ENSC 180 - Assignment 7

Shengcong Zhou
Lab group 9
sza111@sfu.ca

1.

```
V=3:4:133;
M=[0 2 0; 8 0 3; 0 0 5];

a=M+3*(M~=0 & M<5)
b=V+2*(mod(V,7)==0)

copy=V';
copy=num2str(copy)-'0'; %break down numbers into its component digits
[row,col]=find(copy==7); %identify the index of number which contains 7
c=sort(V(row))

Num1=input('Enter the 1st number of v1\n');
Num2=input('Enter the 2nd number of v1\n');
%increment=difference between 1st and 2nd number
%increment may be negative
increment=Num2-Num1;
%calculate the last number of an array of length 30
NumEnd=Num1+increment*29;
%Create an array
v1=Num1:increment:NumEnd;
%Create two copies of v1
copy1=v1;
copy2=v1;
%modify copies to satisfy the following
%-length of the two copy arrays are the same
%-copy2 starts with the 2nd element of copy1 and follows the same sequence
copy1(30)=[];
copy2(1)=[];
%the sum of copy1 and copy2 is an array of v1's consecutive numbers
v2=copy1+copy2
```

Output

```
a =

     0     5     0
     8     0     6
     0     0     5


b =

  Columns 1 through 7

     3     9    11    15    19    23    27

  Columns 8 through 14
```

```
  31    37    39    43    47    51    55

 Columns 15 through 21

  59    65    67    71    75    79    83

 Columns 22 through 28

  87    93    95    99   103   107   111

 Columns 29 through 33

 115   121   123   127   131


c =

 Columns 1 through 7

   7    27    47    67    71    75    79

 Columns 8 through 10

  87   107   127

Enter the 1st number of v1
1
Enter the 2nd number of v1
2

v2 =

 Columns 1 through 11

   3     5     7     9    11    13    15    17    19    21    23

 Columns 12 through 22

  25    27    29    31    33    35    37    39    41    43    45

 Columns 23 through 29

  47    49    51    53    55    57    59
```

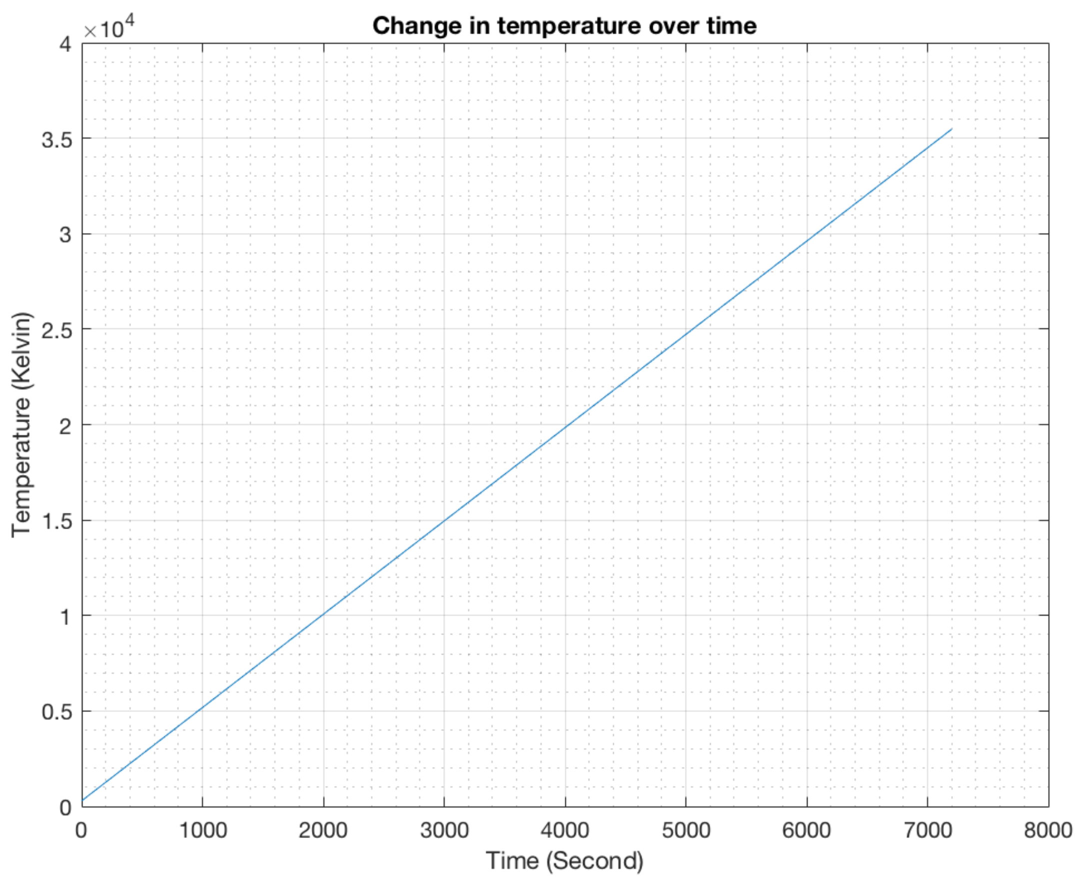
2.
a)b)

```
LengthWire=5; %m
Current=8.4; %A
RadiusWire=0.002; %m
Density=2.6989e06; %g/m^3
MeltingPoint=660.37+273.15; %K
ElectricalResistivity=2.6548e-08; %Ome-m
T0_inK=20+273.15; %K
```

```
SpecificHeat=0.215*4.184; %j/gK

t=0:0.01:7200;
Resistance=ElectricalResistivity*LengthWire/(pi*RadiusWire^2);
Power=Current^2*Resistance;
Energy=Power.*t;
VolumeWire=pi.*RadiusWire.^2.*LengthWire;
Mass=VolumeWire.*Density;
dT=Energy./(Mass.*SpecificHeat);
T=T0_inK+dT;

plot(t,T)
title('Change in temperature over time')
xlabel('Time (Second)')
ylabel('Temperature (Kelvin)')
grid on
grid minor

for i=1:length(t)
    if T(i)>=MeltingPoint
        fprintf('The temperature does reach the melting point at\nt= %.2f\n',t(i));
        break
    end
end
```

Output

The temperature does not reach the melting point



c)

```
LengthWire=5; %m
RadiusWire=0.002; %m
Density=2.6989e06; %g/m^3
MeltingPoint=660.37+273.15; %K
ElectricalResistivity=2.6548e-08; %Ome-m
T0_inK=20+273.15; %K
SpecificHeat=0.215*4.184; %j/gK

t=0:0.01:7200;
for I=0:0.001:3
    Resistance=ElectricalResistivity*LengthWire/(pi*RadiusWire^2);
    Power=I.^2.*Resistance;
    Energy=Power.*t;
    VolumeWire=pi.*RadiusWire.^2.*LengthWire;
    Mass=VolumeWire.*Density;
    dT=Energy./(Mass.*SpecificHeat);
    T=T0_inK+dT;
 if max(T)>=MeltingPoint
     fprintf('The maximum current= %0.3f A\n',I);
     break
 end
end
```

Output

The maximum current= 35.839 A

3.

```
syms x y
y=(2*x^4-x^2+x-1)/(x^2-2);
D1=simplify(diff(y)); %1st derivative of y
D2=simplify(diff(y,2)); %2nd derivative of y
D3=simplify(diff(y,3)); %3rd derivative of y
D4=simplify(diff(y,4)); %4th derivative of y

fplot(y,'r')
xlim([-3,3])
ylim([-200,200])
grid on
grid minor
hold on
fplot(D1,'g')
fplot(D2,'b')
title("y,y',y'' ; y=(2*x^4-x^2+x-1)/(x^2-2)")
xlabel('x')
ylabel('y')

%find local maxima & minima
%initialize x-coordinates array
Maxima=[];
Minima=[];
InflectionP=[];
%initialize y-coordinates array
```

```
Maximaf=[];
Minimaf=[];
InflectionPf=[];
%find critical points for y y' y''
%values are converted from sym to double
%complex answers are ignored
criticalP_y=double(solve(D1));
index_y=find(imag(criticalP_y)==0);
criticalP_y=criticalP_y(index_y);

criticalP_D1=double(solve(D2));
index_D1=find(imag(criticalP_D1)==0);
criticalP_D1=criticalP_D1(index_D1);

criticalP_D2=double(solve(D3));
index_D2=find(imag(criticalP_D2)==0);
criticalP_D2=criticalP_D2(index_D2);
%concavities at critical points
concavity_y=double(subs(D2,x,criticalP_y));
concavity_D1=double(subs(D3,x,criticalP_D1));
concavity_D2=double(subs(D4,x,criticalP_D2));

for i=1:3 %for loop to repeat the same process for y y' y''
  switch i
      case 1
         criticalP=criticalP_y;
         concavity=concavity_y;
         index=index_y;
         f=y;
      case 2
         criticalP=criticalP_D1;
         concavity=concavity_D1;
         index=index_D1;
         f=D1;
      case 3
         criticalP=criticalP_D2;
         concavity=concavity_D2;
         index=index_D2;
         f=D2;
  end

  %determine each critical point is either max or min or inflection point
   for j=1:length(index)
      if concavity(j)<0 %max
         Maxima=[Maxima ; criticalP(j)];
         Maximaf=[Maximaf ; double(subs(f,criticalP(j)))];
      elseif concavity(j)>0 %min
         Minima=[Minima ; criticalP(j)];
         Minimaf=[Minimaf ; double(subs(f,criticalP(j)))];
      else %inflection point
         InflectionP=[InflectionP ; criticalP(j)];
         InflectionPf=[InflectionPf ; double(subs(f,criticalP(j)))];
      end
   end
```

```
end

fprintf('Maxima at\n\tx\ty\n'); disp([Maxima     Maximaf]);
plot(Maxima, Maximaf,'ro')
fprintf('Minima at\n\tx\ty\n'); disp([Minima     Minimaf]);
plot(Minima, Minimaf,'bo')

%find inflection points
InflectionP_y=double(solve(D2));
index_y=find(imag(InflectionP_y)==0);
InflectionP_y=InflectionP_y(index_y);

InflectionP_D1=double(solve(D3));
index_D1=find(imag(InflectionP_D1)==0);
InflectionP_D1=InflectionP_D1(index_D1);

InflectionP_D2=double(solve(D4));
index_D2=find(imag(InflectionP_D2)==0);
InflectionP_D2=InflectionP_D2(index_D2);

InflectionP=[InflectionP ; InflectionP_y ; InflectionP_D1 ; InflectionP_D2];
InflectionPf=[InflectionPf ; double(subs(y,InflectionP_y)) ; double(subs(D1,InflectionP_D1)) ;
double(subs(D2,InflectionP_D2))];

fprintf('Inflection points at\n\tx\ty\n'); disp([InflectionP     InflectionPf]);
plot(InflectionP, InflectionPf,'go')

%find vertical asymptotes
for k=1:3
   switch k
      case 1
         [num,den]=numden(y);
      case 2
         [num,den]=numden(D1);
      case 3
         [num,den]=numden(D2);
   end

   asymptotes=double(solve(den));

end
plot(asymptotes,0,'kx')
fprintf('Vertical asymptotes at x=\n'); disp(asymptotes);

legend('y',"y'","y''",'Maxima','Minima','Inflection points','asymptotes')
hold off
```

Output

```
>> q3
Maxima at
         x         y
   -0.8360    1.1974
    0.3259   -0.1924
```

-0.1026   1.5766

Minima at

x	y
-1.8249	12.0473
1.9496	14.4608
-0.5035	-1.1021
-19.4890	3.9999

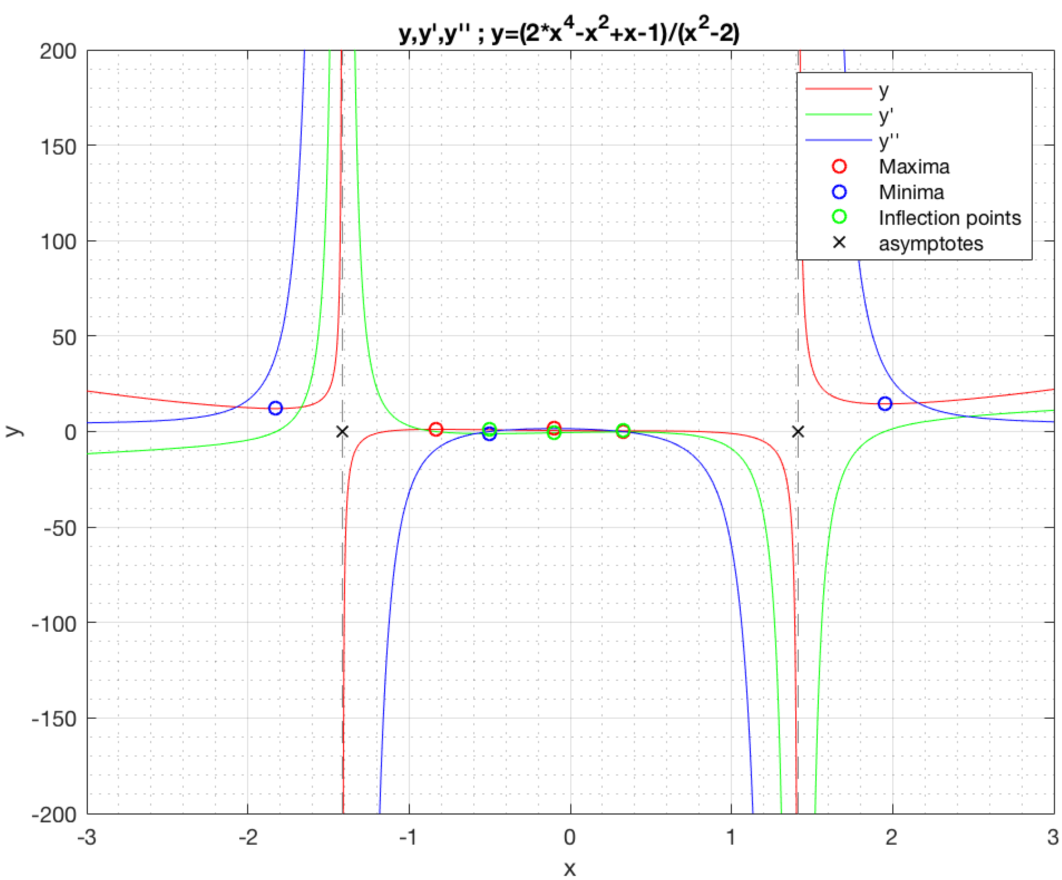
Inflection points at

x	y
-0.5035	0.9324
0.3259	0.4001
-19.4890	-77.9575
-0.1026	-0.6592
-24.3459	3.9999

Vertical asymptotes at x=

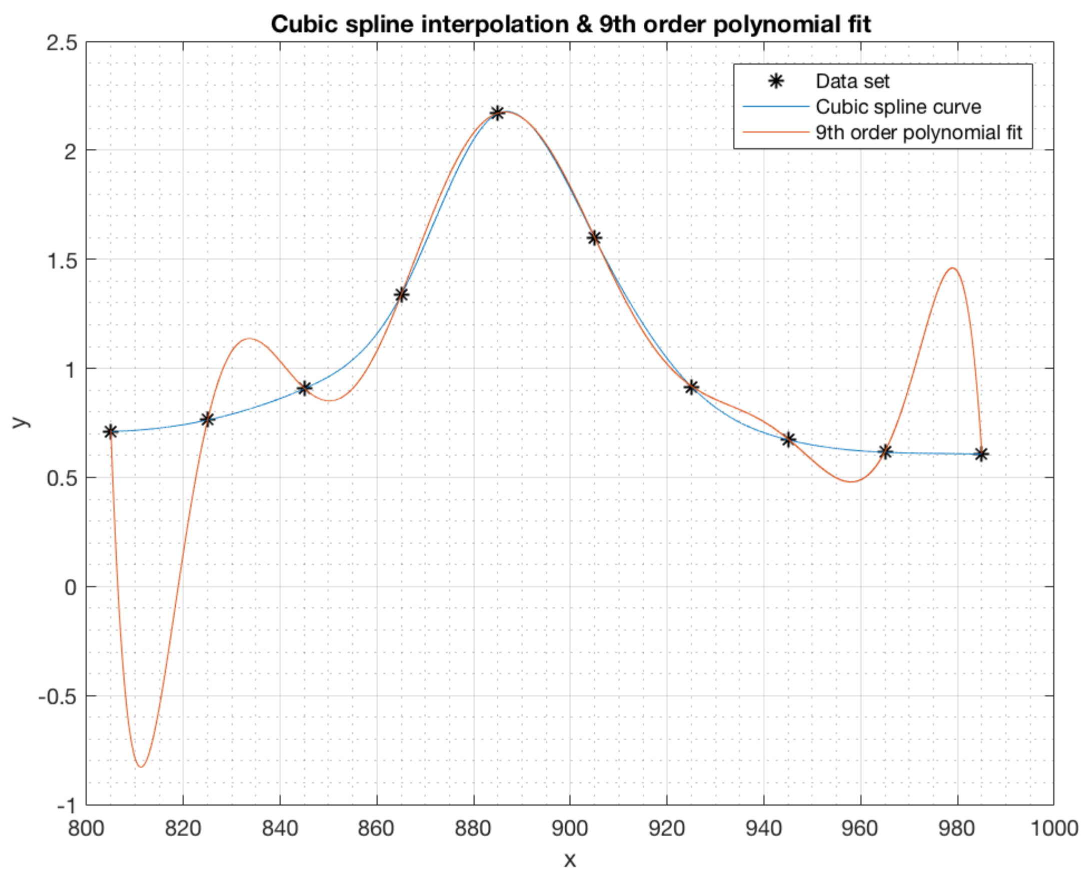
1.4142
1.4142
1.4142
-1.4142
-1.4142
-1.4142

%y, y' and y" have the same asymptotes



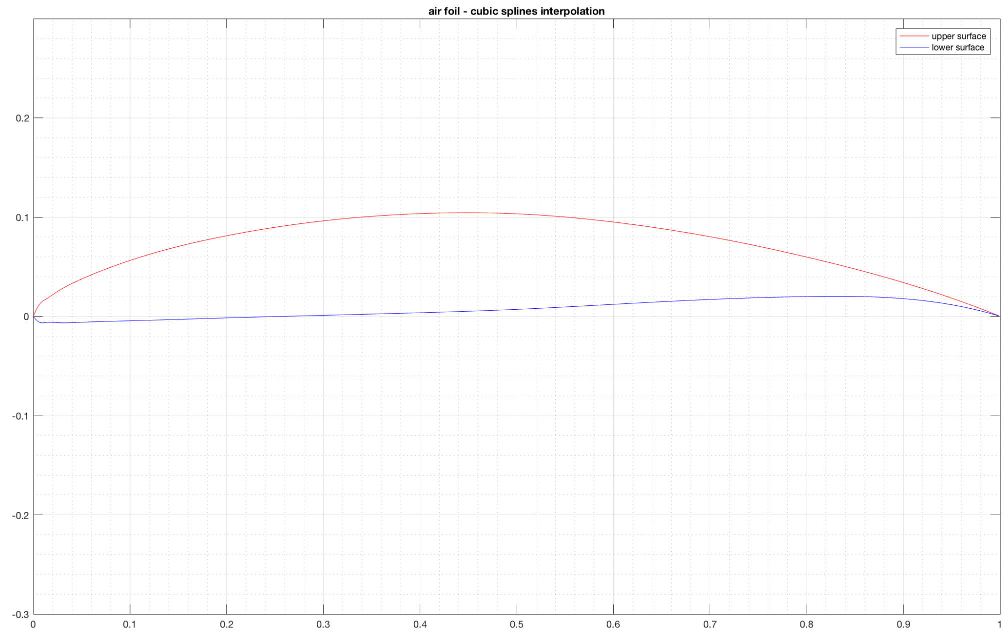
4.

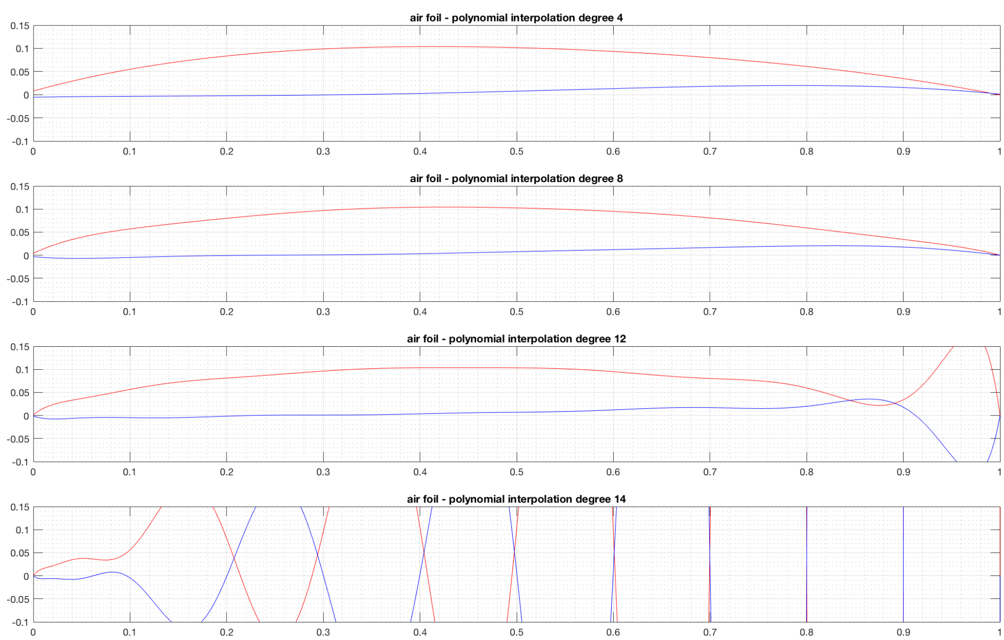
```
x=[805 825 845 865 885 905 925 945 965 985];
y=[0.710 0.763 0.907 1.336 2.169 1.598 0.916 0.672 0.615 0.606];

%cubic spline interpolation
xx=min(x):0.01:max(x);
yy=spline(x,y,xx);

%9th order polynomial fit
xxx=min(x):0.0000001:max(x);
PolyFit_9th=polyval(polyfit(x,y,9),xxx);

plot(x,y,'k*',xx,yy,xxx,PolyFit_9th)
grid on
grid minor
title('Cubic spline interpolation & 9th order polynomial fit')
xlabel('x')
ylabel('y')
legend('Data set','Cubic spline curve','9th order polynomial fit')
```



5.

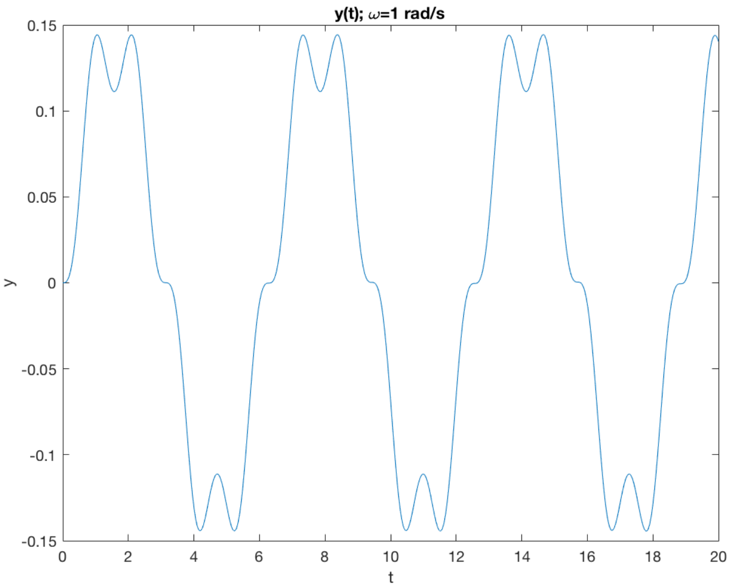
```
x=[0 0.005 0.0075 0.0125 0.025 0.05 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0]';
yu=[0 0.0102 0.0134 0.017 0.025 0.0376 0.0563 0.0812 0.0962 0.1035 0.1033 0.095 0.0802
0.0597 0.034 0]';
yl=[0 -0.0052 -0.0064 -0.0063 -0.0064 -0.006 -0.0045 -0.0016 0.001 0.0036 0.007 0.0121
0.017 0.0199 0.0178 0]';

%a)
xx=min(x):0.001:max(x);
yyu=spline(x,yu,xx);
yyl=spline(x,yl,xx);
plot(xx,yyu,'r',xx,yyl,'b')
ylim([-0.3,0.3])
grid on
grid minor
title('air foil - cubic splines interpolation')
legend('upper surface','lower surface')

%b)
figure
for i=1:4
  switch i
    case 1
      degree=4;
    case 2
      degree=8;
    case 3
      degree=12;
    case 4
      degree=14;
  end

  xxx=min(x):0.0000001:max(x);
  UpperPolyFit=polyval(polyfit(x,yu,degree),xxx);
  LowerPolyFit=polyval(polyfit(x,yl,degree),xxx);
  subplot(4,1,i)
  plot(xxx,UpperPolyFit,'r',xxx,LowerPolyFit,'b')
  ylim([-0.1,0.15])
  grid on
  grid minor
  title(['air foil - polynomial interpolation degree ' num2str(degree)])
end


%The polynomial interpolation gave reasonable curves for lower degree, but the shape
became unrealistic as the degree increased due to more number of inflection points the higher
degree polynomials have
```

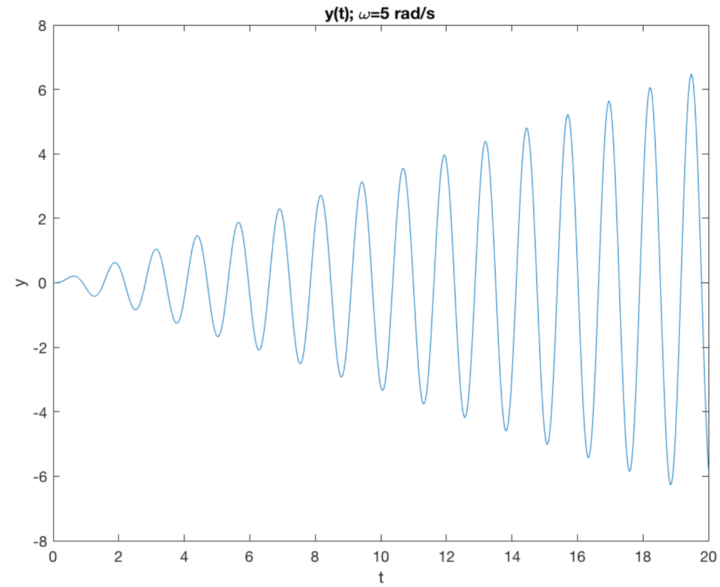

air foil - cubic splines interpolation
upper surface
lower surface

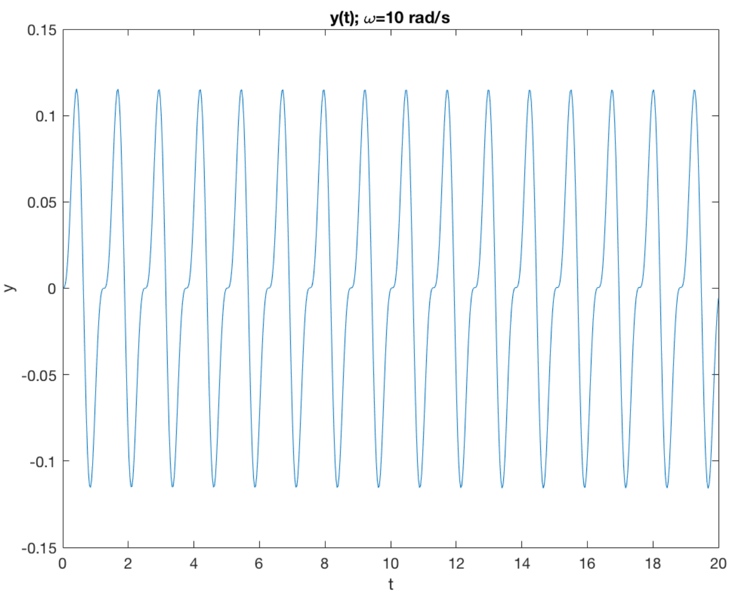

air foil - polynomial interpolation degree 4
air foil - polynomial interpolation degree 8
air foil - polynomial interpolation degree 12
air foil - polynomial interpolation degree 14

6.

```
tspan=[0,20]; %time scale

y0=0;
dy0=0;

[t, y] = ode45(@f,tspan,[y0 dy0]);
plot (t,y(:,1));
title('y(t); \omega=1 rad/s')
xlabel('t'); ylabel('y');

figure
[t, y] = ode45(@f2,tspan,[y0 dy0]);
plot (t,y(:,1));
title('y(t); \omega=5 rad/s')
xlabel('t'); ylabel('y');

figure
[t, y] = ode45(@f3,tspan,[y0 dy0]);
plot (t,y(:,1));
title('y(t); \omega=10 rad/s')
xlabel('t'); ylabel('y');

function dy=f(t, x)

    omega=1;
    dxdt=x(2);
    dx2dt=(1/3)*(10*sin(omega*t)-75*x(1));

    dy=[dxdt; dx2dt];

end

function dy=f2(t, x)

    omega=5;
    dxdt=x(2);
    dx2dt=(1/3)*(10*sin(omega*t)-75*x(1));

    dy=[dxdt; dx2dt];

end

function dy=f3(t, x)

    omega=10;
    dxdt=x(2);
    dx2dt=(1/3)*(10*sin(omega*t)-75*x(1));

    dy=[dxdt; dx2dt];

end
```







%The position of mass where omaga=5 seem to increase in the amplitude as time progresses, thus it is in resonance