

**Topic** : Spline Interpolation  
**Simulation** : Graphical Simulation of the Method  
**Language** : Matlab r12  
**Authors** : Nathan Collier, Autar Kaw  
**Date** : 4 November 2002  
**Abstract** : This simulation illustrates the spline 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 functions sort the x and y arrays in ascending x order.
xy=cat(1,x,y);
xy=xy';
xy=sortrows(xy,1);
for i = 1:numel(x)
    x(i)=xy(i,1);
    y(i)=xy(i,2);
end

%%%%%%%%%% LINEAR SPLINE INTERPOLATION %%%%%%%%%%%

figure('Position',wind)
ymax=max(y);
ymin=min(y);
```

```

z=sym('z');
title('Linear spline interpolation','Fontweight','bold','FontSize',14)
text(0,1,'Sorted Data','Fontweight','bold')
midp=(0.95+(0.98-0.03*numel(x)))/2;
text(0,midp,'x =')
text(0.13,midp,'y =')
for i = 1:numel(x)
    text(0.05,0.98-0.03*i,num2str(x(i)))
    text(0.18,0.98-0.03*i,num2str(y(i)))
end

t=0.9-0.02*numel(x)-0.06;
text(0,t,'Equations of the spline','Fontweight','bold')

for i = 1:numel(x)-1
    m=(y(i+1)-y(i))/(x(i+1)-x(i));
    f=m*(z-x(i))+y(i);
    text(0,t-0.04*i-0.01,['f(x)_',num2str(i),' = ',num2str(m),'x + ',num2str(-m*x(i)+y(i))])
end

axis off

figure('Position',wind)
for i = 1:numel(x)-1
    m=(y(i+1)-y(i))/(x(i+1)-x(i));
    z=sym('z');
    f=m*(z-x(i))+y(i);
    if xdesired < x(i+1) & xdesired >= x(i)
        fxldesired=subs(f,z,xdesired);
    end
    ezplot(f,[x(i),x(i+1)])
    hold on
end
plot(x,y,'ro','MarkerSize',10,'MarkerFaceColor',[1,0,0])
plot(xdesired,fxldesired,'kx','Linewidth',2,'MarkerSize',12)
title('Linear spline interpolation','Fontweight','bold','FontSize',14)
text(0.1*(max(x)-min(x)),0.8*(max(y)-min(y)),['f(',num2str(xdesired),') = ',num2str(fxldesired)])
xlim([min(x) max(x)])
ylim([min(y) max(y)])

%%%%%%%%%% QUADRATIC SPLINE INTERPOLATION %%%%%%%%%%%

% The following displays the sorted data for the quadratic spline
figure('Position',wind)

```

```

title('Quadratic spline interpolation','Fontweight','bold','FontSize',14)
text(0,1,'Sorted Data','Fontweight','bold')
midp=(0.95+(0.98-0.03*numel(x)))/2;
text(0,midp,'x =')
text(0.13,midp,'y =')
for i = 1:numel(x)
    text(0.05,0.98-0.03*i,num2str(x(i)))
    text(0.18,0.98-0.03*i,num2str(y(i)))
end
axis off

% The following assembles the matrix needed to find the coefficients of the spline
% polynomials.
% Assembly of A
n=numel(x);
A=zeros(3*(n-1));
for i=1:n-1
    for j=0:1
        for k=0:2
            A(2*i-1+j+1,3*i-3+k+1)=x(i+j)^k;
        end
    end
end
for i=1:n-2
    for j=0:1
        for k=0:1
            A(2*(n-1)+i+1,3*i-2+k+j*3+1)=((-1)^j)*(2*x(i+1))^k;
        end
    end
end
end
A(1,3)=1;
% Assembly of Y
Y=zeros(3*(n-1),1);
for i=0:n-2
    for j=0:1
        Y(2*(i+1)+j)=y(i+j+1);
    end
end
end
% Calculation of C (Coefficient Matrix)
C=inv(A)*Y;

% The following prints the A, Y, and C matrix in the form A^-1*Y=C
figure('Position',wind)
title('Quadratic spline interpolation','Fontweight','bold','FontSize',14)
axis off
hold on

```

```

fo='%1.4g';
sh=1/((3*(n-1))+8);
sv=1/((3*(n-1)));
for c=1:3*(n-1)+8
    for r = 1:3*(n-1)
        if c <= 3*(n-1)
            text(0+(c-1)*sh,1-(r-1)*sv,num2str(A(r,c),fo),'FontSize',10)
        end
        if c == 3*(n-1)+8
            text(0+(c-1)*sh,1-(r-1)*sv,num2str(C(r),fo),'FontSize',10)
        end
        if c == 3*(n-1)+4
            text(0+(c-1)*sh,1-(r-1)*sv,num2str(Y(r),fo),'FontSize',10)
        end
    end
end
end
text((3*(n-1))*sh-0.5*sh,1,'^-1','fontsize',12)
text((3*(n-1)+5)*sh,0.5,'=','fontsize',12)
text((3*(n-1)+1)*sh,0.5,'X','fontsize',12)

% The following prints the equations of the splines
figure('Position',wind)
title('Quadratic spline interpolation','Fontweight','bold','FontSize',14)
text(0,1,'Equations of the spline','Fontweight','bold')
for i=0:n-2
    f=C(3*i+1)+C(3*i+2)*z+C(3*i+3)*z^2;
    text(0,0.98-(i+1)*0.04,['f(x)_' ,num2str(i+1),' = ',num2str(C(3*i+1)),'+
',num2str(C(3*i+2)), 'x + ',num2str(C(3*i+3)), 'x^2']
end
axis off

% The following plots the splines as well as finding the maximum and minimum y value
% These values are needed for plotting
figure('Position',wind)
title('Quadratic spline interpolation','Fontweight','bold','FontSize',14)
for i=0:n-2

    f=C(3*i+1)+C(3*i+2)*z+C(3*i+3)*z^2;
    if xdesired < x(i+2) & xdesired >= x(i+1)
        fxqdesired=subs(f,z,xdesired);
    end
    for j = x(i+1):((x(i+2)-x(i+1))/4):x(i+2)
        if subs(f,z,j) > ymax
            ymax = subs(f,z,j);
        end
        if subs(f,z,j) < ymin

```

```

        ymin = subs(f,z,j);
    end
end
ezplot(f,[x(i+1),x(i+2)])
hold on
end

text(0.1*(max(x)-min(x)),0.8*(max(y)-min(y)),['f(',num2str(xdesired),') =
',num2str(fxqdesired)])
plot(x,y,'ro','MarkerSize',10,'MarkerFaceColor',[1,0,0])
plot(xdesired,fxqdesired,'kx','Linewidth',2,'MarkerSize',12)
title('Quadratic spline interpolation','Fontweight','bold','FontSize',14)
xlim([min(x) max(x)])
tot=(ymax-ymin)*0.05;
ylim([ymin-tot ymax+tot])

```

%%%%%%%%%% CUBIC SPLINE INTERPOLATION %%%%%%%%%%%

```

ymax=max(y);
ymin=min(y);

xx=min(x):(max(x)-min(x))/1000:max(x);
yy=spline(x,y,xx);

figure('Position',wind)
hold on
plot(x,y,'ro','MarkerSize',10,'MarkerFaceColor',[1,0,0])
plot(xx,yy,'b')
fxcdesired=spline(xx,yy,xdesired);
text(0.1*(max(x)-min(x)),0.8*(max(y)-min(y)),['f(',num2str(xdesired),') =
',num2str(fxcdesired)])
plot(xdesired,fxcdesired,'kx','Linewidth',2,'MarkerSize',12)
title('Cubic spline interpolation','Fontweight','bold','FontSize',14)
xlim([min(x) max(x)])
tot=(max(yy)-min(yy))*0.05;
ylim([min(yy)-tot max(yy)+tot])

```

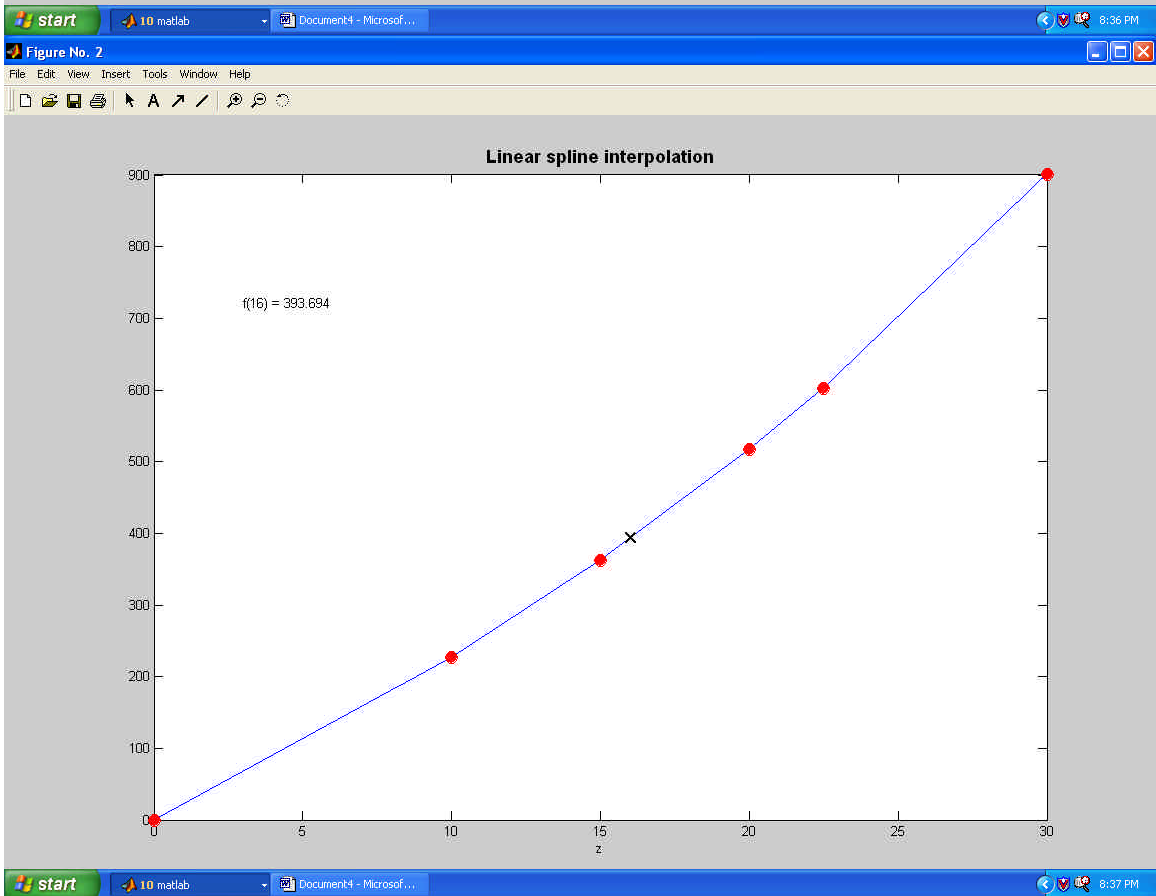
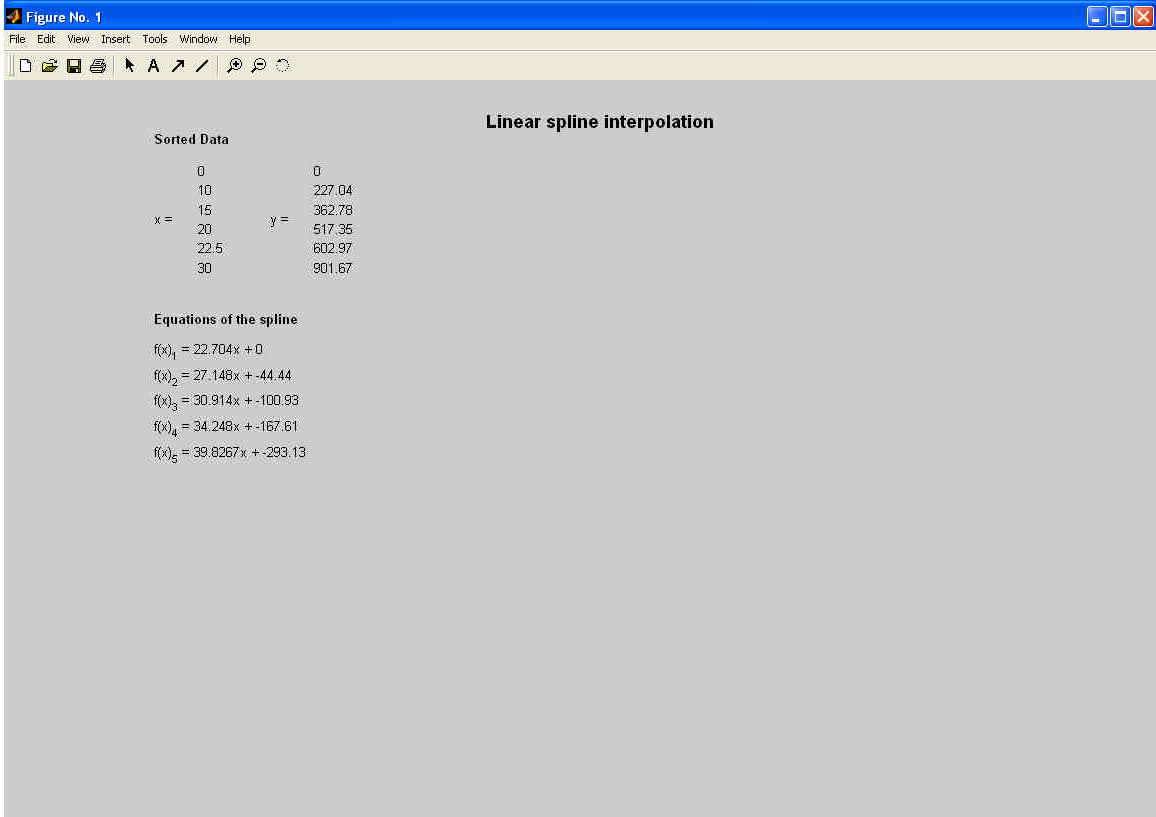
%%%%%%%%%% SUMMARY %%%%%%%%%%%

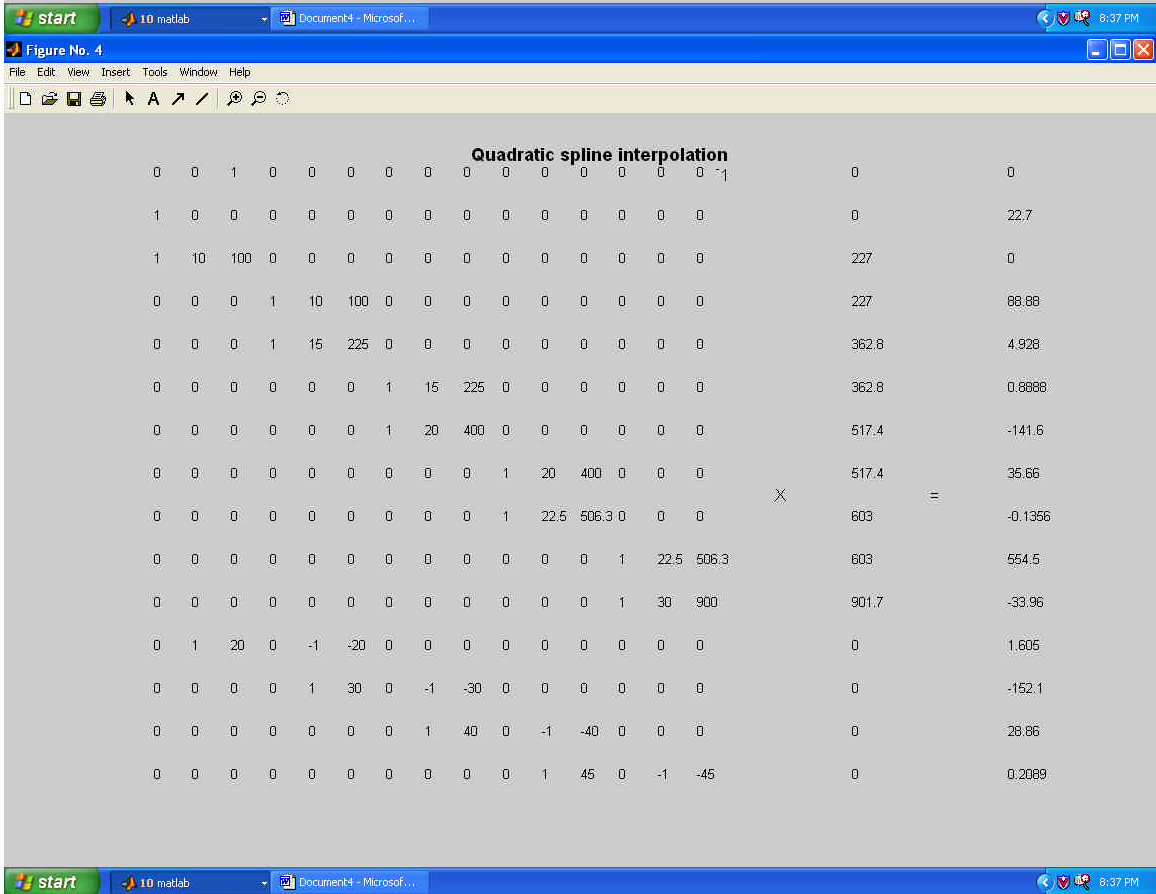
```

figure('Position',wind)
title('Summary of spline interpolation','Fontweight','bold','FontSize',14)
axis off
for j = 0:1
for i = 0:3
rectangle('Position',[0.1+i*0.8/4,0.8-j*0.06,0.8/4,0.06])

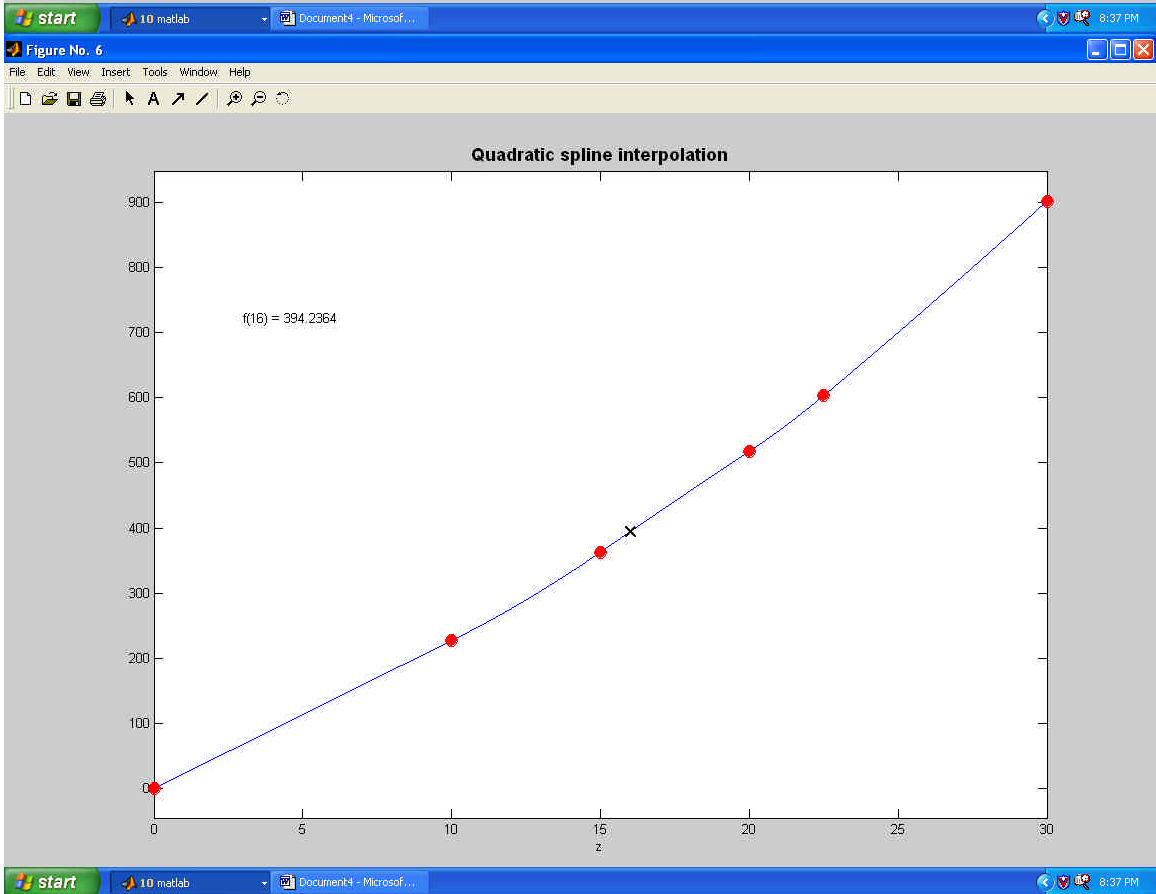
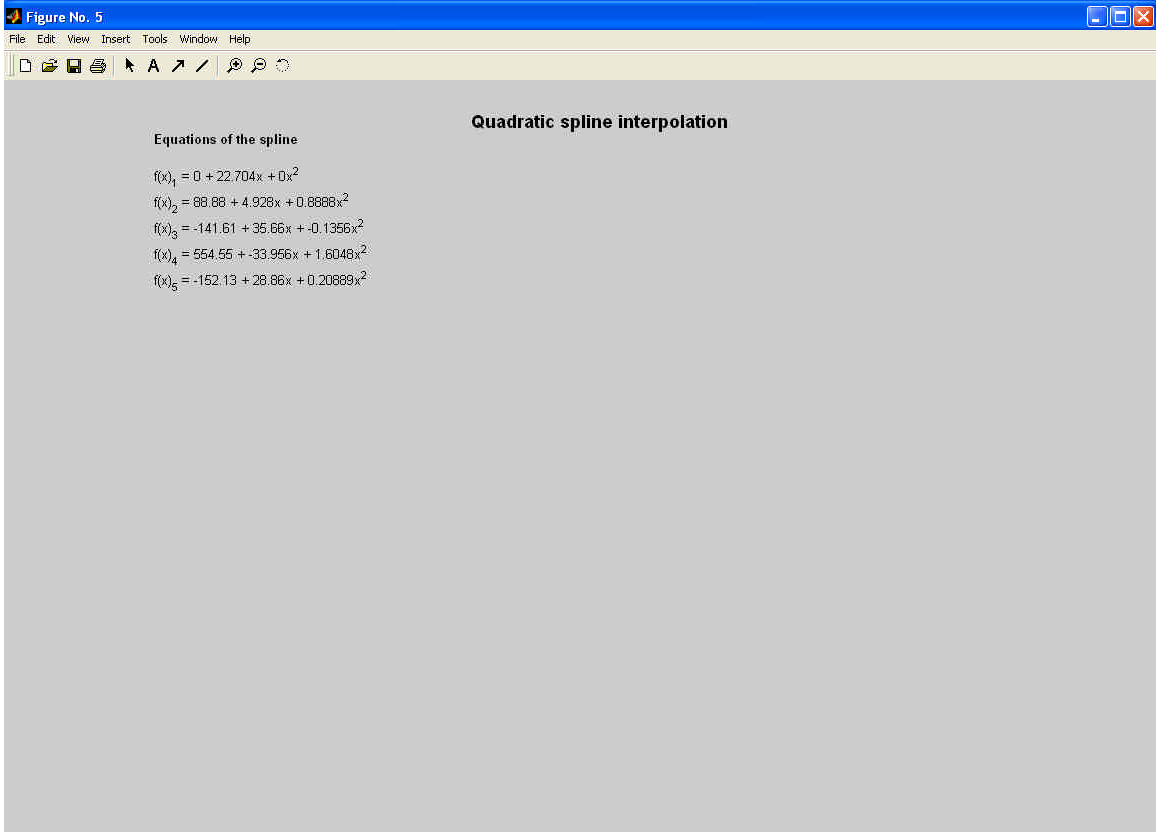
```

```
end
end
xbias = 0.01;
ybias = 0.03;
text(0.1+xbias,0.8+ybias,'x_d_e_s_i_r_e_d')
text(0.3+xbias,0.8+ybias,'Linear')
text(0.5+xbias,0.8+ybias,'Quadratic')
text(0.7+xbias,0.8+ybias,'Cubic')
text(0.1+xbias,0.74+ybias,num2str(xdesired))
text(0.3+xbias,0.74+ybias,num2str(fxldesired))
text(0.5+xbias,0.74+ybias,num2str(fxqdesired))
text(0.7+xbias,0.74+ybias,num2str(fxcdesired))
xlim([0,1])
ylim([0,1])
```









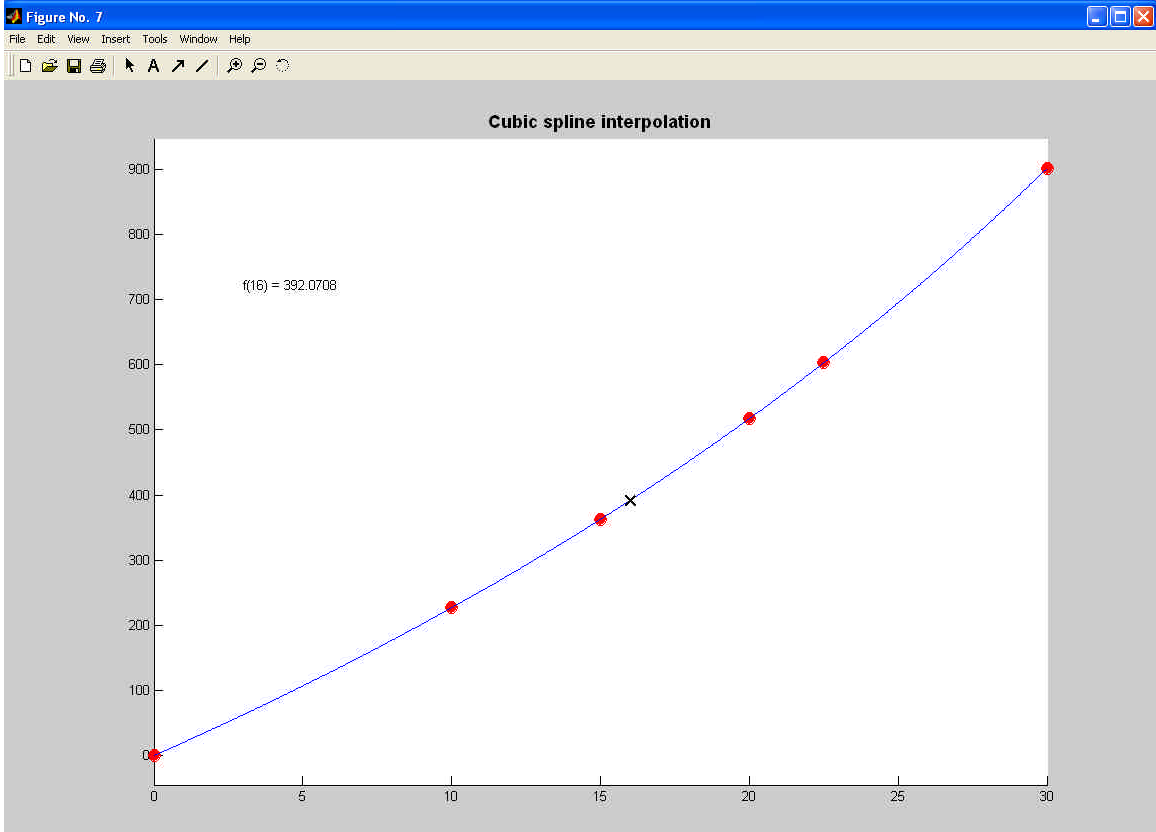


Figure No. 8

File Edit View Insert Tools Window Help

Summary of spline interpolation

$x_{\text{desired}}$	Linear	Quadratic	Cubic
16	393.694	394.2364	392.0708