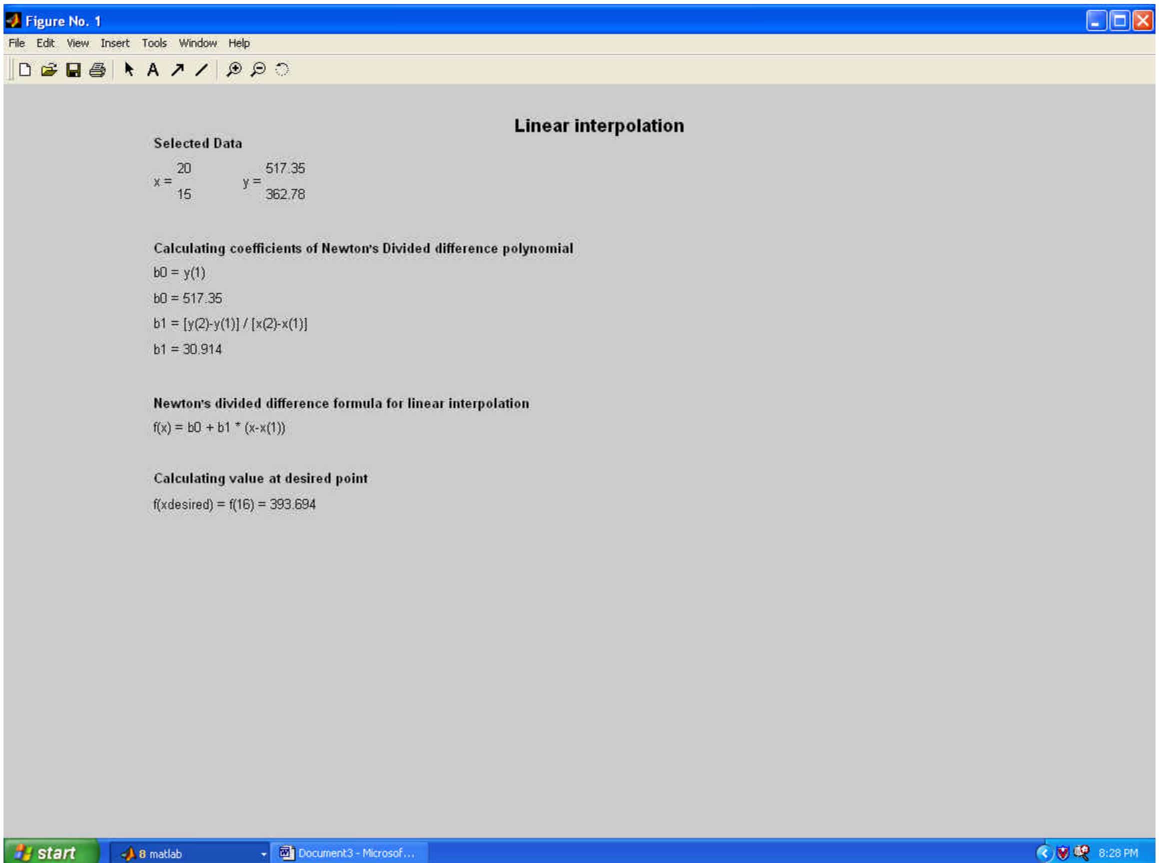


Figure No. 1
File Edit View Insert Tools Window Help
Linear interpolation
Selected Data
x = 20 15
y = 517.35 362.78
Calculating coefficients of Newton's Divided difference polynomial
b0 = y(1)
b0 = 517.35
b1 = [y(2)-y(1)] / [x(2)-x(1)]
b1 = 30.914
Newton's divided difference formula for linear interpolation
f(x) = b0 + b1 * (x-x(1))
Calculating value at desired point
f(xdesired) = f(16) = 393.694
start
8 matlab
Document3 - Microsof...
8:28 PM

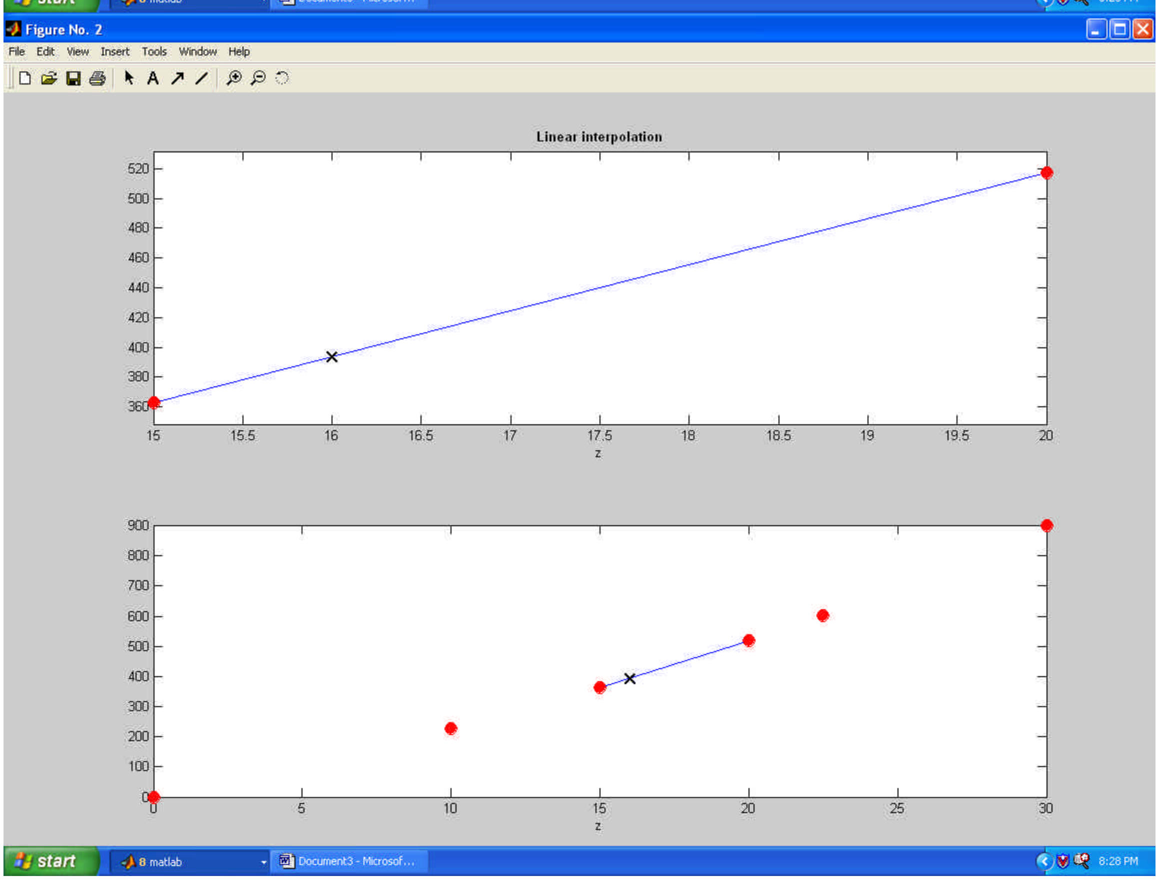
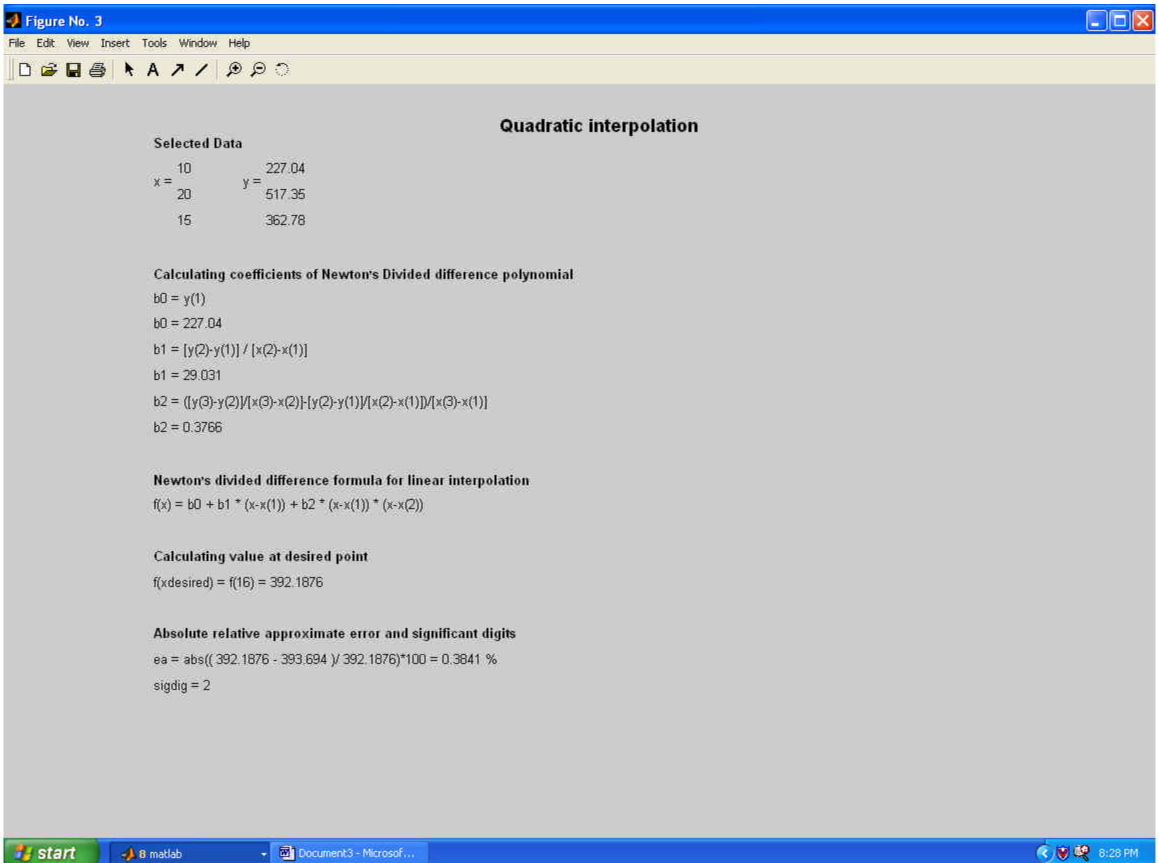


Figure No. 2
File Edit View Insert Tools Window Help
Linear interpolation
520
500
480
460
440
420
400
380
360
15
15.5
16
16.5
17
17.5
18
18.5
19
19.5
20
z
900
800
700
600
500
400
300
200
100
0
0
5
10
15
20
25
30
z
start
8 matlab
Document3 - Microsof...
8:28 PM



Quadratic interpolation

Selected Data

x	y
10	227.04
20	517.35
15	362.78

Calculating coefficients of Newton's Divided difference polynomial

b0 = y(1)

b0 = 227.04

b1 = [y(2)-y(1)] / [x(2)-x(1)]

b1 = 29.031

b2 = ([y(3)-y(2)]/[x(3)-x(2)]-[y(2)-y(1)]/[x(2)-x(1)])/[x(3)-x(1)]

b2 = 0.3766

Newton's divided difference formula for linear interpolation

f(x) = b0 + b1 * (x-x(1)) + b2 * (x-x(1)) * (x-x(2))

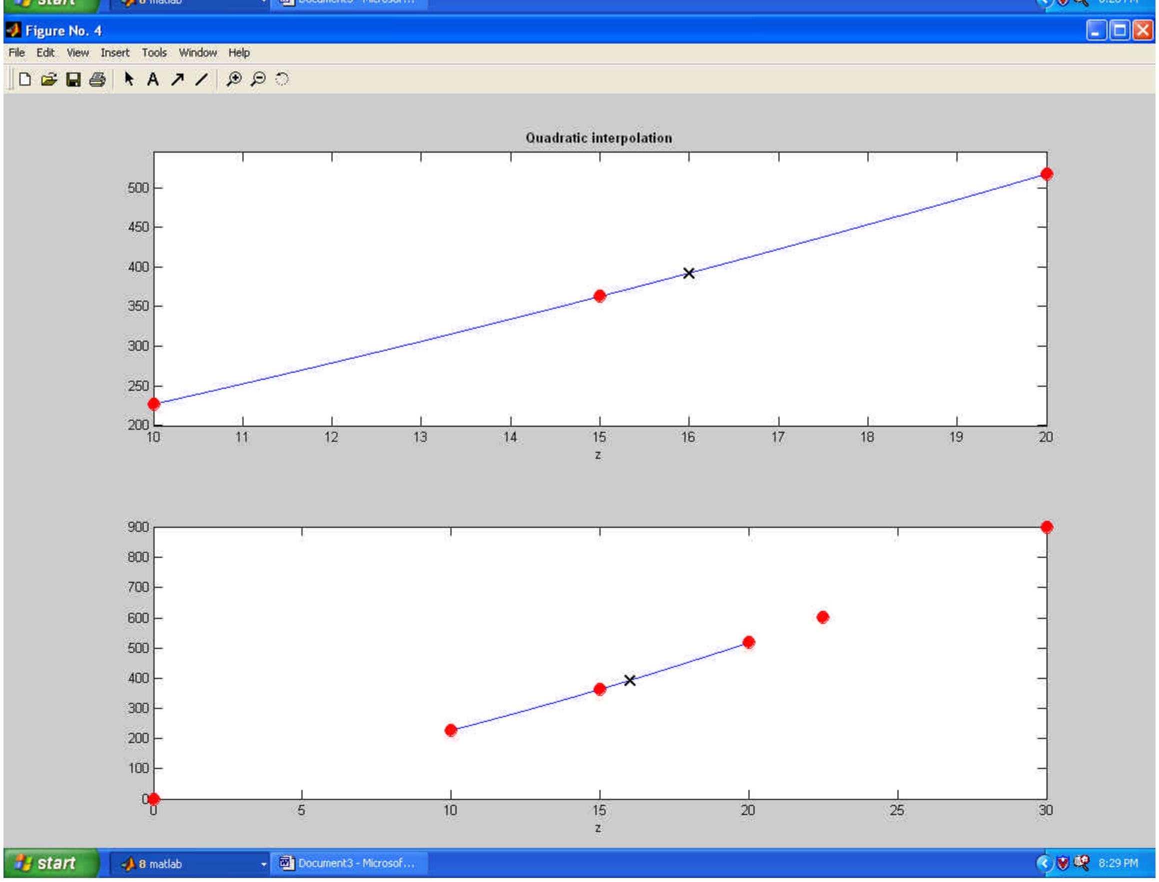
Calculating value at desired point

f(xdesired) = f(16) = 392.1876

Absolute relative approximate error and significant digits

ea = abs((392.1876 - 393.694)/ 392.1876)*100 = 0.3841 %

sigdig = 2



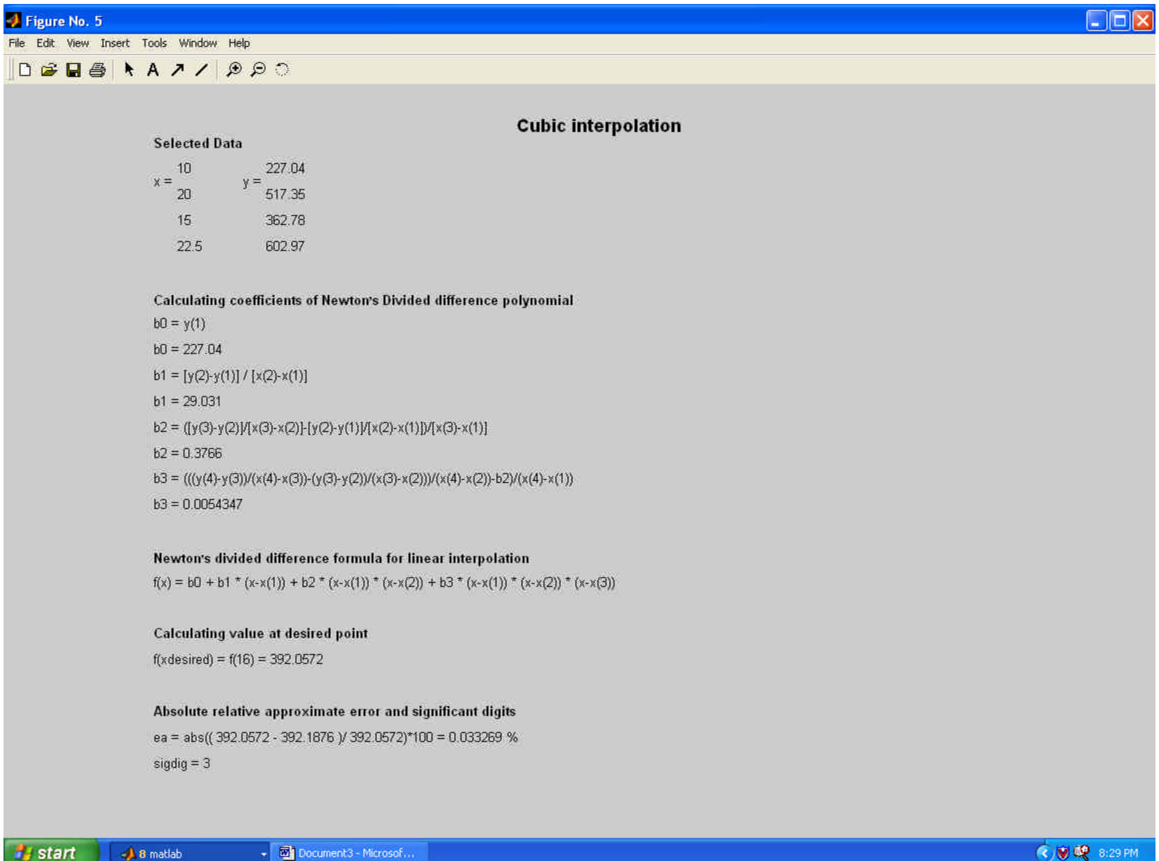


Figure No. 5

Cubic interpolation

Selected Data

x =	y =
10	227.04
20	517.35
15	362.78
22.5	602.97

Calculating coefficients of Newton's Divided difference polynomial

b0 = y(1)

b0 = 227.04

b1 = [y(2)-y(1)] / [x(2)-x(1)]

b1 = 29.031

b2 = ([y(3)-y(2)]/[x(3)-x(2)]-[y(2)-y(1)]/[x(2)-x(1)])/[x(3)-x(1)]

b2 = 0.3766

b3 = (((y(4)-y(3))/(x(4)-x(3))-(y(3)-y(2))/(x(3)-x(2)))/(x(4)-x(2))-b2)/(x(4)-x(1))

b3 = 0.0054347

Newton's divided difference formula for linear interpolation

f(x) = b0 + b1 * (x-x(1)) + b2 * (x-x(1)) * (x-x(2)) + b3 * (x-x(1)) * (x-x(2)) * (x-x(3))

Calculating value at desired point

f(xdesired) = f(16) = 392.0572

Absolute relative approximate error and significant digits

ea = abs((392.0572 - 392.1876)/ 392.0572)*100 = 0.033269 %

sigdig = 3