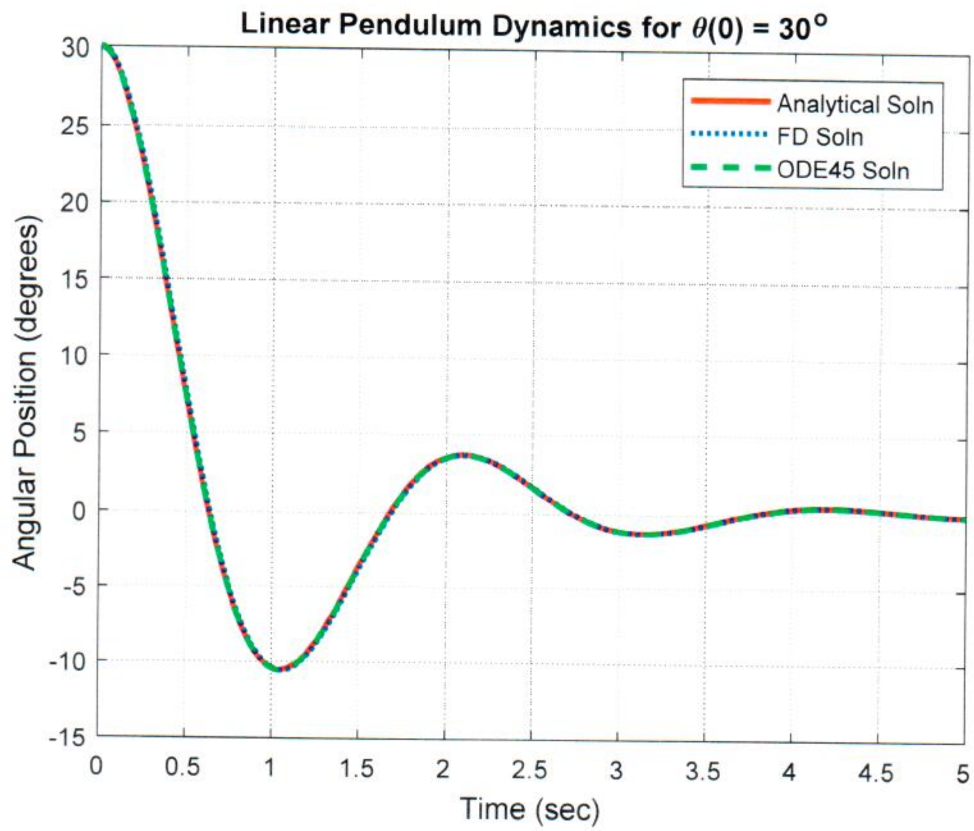
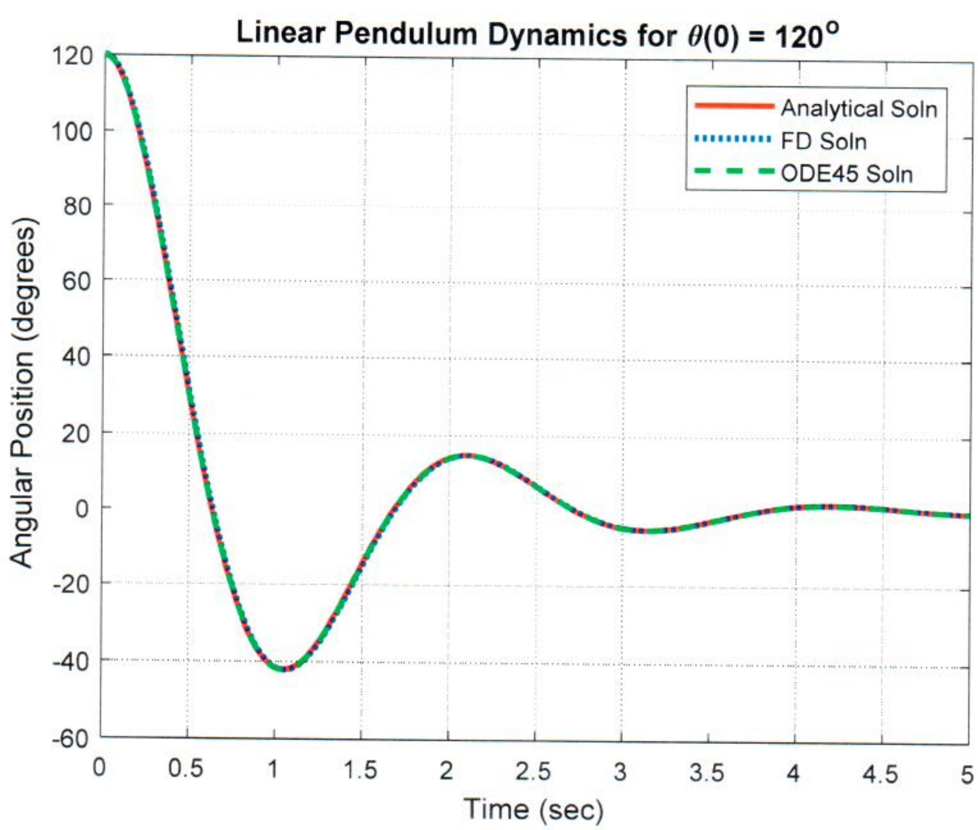
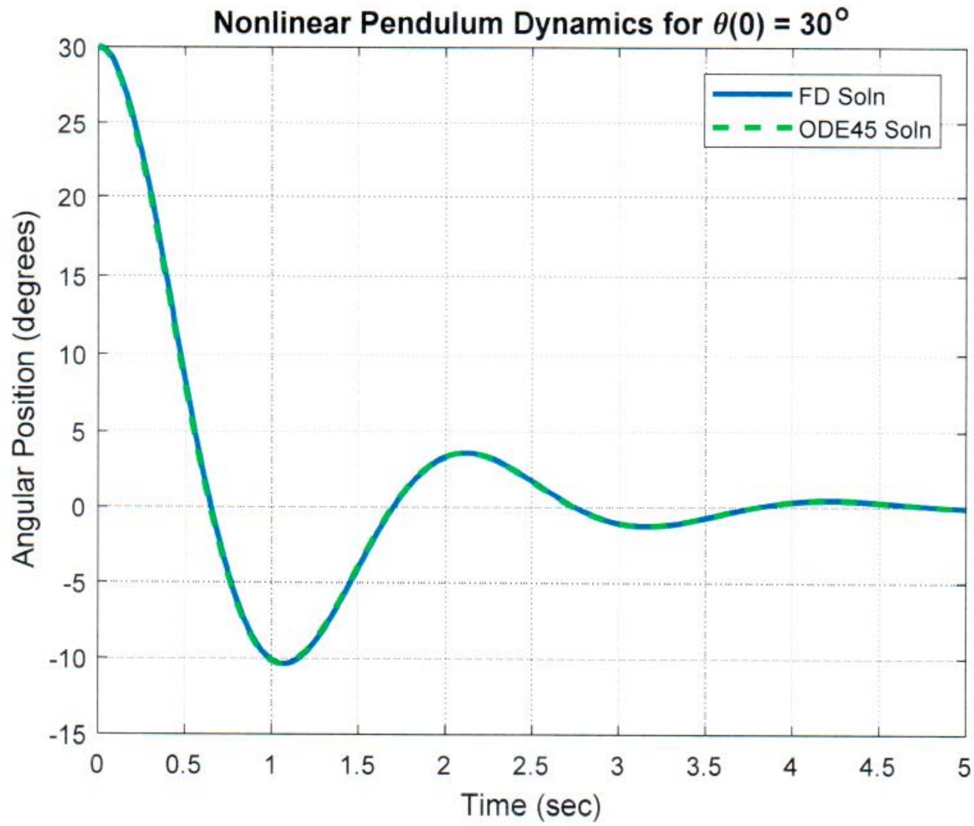
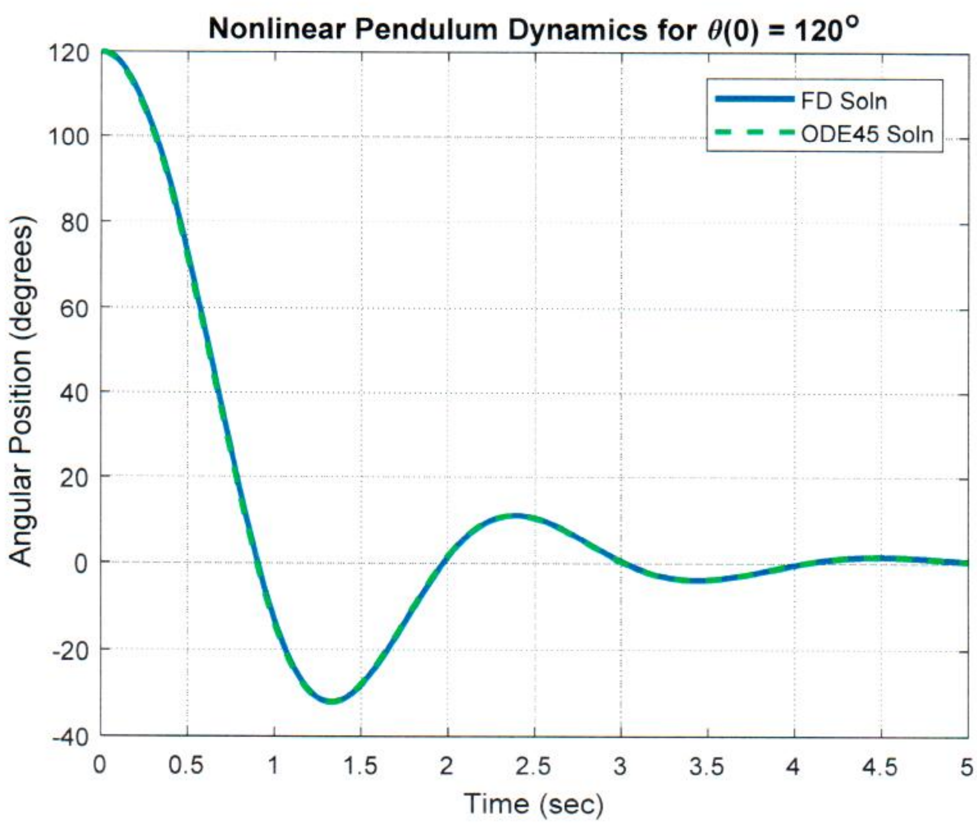
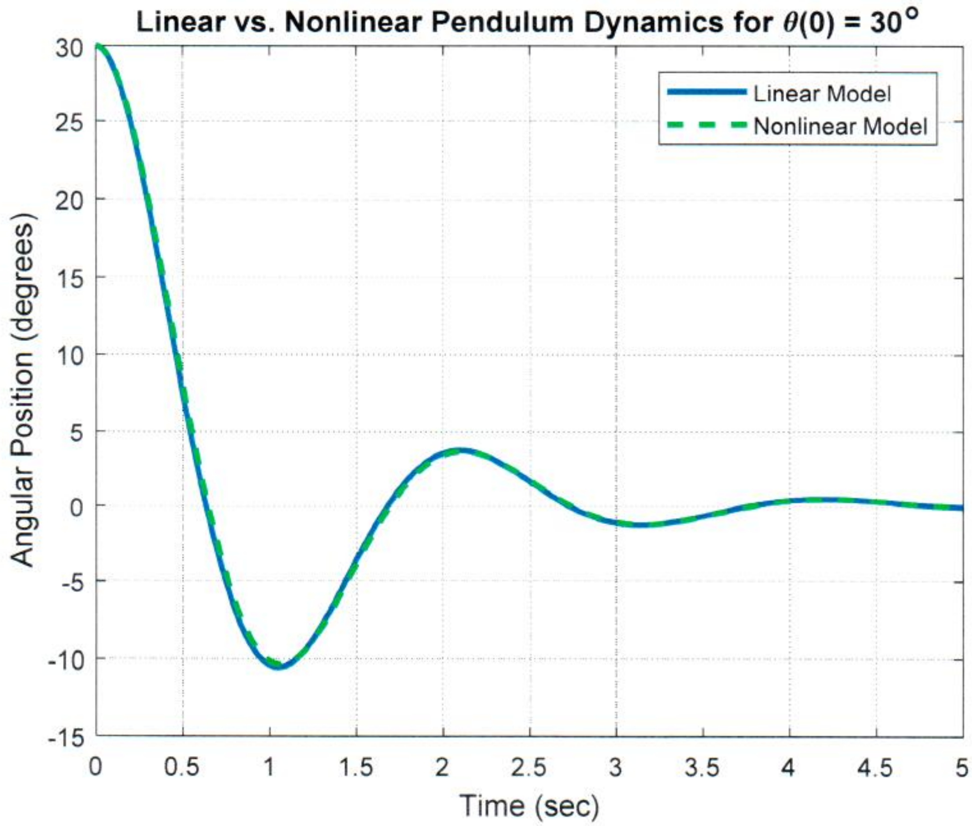
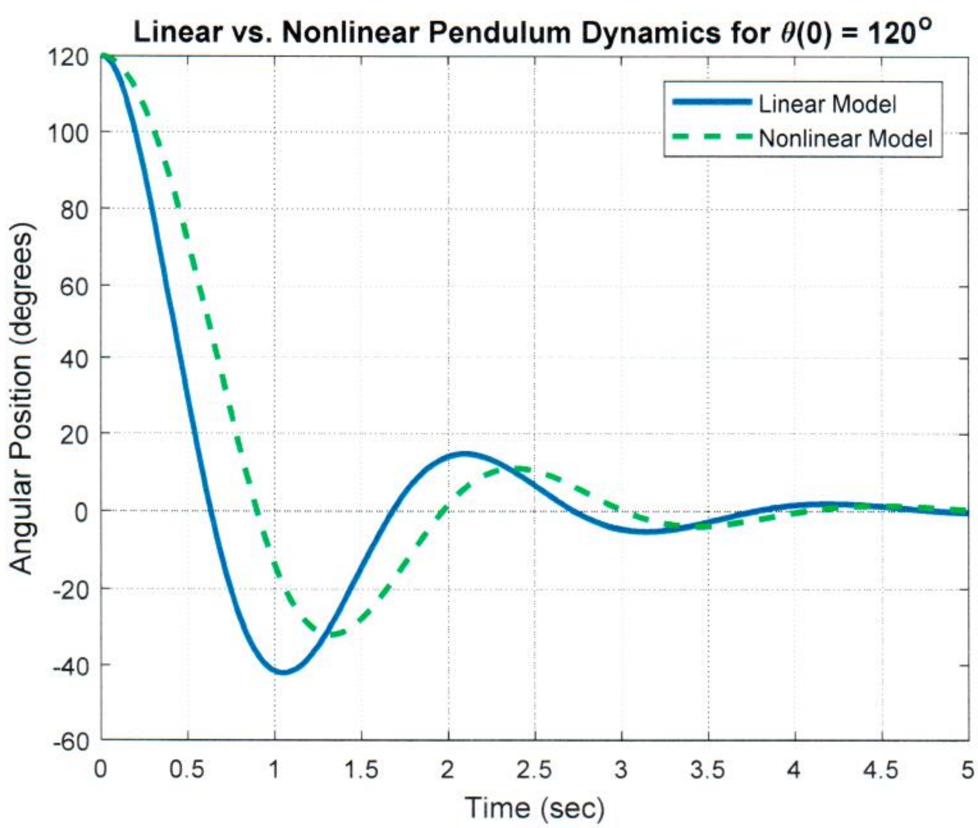

Linear Pendulum Dynamics for $\theta(0)$ = 30°
Analytical Soln
FD Soln
ODE45 Soln
Angular Position (degrees)
Time (sec)
30
25
20
15
10
5
0
-5
-10
-15
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5

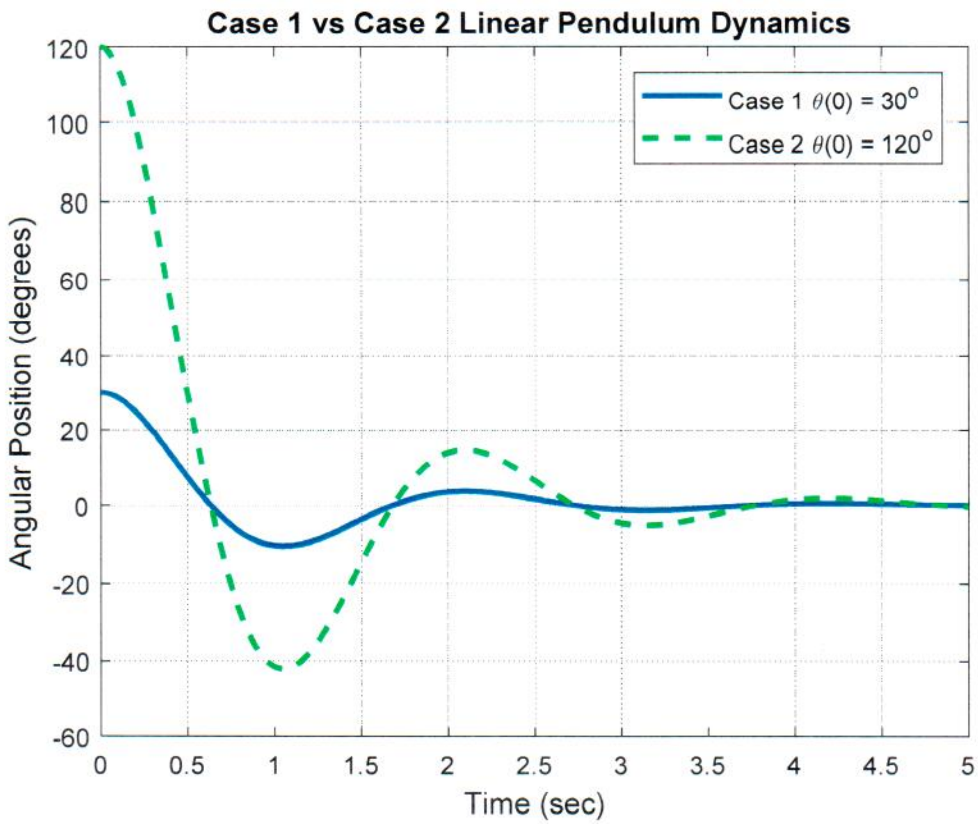

Linear Pendulum Dynamics for $\theta(0)$ = 120°
Analytical Soln
FD Soln
ODE45 Soln
Angular Position (degrees)
Time (sec)
120
100
80
60
40
20
0
-20
-40
-60
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5

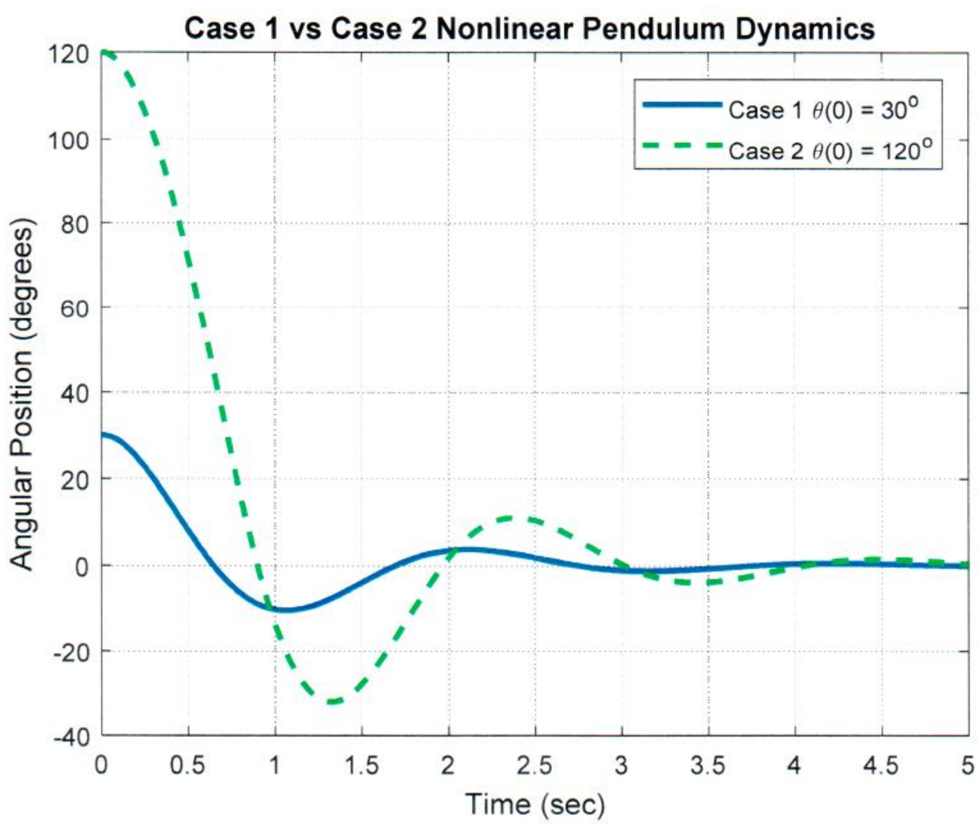

Nonlinear Pendulum Dynamics for $\theta(0) = 30^\circ$
FD Soln
ODE45 Soln
Angular Position (degrees)
Time (sec)
30
25
20
15
10
5
0
-5
-10
-15
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5


Nonlinear Pendulum Dynamics for $\theta(0) = 120^\circ$
FD Soln
ODE45 Soln
Angular Position (degrees)
Time (sec)
120
100
80
60
40
20
0
-20
-40
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5


Linear vs. Nonlinear Pendulum Dynamics for θ(0) = 30°
Linear Model
Nonlinear Model
Angular Position (degrees)
Time (sec)
30
25
20
15
10
5
0
-5
-10
-15
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5


Linear vs. Nonlinear Pendulum Dynamics for θ(0) = 120°
Linear Model
Nonlinear Model
Angular Position (degrees)
Time (sec)
120
100
80
60
40
20
0
-20
-40
-60
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5


Case 1 vs Case 2 Linear Pendulum Dynamics
Case 1 θ(0) = 30°
Case 2 θ(0) = 120°
Angular Position (degrees)
Time (sec)
120
100
80
60
40
20
0
-20
-40
-60
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5


Case 1 vs Case 2 Nonlinear Pendulum Dynamics
Case 1 θ(0) = 30°
Case 2 θ(0) = 120°
Angular Position (degrees)
Time (sec)
120
100
80
60
40
20
0
-20
-40
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5

```
%
%   PENDULUM_STUDY.M  Evaluate and plot the dynamics of both linear and nonlinear
%                         pendulum models via several different methods
%                   (analytical method, FD method, and use of ODE45 routine)
%
%   This example evaluates and plots the angular position of a pendulum with a point
%   mass at the end.  The focus is on the solution of IVPs using several methods,
%   (analytical for linear problems, and the finite difference (FD) method and the
%   use of ODE45 for both linear and nonlinear problems).
%   Thus, for the linear  model, all three solutions are compared and for the
%   nonlinear model, only the two numerical methods are studied.
%   The study also compares the behavior of the linear and nonlinear models for two
%   different sets of initial conditions (ICs).
%
%   A similar program, pendulum_1.m, focuses only on evaluating and plotting the
%   analytical solution.
%
%   File prepared by J. R. White, UMass-Lowell (last update: Nov. 2017)

      clear all;   close all;   nfig = 0;
%
%   define problem parameters
      m = 1;                      % pendulum mass (kg)
      c = 2;                      % friction coeff (kg/s)
      L = 1;                      % length of pendulum arm (m)
      g = 10;                     % gravitational acceleration (m/s^2)
      poso = [30 120]*pi/180;     % initial position (degrees converted to radians)
      NIC = length(poso);         % number of IC sets
      velo = 0;                   % initial velocity (radians/s)
%
%   define discrete time domain variable (with time in seconds)
      Nt = 401;   to = 0;   tf = 5;   t = linspace(to,tf,Nt)';
%
%       ******************************************
%       *  Method 1: LINEAR ANALYTICAL SOLUTION  *
%       ******************************************
%
      a = -c/(2*m);   b = sqrt(g/L - a^2);    % roots of characteristic equation
      pos_lin1 = zeros(Nt,NIC);
      for j = 1:NIC
        c1 = poso(j);      c2 = (velo - a*c1)/b;
        d1 = b*c2 + a*c1;   d2 = a*c2 - b*c1;
        pos_lin1(:,j) = exp(a*t).*(c1*cos(b*t) + c2*sin(b*t));
      end
      pos_lin1 = pos_lin1*180/pi;   % convert position into degrees
%
%       **********************************
%       *  Method 2: LINEAR FD SOLUTION  *
%       **********************************
%
      N = Nt-1;  dt = (tf-to)/N;   % dt is consistent with the time vector from above
      a = 1 + c*dt/(2*m);   b = 2 - g*dt*dt/L;   d = c*dt/(2*m) - 1;
      pos_lin2 = zeros(Nt,NIC);
      for j = 1:NIC
        pos_lin2(1,j) = poso(j);    pos_lin2(2,j) = poso(j) + velo*dt;
        for i = 2:N
```

```
      pos_lin2(i+1,j) = b*pos_lin2(i,j)/a + d*pos_lin2(i-1,j)/a;
    end
  end
  pos_lin2 = pos_lin2*180/pi;    % convert position into degrees
%
%     ************************************
%     *  Method 3: LINEAR ODE45 SOLUTION  *
%     ************************************
%
  pos_lin3 = zeros(Nt,NIC);
  ftx = @(t,x) [x(2); -g*x(1)/L - c*x(2)/m];  % anonymous function for ode45
  for j = 1:NIC
    xo = [poso(j) velo]';
    [t,x] = ode45(ftx,t,xo);
    pos_lin3(:,j) = x(:,1);      % stores position vs time for case j
   end
  pos_lin3 = pos_lin3*180/pi;    % convert position into degrees
%
%     **************************
%     *  PLOT LINEAR SOLUTIONS  *
%     **************************
%   Case 1 ICs  --  30 degrees
  nfig = nfig+1;   figure(nfig)
  plot(t,pos_lin1(:,1),'r-',t,pos_lin2(:,1),'b:',t,pos_lin3(:,1),'g--', ...
       'LineWidth',2),grid
  title('Linear Pendulum Dynamics for \theta(0) = 30^o')
  xlabel('Time (sec)'),ylabel('Angular Position (degrees)')
  legend('Analytical Soln','FD Soln','ODE45 Soln')
%   Case 2 ICs  --  120 degrees
  nfig = nfig+1;   figure(nfig)
  plot(t,pos_lin1(:,2),'r-',t,pos_lin2(:,2),'b:',t,pos_lin3(:,2),'g--', ...
       'LineWidth',2),grid
  title('Linear Pendulum Dynamics for \theta(0) = 120^o')
  xlabel('Time (sec)'),ylabel('Angular Position (degrees)')
  legend('Analytical Soln','FD Soln','ODE45 Soln')
%
%     ************************************
%     *  Method 2: NONLINEAR FD SOLUTION  *
%     ************************************
%
  N = Nt-1;  dt = (tf-to)/N;   % dt is consistent with the time vector from above
  a = 1 + c*dt/(2*m);    b = -g*dt*dt/L;    d = c*dt/(2*m) - 1;
  pos_nlin2 = zeros(Nt,NIC);
  for j = 1:NIC
    pos_nlin2(1,j) = poso(j);    pos_nlin2(2,j) = poso(j) + velo*dt;
    for i = 2:N
      pos_nlin2(i+1,j) = 2*pos_nlin2(i,j)/a + b*sin(pos_nlin2(i,j))/a + ...
                         d*pos_nlin2(i-1,j)/a;
    end
  end
  pos_nlin2 = pos_nlin2*180/pi;    % convert position into degrees
%
%     ************************************
%     *  Method 3: NONLINEAR ODE45 SOLUTION  *
%     ************************************
%
```

```
   pos_nlin3 = zeros(Nt,NIC);
   ftx = @(t,x) [x(2); -g*sin(x(1))/L - c*x(2)/m];  % anonymous function for ode45
   for j = 1:NIC
     xo = [poso(j) velo]';
     [t,x] = ode45(ftx,t,xo);
     pos_nlin3(:,j) = x(:,1);       % stores position vs time for case j
    end
   pos_nlin3 = pos_nlin3*180/pi;    % convert position into degrees
%
%     **************************
%     *  PLOT NONLINEAR SOLUTIONS  *
%     **************************
%  Case 1 ICs  --  30 degrees
   nfig = nfig+1;   figure(nfig)
   plot(t,pos_nlin2(:,1),'b-',t,pos_nlin3(:,1),'g--', ...
        'LineWidth',2),grid
   title('Nonlinear Pendulum Dynamics for \theta(0) = 30^o')
   xlabel('Time (sec)'),ylabel('Angular Position (degrees)')
   legend('FD Soln','ODE45 Soln')
%  Case 2 ICs  --  120 degrees
   nfig = nfig+1;   figure(nfig)
   plot(t,pos_nlin2(:,2),'b-',t,pos_nlin3(:,2),'g--', ...
        'LineWidth',2),grid
   title('Nonlinear Pendulum Dynamics for \theta(0) = 120^o')
   xlabel('Time (sec)'),ylabel('Angular Position (degrees)')
   legend('FD Soln','ODE45 Soln')
%
%     ****************************************
%     *  PLOT LINEAR & NONLINEAR SOLUTIONS  *
%     ****************************************
%  Note: since all the linear and all the nonlinear solns are identical, let's
%        pick just one for comparison purposes --  I chose the ODE45 solns
%  Case 1 ICs  --  30 degrees
   nfig = nfig+1;   figure(nfig)
   plot(t,pos_lin1(:,1),'b-',t,pos_nlin3(:,1),'g--', ...
        'LineWidth',2),grid
   title('Linear vs. Nonlinear Pendulum Dynamics for \theta(0) = 30^o')
   xlabel('Time (sec)'),ylabel('Angular Position (degrees)')
   legend('Linear Model','Nonlinear Model')
%  Case 2 ICs  --  120 degrees
   nfig = nfig+1;   figure(nfig)
   plot(t,pos_lin3(:,2),'b-',t,pos_nlin3(:,2),'g--', ...
        'LineWidth',2),grid
   title('Linear vs. Nonlinear Pendulum Dynamics for \theta(0) = 120^o')
   xlabel('Time (sec)'),ylabel('Angular Position (degrees)')
   legend('Linear Model','Nonlinear Model')
%
%     ****************************************
%     *  PLOT Case 1 vs Case 2 LINEAR SOLUTIONS  *
%     ****************************************
   nfig = nfig+1;   figure(nfig)
   plot(t,pos_lin3(:,1),'b-',t,pos_lin3(:,2),'g--', ...
        'LineWidth',2),grid
   title('Case 1 vs Case 2 Linear Pendulum Dynamics ')
   xlabel('Time (sec)'),ylabel('Angular Position (degrees)')
   legend('Case 1 \theta(0) = 30^o','Case 2 \theta(0) = 120^o')
```

```
%
%       **************************************
%       *  PLOT Case 1 vs Case 2 NONLINEAR SOLUTIONS  *
%       **************************************
     nfig = nfig+1;   figure(nfig)
     plot(t,pos_nlin3(:,1),'b-',t,pos_nlin3(:,2),'g--', ...
          'LineWidth',2),grid
     title('Case 1 vs Case 2 Nonlinear Pendulum Dynamics ')
     xlabel('Time (sec)'),ylabel('Angular Position (degrees)')
     legend('Case 1 \theta(0) = 30^o','Case 2 \theta(0) = 120^o')
%
%   end of problem
```