Bangladesh University of Engineering and Technology

Numerical Technique Laboratory

EEE 212

Experiment No.: 03

Name of the Experiment: Interpolation

Department: EEE

Section: C1

Group: 01

Student No.: 1406131
1406132

Date of Performance: September 17, 2016

Date of Submission: September 23, 2016

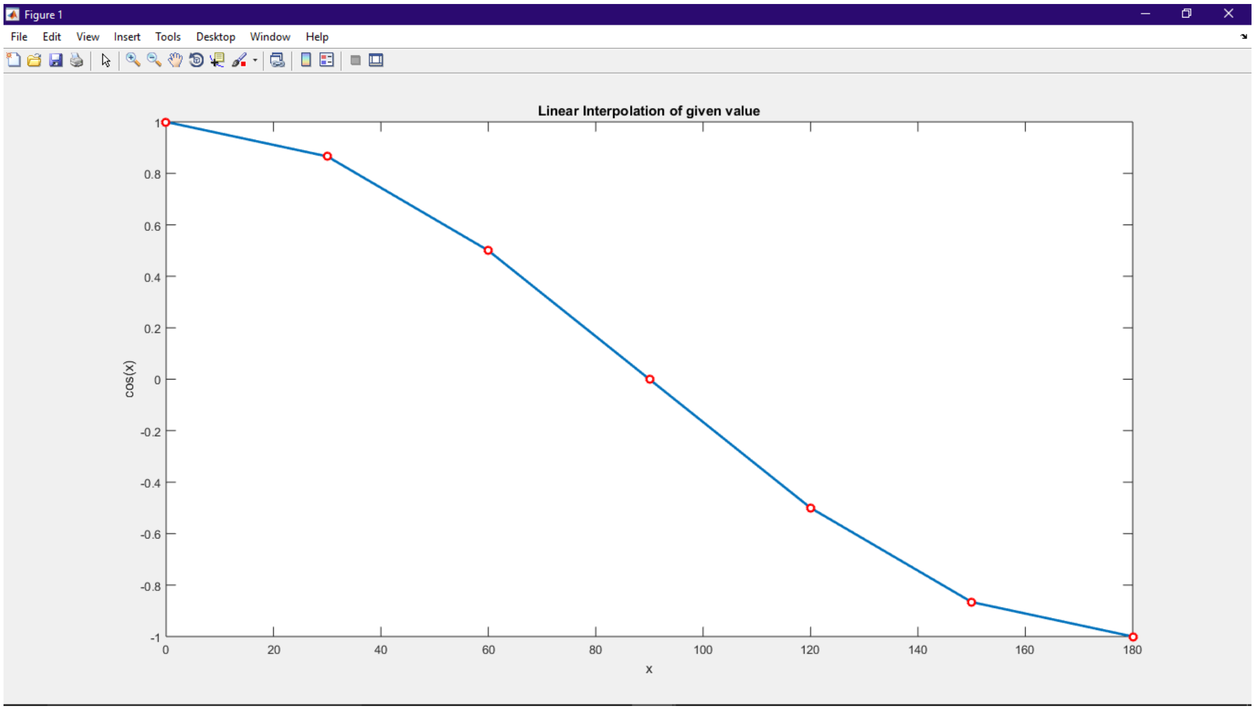
Lagrange Interpolation

Code:

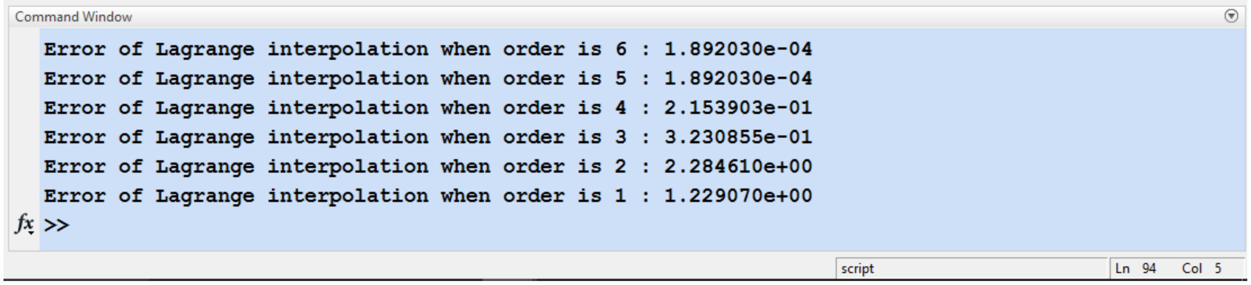
```
clc , close all ;
clear all ;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
x = [ 0 30 60 90 120 150 180 ] ;
y = cosd (x) ;
plot(x,y,'-O','LineWidth',2,'MarkerEdgeColor','r','MarkerSize',6,...
    'MarkerFaceColor','w') ;
title('Linear Interpolation of given value','LineWidth',2);

% Linear Interpolation
lin_x = linspace(0,180) ;   % lin_x is choosen
lin_y = [] ;      % lin_y will be found out by linear interpolation

for i = 1 : length(lin_x);
    if  lin_x(i) >=0 && lin_x(i) < 30
        n = 1;
        a = ( lin_x(i) - x(n+1) ) / (x(n) - x(n+1)) ;
        b = ( lin_x(i) - x(n) ) / (x(n) - x(n+1)) ;
        c = a * y(n) - b * y(n+1) ;
        lin_y = [ lin_y c ] ;

    elseif lin_x(i) >=30 && lin_x(i) <60
        n = 2;
        a = ( lin_x(i) - x(n+1) ) / (x(n) - x(n+1)) ;
        b = ( lin_x(i) - x(n) ) / (x(n) - x(n+1)) ;
        c = a * y(n) - b * y(n+1) ;
        lin_y = [ lin_y c ] ;
    elseif lin_x(i) >=60 && lin_x(i) <90
        n = 3;
        a = ( lin_x(i) - x(n+1) ) / (x(n) - x(n+1)) ;
        b = ( lin_x(i) - x(n) ) / (x(n) - x(n+1)) ;
        c = a * y(n) - b * y(n+1) ;
        lin_y = [ lin_y c ] ;
     elseif lin_x(i) >=90 && lin_x(i) <120
        n = 4;
        a = ( lin_x(i) - x(n+1) ) / (x(n) - x(n+1)) ;
        b = ( lin_x(i) - x(n) ) / (x(n) - x(n+1)) ;
        c = a * y(n) - b * y(n+1) ;
        lin_y = [ lin_y c ] ;
     elseif lin_x(i) >=120 && lin_x(i) <150
        n = 5;
        a = ( lin_x(i) - x(n+1) ) / (x(n) - x(n+1)) ;
        b = ( lin_x(i) - x(n) ) / (x(n) - x(n+1)) ;
        c = a * y(n) - b * y(n+1) ;
        lin_y = [ lin_y c ] ;
     elseif lin_x(i) >=150 && lin_x(i) <=180
        n = 6;
        a = ( lin_x(i) - x(n+1) ) / (x(n) - x(n+1)) ;
        b = ( lin_x(i) - x(n) ) / (x(n) - x(n+1)) ;
        c = a * y(n) - b * y(n+1) ;
        lin_y = [ lin_y c ] ;
    end
end



order = length(x) - 1 ;
sum = 0 ;
lagrange_x = linspace(0,180,1000) ;         % lagrange_x is choosen
lagrange_y = [] ;            % lagrange_y will be found out by lagrange
                             % interpolaion

% Lagrange Interpolation

while order > 0
    for n = 1 : length(lagrange_x)
        for i = 1 : order + 1
            weight = 1 ;
            for j = 1 : order + 1
                if j ~= i
                   a = lagrange_x(n) - x(j) ;
                   b = x(i)-x(j) ;
                   weight = weight * ( a/b ) ;
                end
            end
            sum = sum + weight * y(i) ;
        end
        lagrange_y = [lagrange_y sum ];
        sum = 0 ;
    end
    figure

    subplot(3,1,1) ;
    plot(lin_x,lin_y,'LineWidth',2) ;
    title('Linear Interpolation','LineWidth',2);

    subplot(3,1,2) ;
    plot(lagrange_x, lagrange_y,'LineWidth',2);
    title('Lagrange Interpolation') ;

    original_x = linspace(0,180,1000) ;
    original_y = cosd(original_x) ;

    error = lagrange_y - original_y ;
    error = abs(error)  ;
    subplot(3,1,3)  ;
    plot(error,'LineWidth',2)  ;
    title('Error','LineWidth',2);
    value = max(abs(error));
    fprintf('Error of Lagrange interpolation when order is %d : %d
\n',order,value);

    lagrange_y = [] ;
    order = order - 1 ;
end
```

Output:

Linear Interpolation of given points:

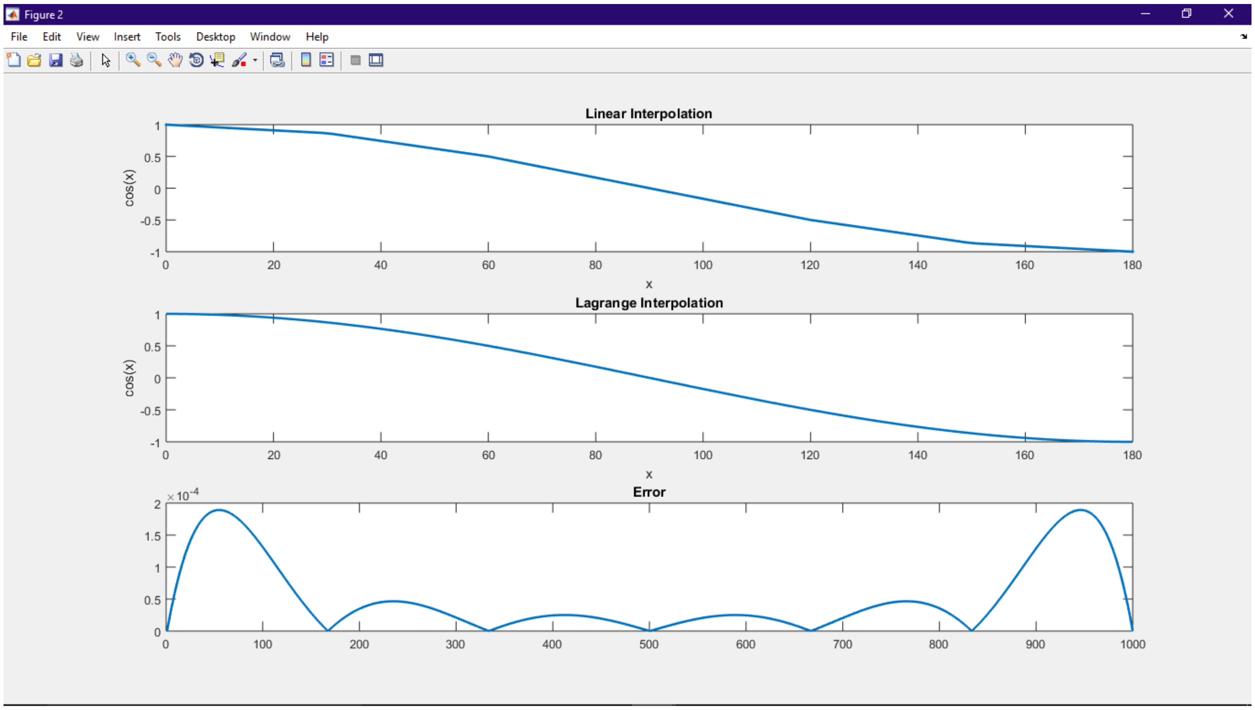


Command Window Output of Lagrange interpolation:

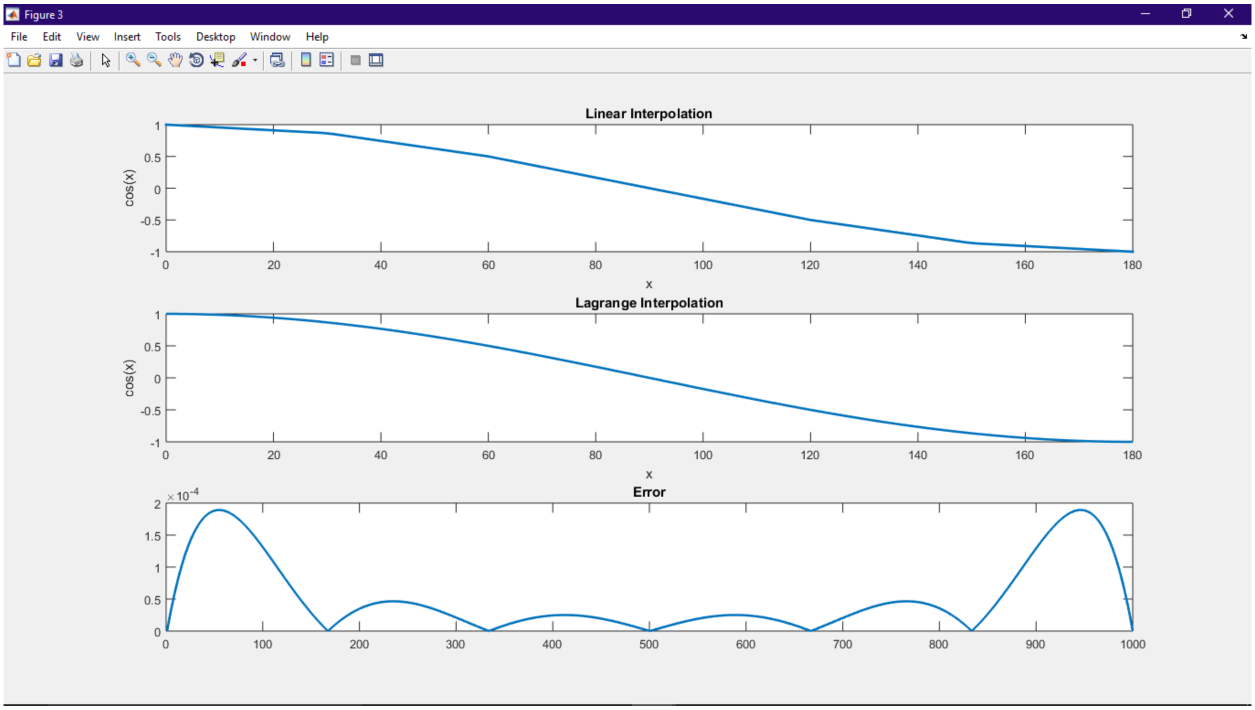


```
Error of Lagrange interpolation when order is 6 : 1.892030e-04
Error of Lagrange interpolation when order is 5 : 1.892030e-04
Error of Lagrange interpolation when order is 4 : 2.153903e-01
Error of Lagrange interpolation when order is 3 : 3.230855e-01
Error of Lagrange interpolation when order is 2 : 2.284610e+00
Error of Lagrange interpolation when order is 1 : 1.229070e+00
fx >>
```

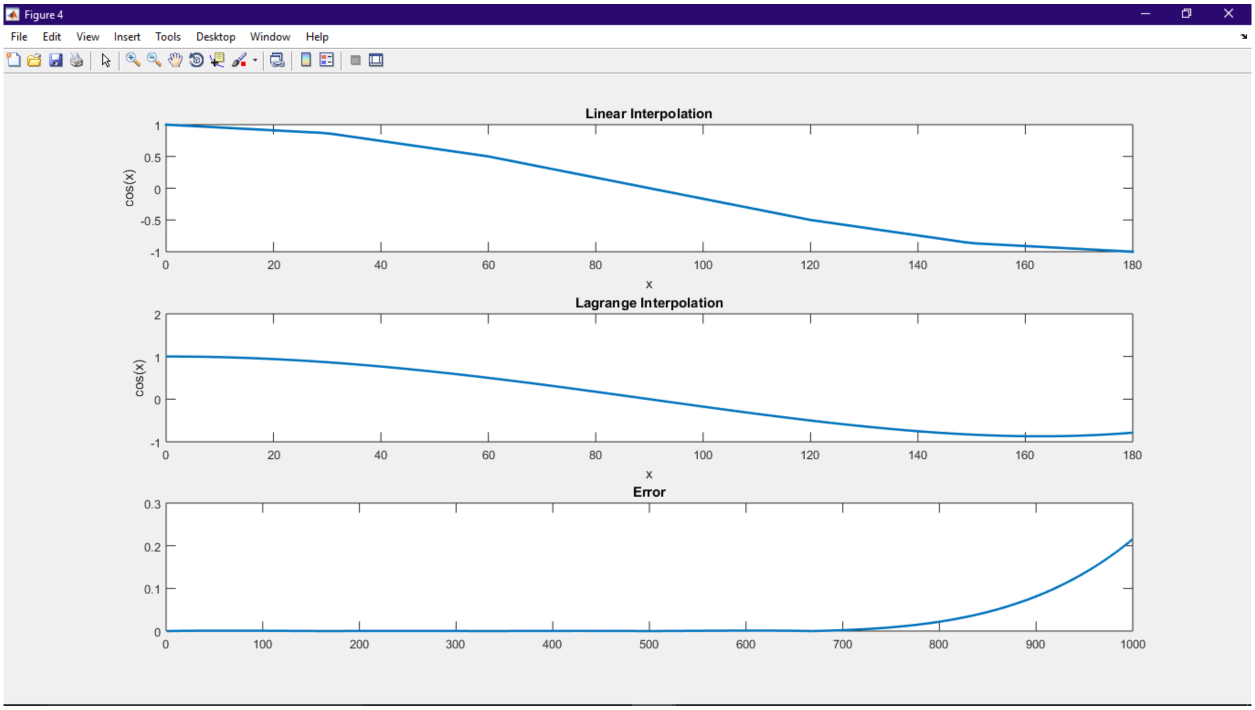
Lagrange Interpolation of Order: 6



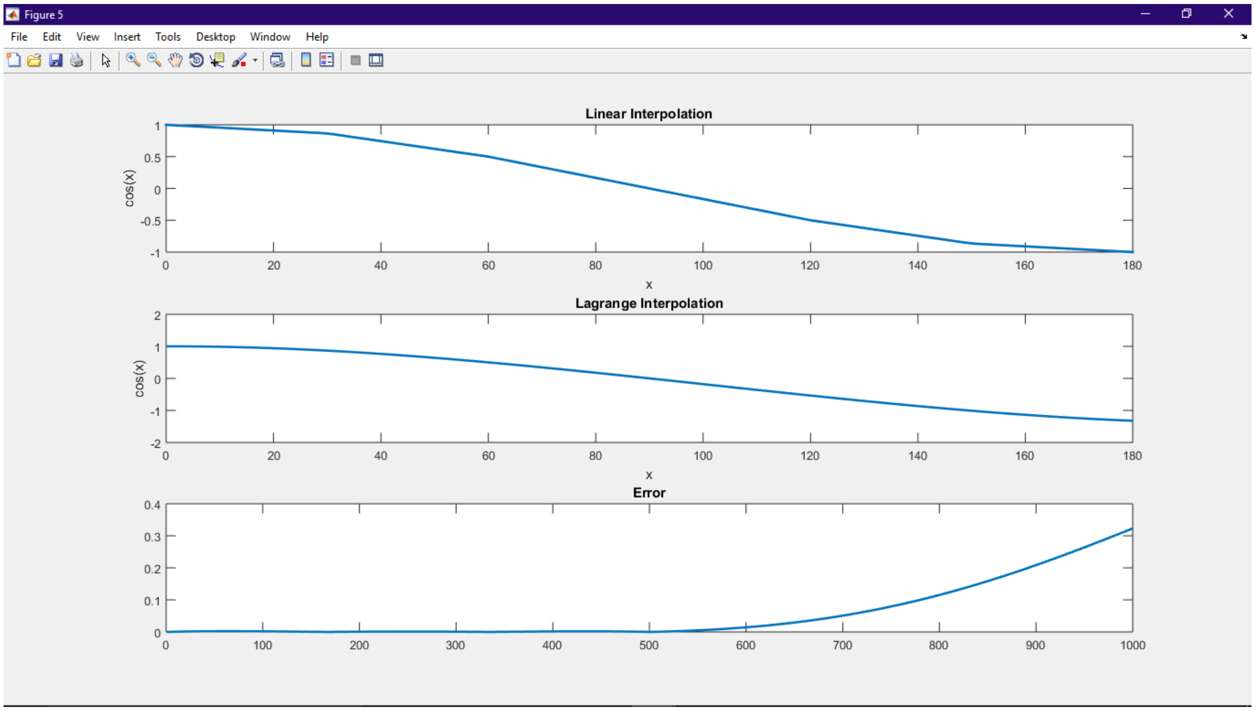
Lagrange Interpolation of Order: 5



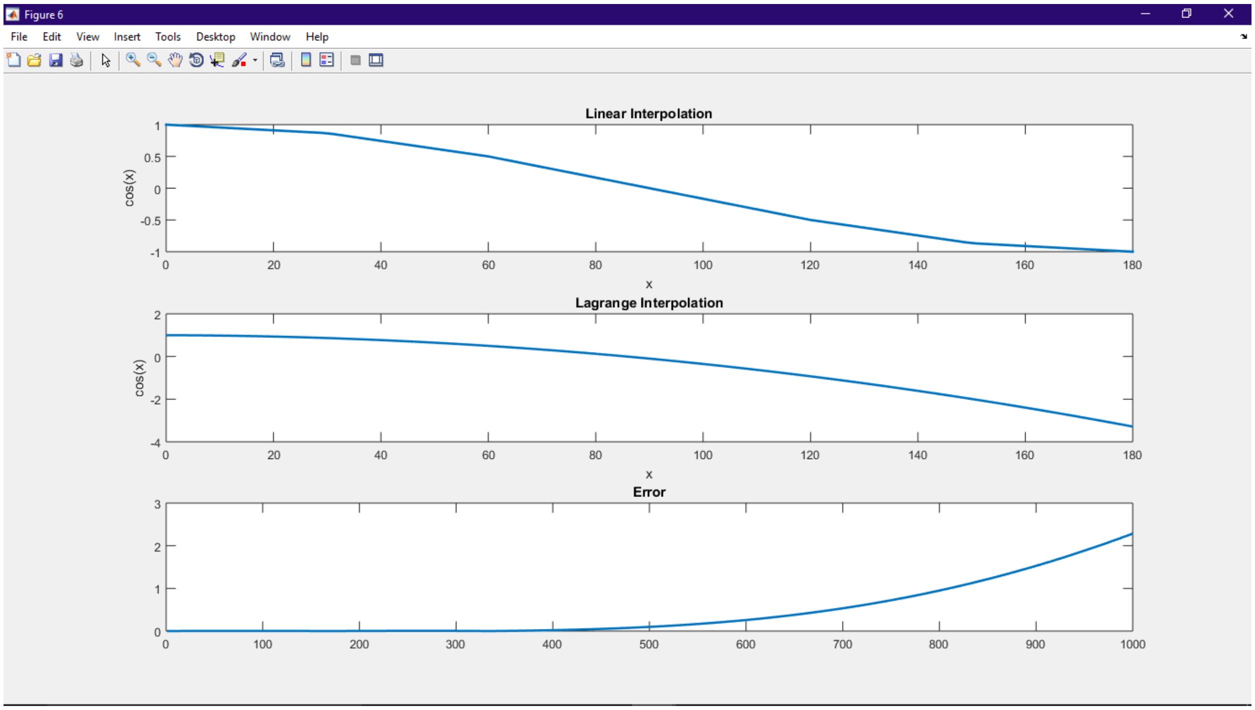
Lagrange Interpolation of Order: 4



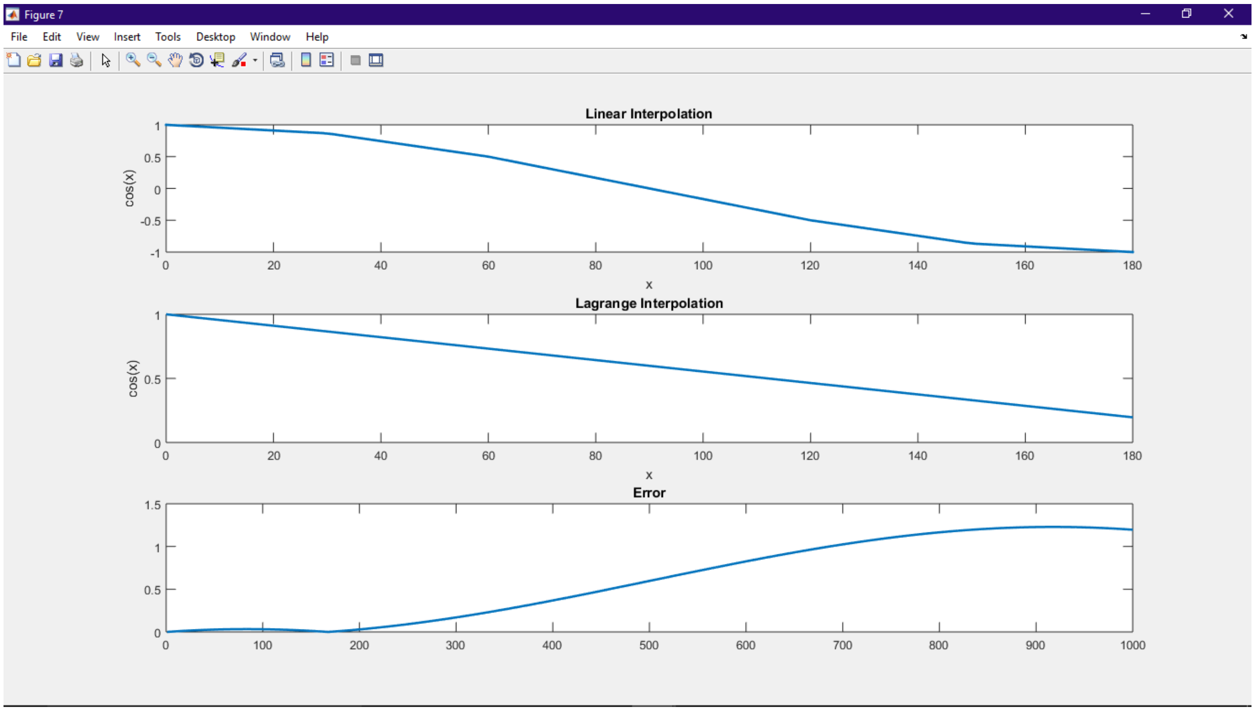
Lagrange Interpolation of Order: 3



Lagrange Interpolation of Order: 2



Lagrange Interpolation of Order: 1



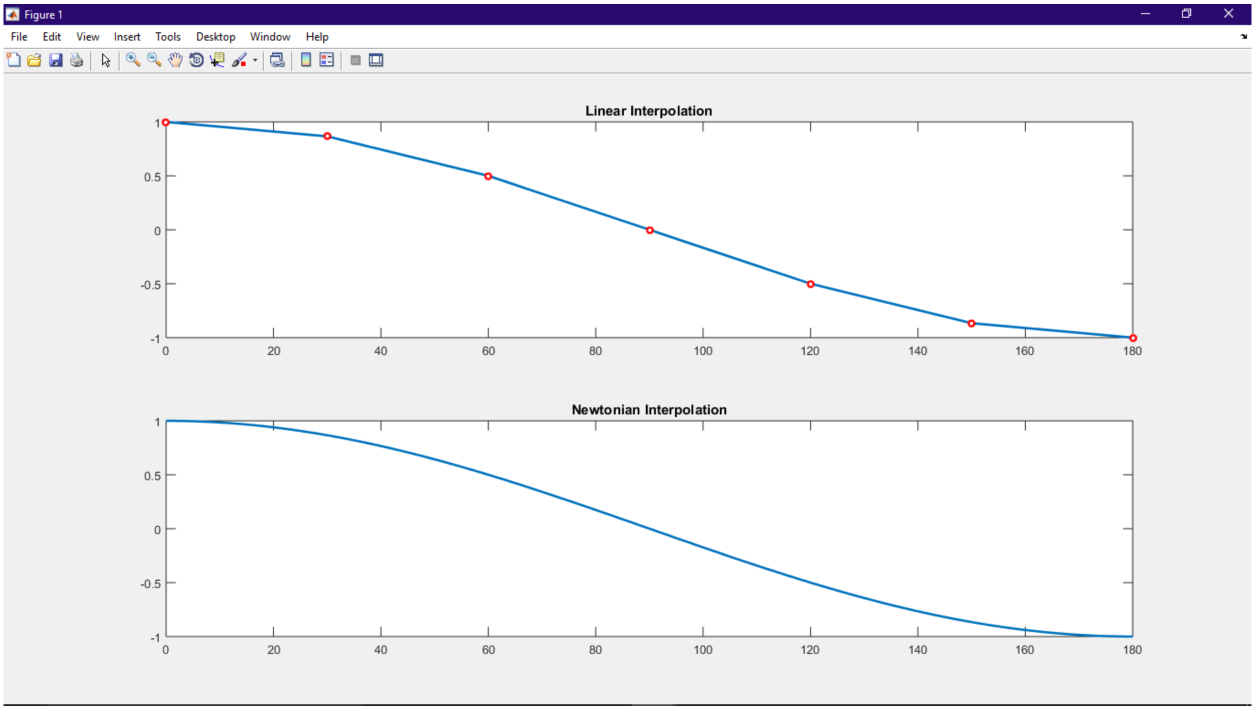
Newton Interpolation

Code:

```
clear all ;
clc , close all ;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
x = [0 30 60 90 120 150 180] ;
y = cosd(x) ;
subplot(2,1,1) ;
plot(x,y,'-O','MarkerSize',5,'MarkerEdgeColor','r','LineWidth',2,...
     'MarkerFaceColor','w') ;
title('Linear Interpolation','LineWidth',2);

% Newtonian Interpolation
d = y ;
d = d' ;
n = length(x) ;

for j = 2 : n
    for i = j : n
        a = d(i,j-1) - d(i-1,j-1) ;
        b = x(i) - x(i-j+1) ;
        d(i,j) = a/b ;
    end
end

% d is the van der monde matrix
d = diag(d) ;
d = d' ;
newton_x = linspace(0,180,1000) ;
newton_y = [] ;

order = length(x)-1 ;
sum = 0 ;
for i = 1 : length(newton_x)
    for j = 1 : order + 1
        p = 1 ;
        for k = 1 : j - 1
            p = p * (newton_x(i) - x(k));
        end
        sum = sum + p * d(j) ;
    end
    newton_y = [newton_y sum] ;
    sum = 0 ;
end

subplot(2,1,2) ;
plot(newton_x,newton_y,'LineWidth',2) ;
```

```
title('Newtonian Interpolation','LineWidth',2);

figure

original_x = linspace(0,180,1000) ;
original_y = cosd(original_x) ;

error = abs (original_y - newton_y) ;
plot(error,'LineWidth',2) ;
title('Error','LineWidth',2) ;
```

Output:



Error:

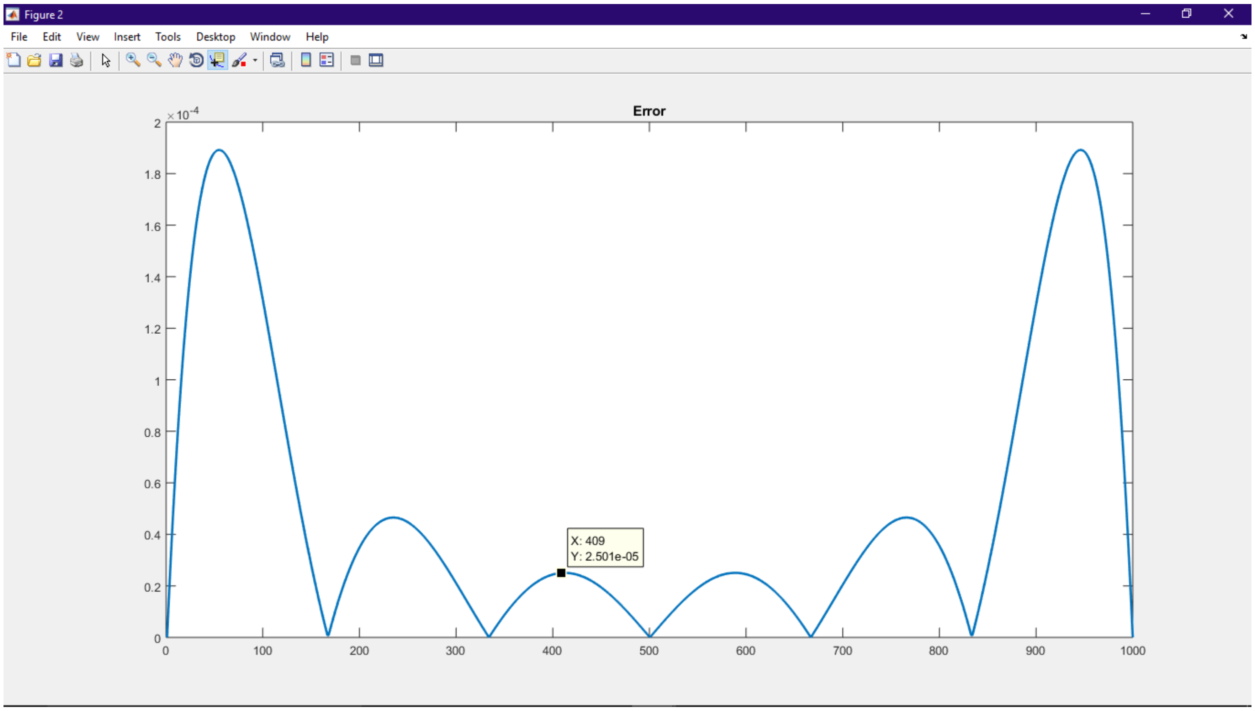


Figure 2
File
Edit
View
Insert
Tools
Desktop
Window
Help
Error
× 10-4
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
100
200
300
400
500
600
700
800
900
1000
X: 409
Y: 2.501e-05

Audio wave Interpolation

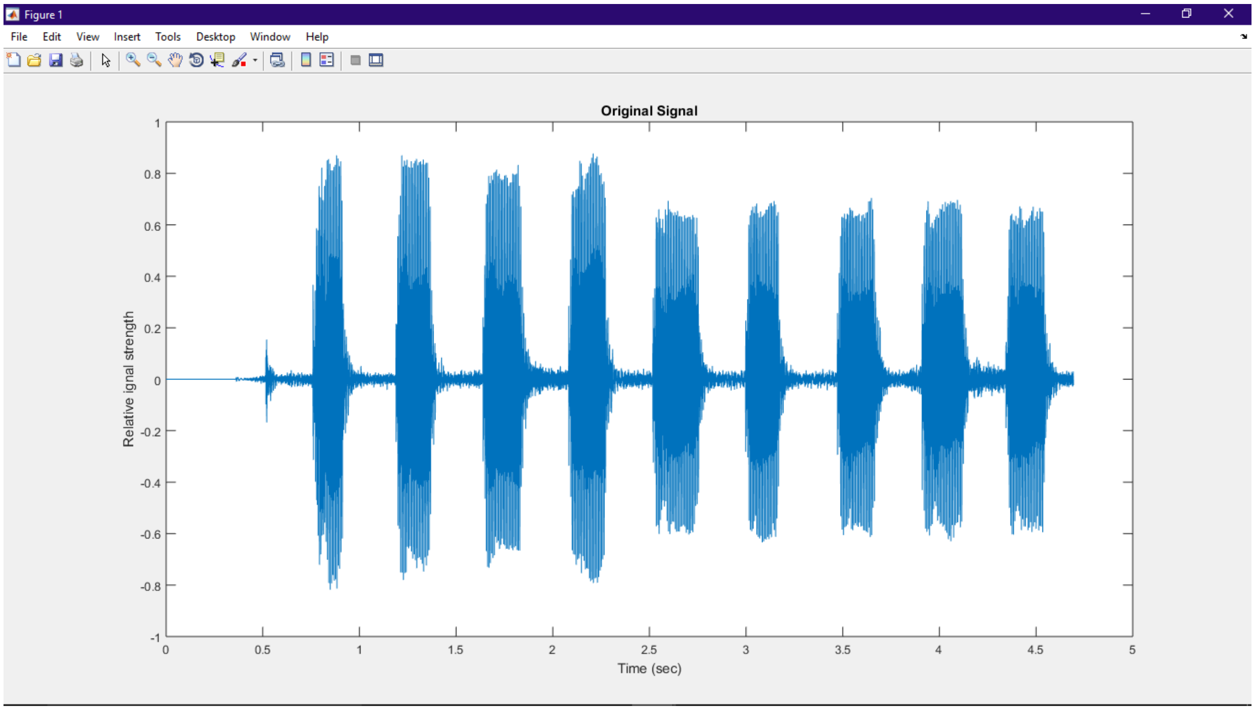
Code:

```
clear all , clc;
clf , close all ;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% record voice %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[stereo,fs] = wavread('voice.wav');

% calculate time of the audio
time = (1/fs) * length(stereo) ;

% plotting original audio signal
t = linspace (0, time, length(stereo) ) ;
plot(t,stereo) ;           % turning column vector to row vector
xlabel('Time (sec)','LineWidth',2);
ylabel('Relative ignal strength','LineWidth',2) ;
title('Original Signal','LineWidth',2) ;

figure

%plotting reduce audio signal
reduce_wave_length = stereo(44111*4.8540:44111*4.85433);
subplot(2,1,1);
plot(reduce_wave_length,'LineWidth',2);
xlabel('Time (sec)','LineWidth',2);
ylabel('Relative ignal strength','LineWidth',2) ;
title('Reduced audio Signal','LineWidth',2) ;
grid on;

% taking points
taken_x = 1 : 1 : 15;
taken_y = [-.1801 -.183 -.1911 -.208 -.2268 -.2418 -.2575 -.2762 -.2878 -
.2924 -.2974 -.295 -.2842 -.2741 -.2582];

new_x = [1.002 2.003 3.001 4.00305 5.00201 6.001 7.0035 8.002 9.0011 10.001
11.0016 12.0041 13.0015 14.00224 14.998];

for k = 1:length(new_x)
    sum = 0;
    for i = 1:length(taken_x)
        P = taken_y(i);
        for j = 1 : length(taken_x)
            if j~=i
                P = P*(new_x(k)-taken_x(j))/(taken_x(i)-taken_x(j));
            end
        end
        sum = sum + P;
    end
    new_y(k) = sum;
end

% plotting interpolated audio signal
subplot(2,1,2);
plot(new_x,new_y,'LineWidth',2);
xlabel('Time (sec)','LineWidth',2);
```

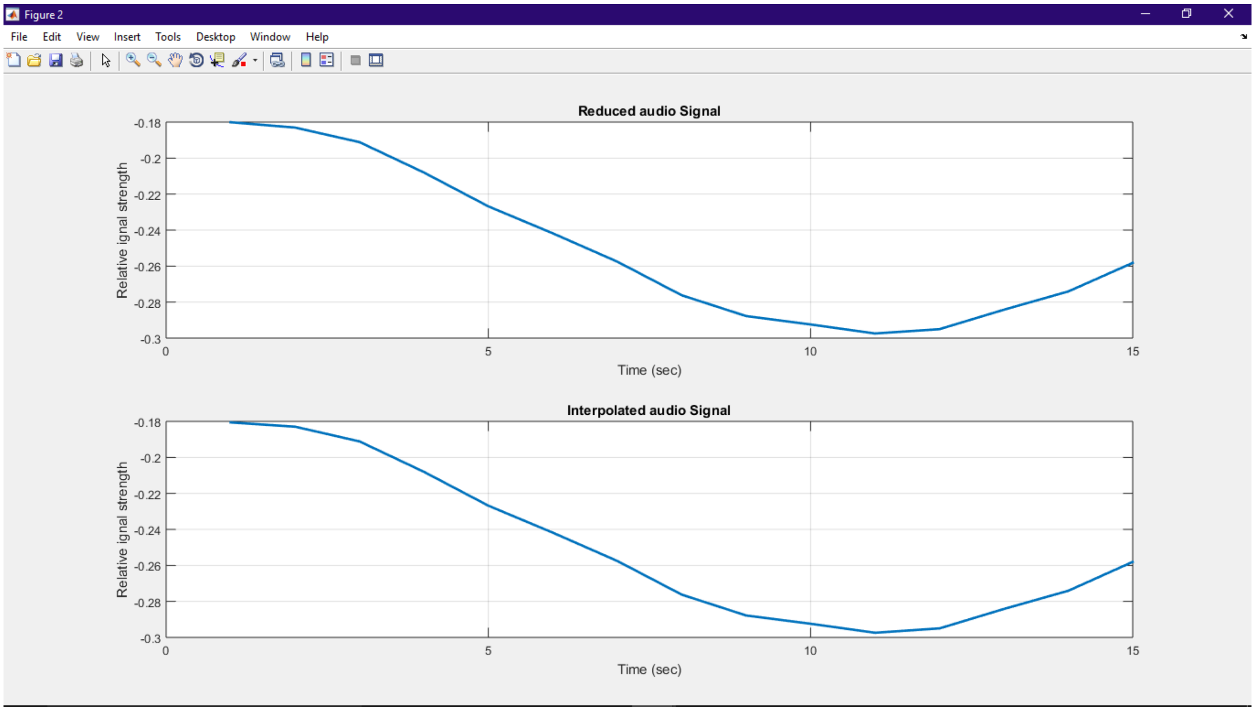
```
ylabel('Relative ignal strength','LineWidth',2) ;
title('Interpolated audio Signal','LineWidth',2) ;
grid on;

figure

% plotting error
Error = taken_y - new_y;
plot(Error,'LineWidth',2);
ylabel('Error','LineWidth',2);
title('Error','LineWidth',2) ;
grid on;
```

Output:

Original audio signal:



Reduced and interpolated audio signal:



Error:

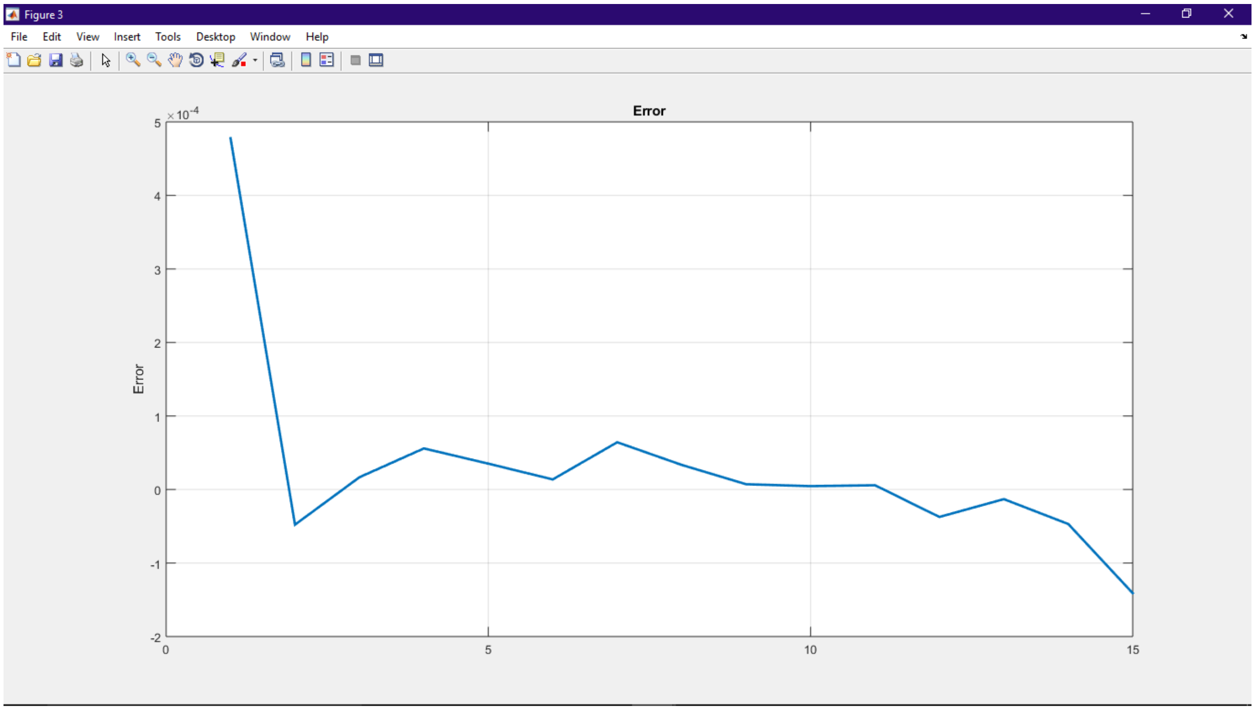


Figure 3
File Edit View Insert Tools Desktop Window Help
Error
× 10^-4
Error
5
4
3
2
1
0
-1
-2
0
5
10
15