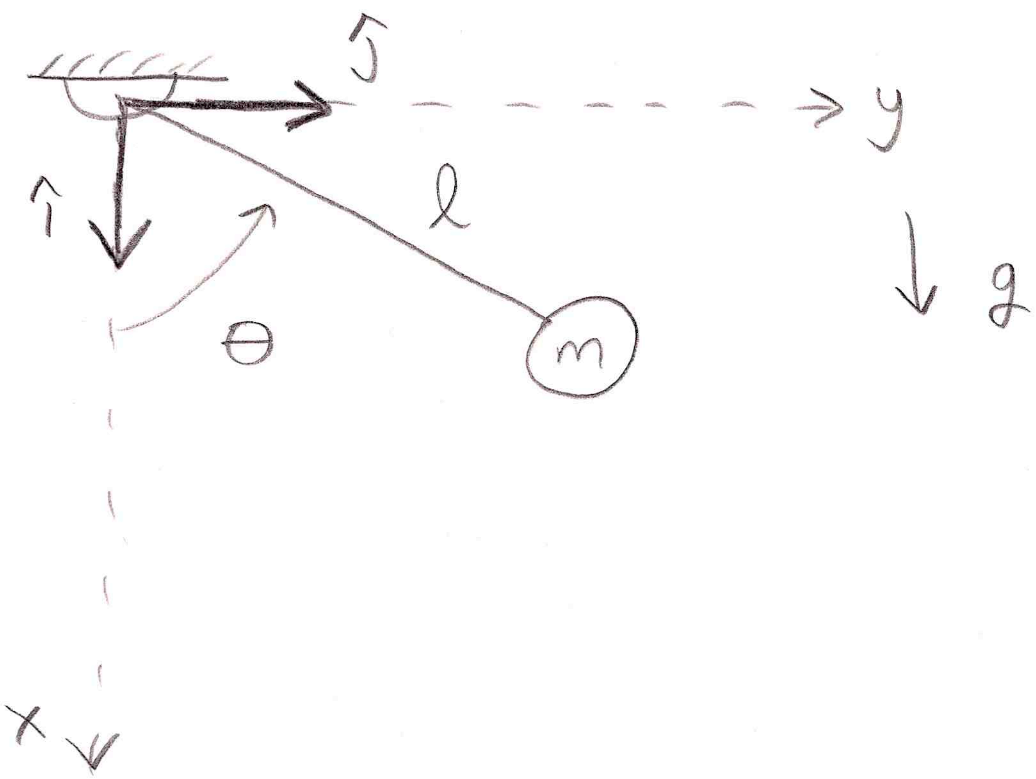
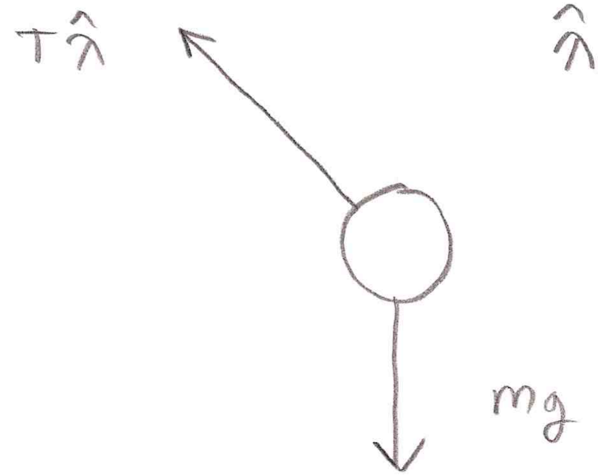
12) Set up the pendulum in cartesian coordinates. Express the constant length constraint as a set of linear equations restricting the acceleration.

We have



12) (continued)

FBD



$\hat{\lambda}$ in direction of string

LMB

$$mg\hat{\imath} + T\hat{\lambda} = m\vec{a}$$

$$\hat{\lambda} = \frac{-x\hat{\imath} - y\hat{\jmath}}{\sqrt{x^2+y^2}} \quad \text{and} \quad \vec{a} = \ddot{x}\hat{\imath} + \ddot{y}\hat{\jmath}$$

$$\Rightarrow \left(mg - \frac{Tx}{\sqrt{x^2+y^2}}\right)\hat{\imath} - \left(\frac{Ty}{\sqrt{x^2+y^2}}\right)\hat{\jmath} = m(\ddot{x}\hat{\imath} + \ddot{y}\hat{\jmath})$$

12) (continued)

Dotting the LMB equation with $\hat{\imath}$ and $\hat{\jmath}$ gives

$$\{\}\cdot\hat{\imath} \Rightarrow mg - \frac{Tx}{\sqrt{x^2+y^2}} = m\ddot{x}$$ ✓

$$\{\}\cdot\hat{\jmath} \Rightarrow -\frac{Ty}{\sqrt{x^2+y^2}} = m\ddot{y}$$ ✓

Constraint Equation

$$x^2+y^2=l^2$$ ✓

Differentiating this equation twice with respect to time gives

$$\frac{d}{dt}\left(x^2+y^2=l^2\right) \Rightarrow 2x\dot{x}+2y\dot{y}=0$$

$$\frac{d^2}{dt^2}\left(x^2+y^2=l^2\right) \Rightarrow 2\dot{x}^2+2x\ddot{x}+2\dot{y}^2+2y\ddot{y}=0$$

12) (continued)

we thus have the following three equations:

$$m\ddot{x} + \left(\frac{x}{\sqrt{x^2+y^2}}\right) T = mg$$

$$m\ddot{y} + \left(\frac{y}{\sqrt{x^2+y^2}}\right) T = 0$$

$$x\ddot{x} + y\ddot{y} = -(\dot{x}^2 + \dot{y}^2)$$

$$\begin{bmatrix} m & 0 & x/\ell \\ 0 & m & y/\ell \\ x & y & 0 \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ T \end{bmatrix} = \begin{bmatrix} mg \\ 0 \\ -(\dot{x}^2 + \dot{y}^2) \end{bmatrix}$$ ✓

See attached Matlab code which solves these
DAEs as well as plots comparing this solution
with that of the simple pendulum equation ✳

12) (continued)

Initial Conditions and Givens for Matlab code

Let $m = 1 kg$

$l = 1 m$

$g = 1 m/s^2$

$$\begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$ } Pendulum initially hanging vertically

$$\begin{bmatrix} \dot{x}_0 \\ \dot{y}_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

For simple pendulum equation, we have that

$\begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \Rightarrow \underline{\theta_0 = 0 \text{ radians}}$

12) (continued)

we can also get an initial condition for $\dot{\theta}_0$:

$$y = l \sin\theta$$

$$\Rightarrow \quad \dot{y} = l \dot{\theta} \cos\theta$$

$$\dot{\theta} = \frac{\dot{y}}{l \cos\theta}$$

Thus,

$$\dot{\theta}_0 = \frac{\dot{y}_0}{l \cos\theta_0}$$

For $\dot{y}_0 = 1$, $l = 1$, and $\theta_0 = 0$ we get

$$\dot{\theta}_0 = \frac{(1)}{(1)\cos(0)} \quad \Rightarrow \quad \underline{\dot{\theta}_0 = 1}$$

```
%Homework 5, Problem 12

function MAE5735_Pendulum()
%This function calculates the motion of a simple pendulum using 3 second-order
%DAEs as well as by using the simple pendulum equation. These two methods
%are then compared.

clc
clear all
close all

%The following are pertinent constants for this scenario.
p.m = 1; %Mass of the pendulum in kg
p.L = 1; %Length of the pendulum in m
p.g = 1; %Gravitational acceleration in m/s^2

%The following establishes the time span and initial conditions for the
%ODE solution.

tspan = linspace(0,100,10001); %Timespan under consideration
x0 = [1 0]'; %Initial positon vector.
v0 = [0 1]'; %Initial velocity vector. Initially the speeds of the two
             %masses are zero.
z0_DAE = [x0;v0]; %This vector contains the position and velocity information
              %of the pendulum to be solved by the ODE solver.

%Initial conditions for simple pendulum equation
theta0 = 0; %Theta starts are zero radians, i.e., the pendulum is hanging straight
down
theta_dot0 = 1; %Initial angular velocity is 1 rad/s
z0_theta = [theta0;theta_dot0]; %Vector containing intial angle and angular velocity
information

%The following solves the differential equation outlined in the subfunction
%"zdot_DAE" below as well as sets the tolerances for the ODE solver
options = odeset('reltol',1e-13,'abstol',1e-13);
[t zarray] = ode23(@rhs_DAE,tspan,z0_DAE,options,p);

%Isolate the positions of m1 and m2 to facilitate plotting of the motion.
x = zarray(:,1); %x-position of pendulum
y = zarray(:,2); %y-position of pendulum

%The following solves the differential equation outlined in the subfunction
%"zdot_theta" below for the simple pendulum equation
[t z_theta] = ode23(@rhs_theta,tspan,z0_theta,[],p);

%Isolate the angle of the pendulum.
theta = z_theta(:,1);
```

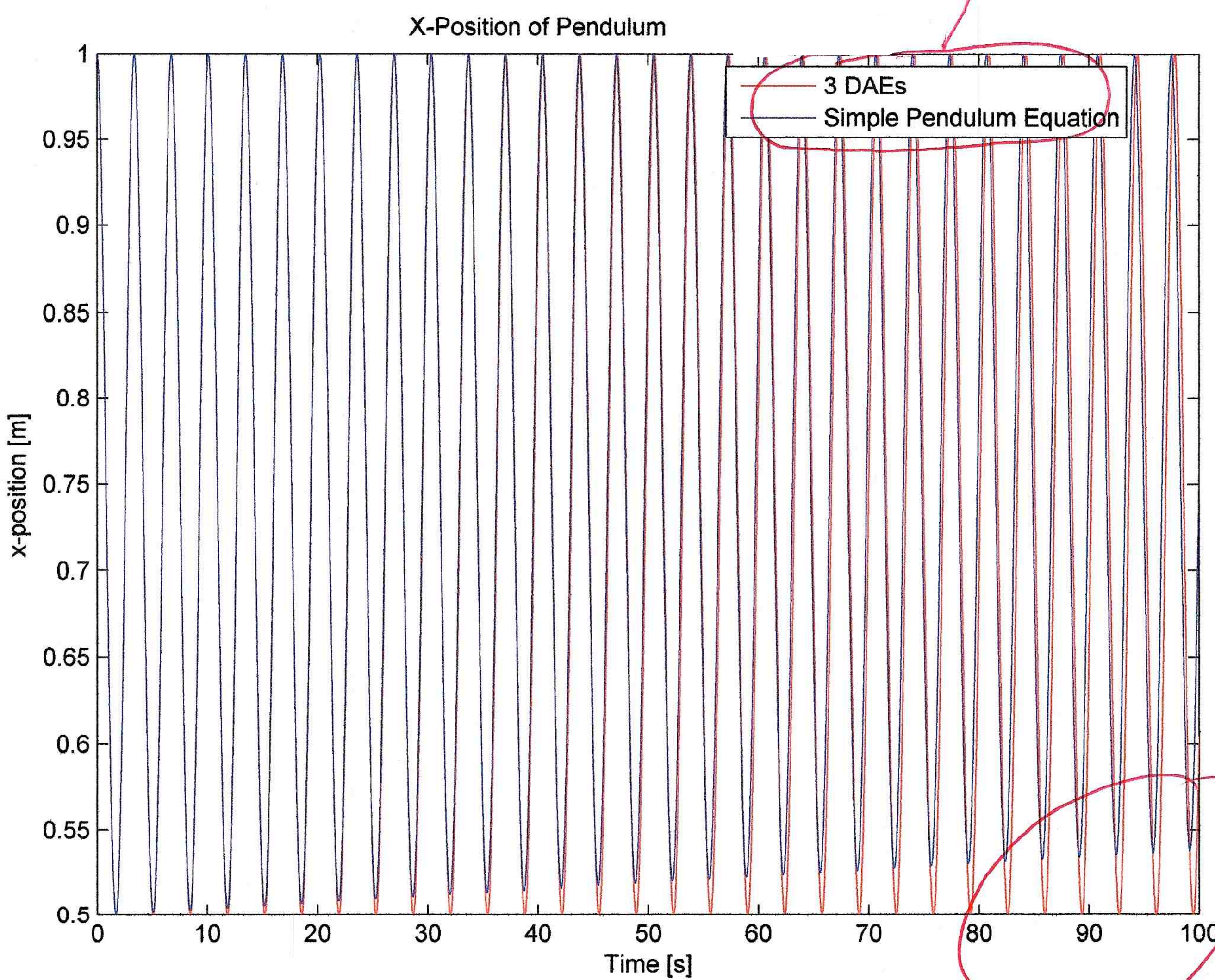
7

```
%We can now compute the x-position and y-position from the angle information since
%x = l*cos(theta) and y = l*sin(theta)
x_simplependulum = (p.L).*cos(theta);
y_simplependulum = (p.L).*sin(theta);

%The following plots the motion of the pendulum as well as the error
%assosicated with the constraint
plot(t,x,'r',t,x_simplependulum,'b')
xlabel('Time [s]')
ylabel('x-position [m]')
title('X-Position of Pendulum                   ')
legend('3 DAEs','Simple Pendulum Equation')
figure

plot(t,y,'r',t,y_simplependulum,'b')
xlabel('Time [s]')
ylabel('y-position [m]')
title('Y-Position of Pendulum                   ')
legend('3 DAEs','Simple Pendulum Equation')
figure

%The following shows how close the solution was to satisfying the kinematic
%constraint
plot(t,sqrt((x.^2+y.^2))-p.L,'r')
xlabel('Time [s]')
ylabel('sqrt((x^2+y^2))-l')
title('How close the 3DAE solution satisfies the kinematic constant length
constraint')

figure
%The following plots the difference between the simple pendulum equation
%solution and the solution using the 3 DAEs
plot(t,x_simplependulum-x,'b',t,y_simplependulum-y,'r')
xlabel('Time [s]')
ylabel('x/y_s_i_m_p_l_e_ _p_e_n_d_u_l_u_m - x/y')
title('Difference between simple pendulum equation and the 3DAE solution
      ')
legend('x_s_i_m_p_l_e_ _p_e_n_d_u_l_u_m - x [m]','y_s_i_m_p_l_e_ _p_e_n_d_u_l_u_m - y
[m]')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%THE FOLLOWING IS A SUBFUNCTION UTILIZED BY THIS CODE%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function zdot_DAE =rhs_DAE(t,z,p)
%This function calculates the time derivative of a given input vector.
x=z(1:2); %The position vector containing x and y.
v=z(3:4); %The velocity vector containing xdot and ydot.
```

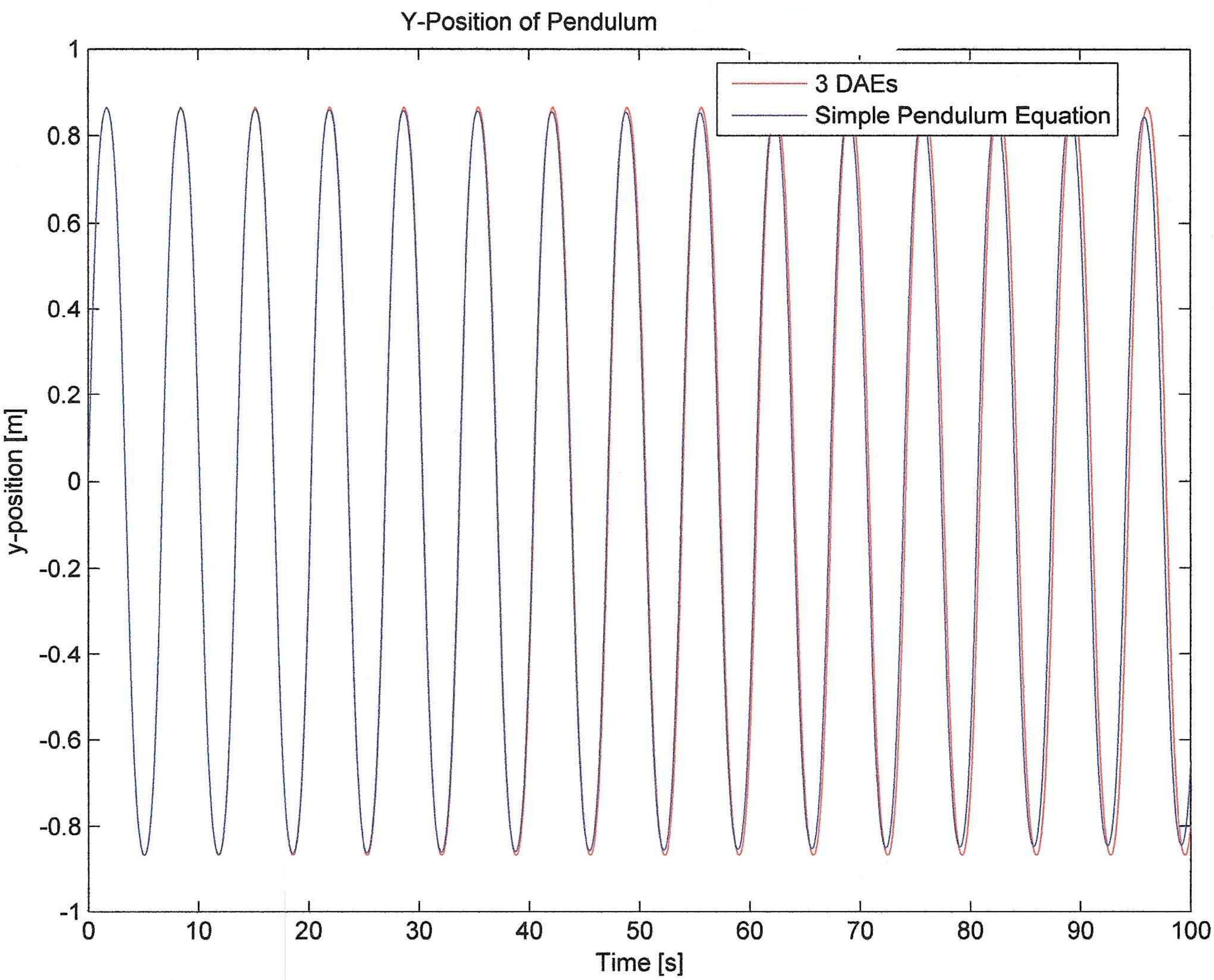
8

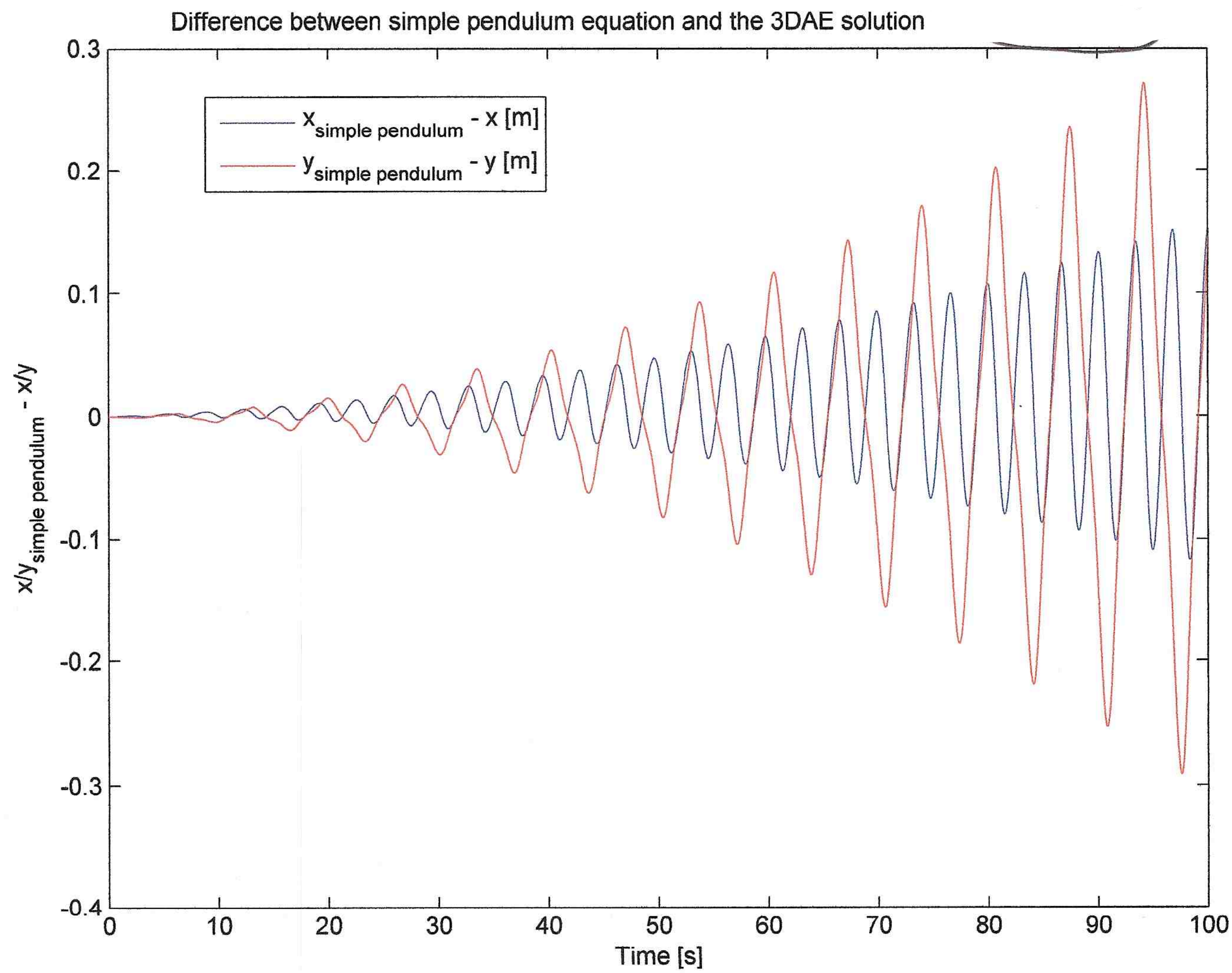
```
N = [p.m 0 z(1)/p.L;0 p.m z(2)/p.L;z(1) z(2) 0]; %The matrix on the left side of the
DAE matrix equation
b = [p.m*p.g;0;-(v(1)^2+v(2)^2)]; %Right hand side of the DAE matrix equation

solution =N\b; %The accelerations and the constraint tension

xdot = v; %Says that the time derivatives of the positions are just the respective
velocities.
vdot = solution(1:2); %Extracts the appropriate accelerations

zdot_DAE = [xdot;vdot]; %Collects the important parameters, i.e., the time derivatives
                        %of the input quantities.
end

function zdot_theta =rhs_theta(t,z,p)
%This function calculates the time derivative of a given input vector.
theta = z(1); %The angle of the pendulum
theta_dot = z(2); %The angular velocity of the pendulum

%The following evaluates the second derivative of theta based on the simple
%pendulum equation
theta_doubledot = (-(p.g)/(p.L))*sin(theta);

zdot_theta = [theta_dot;theta_doubledot]; %Collects the important parameters, i.e.,
the time derivatives
                                          %of the input quantities.
end

end
```

mislabeled



one would think DAEs should be worse.

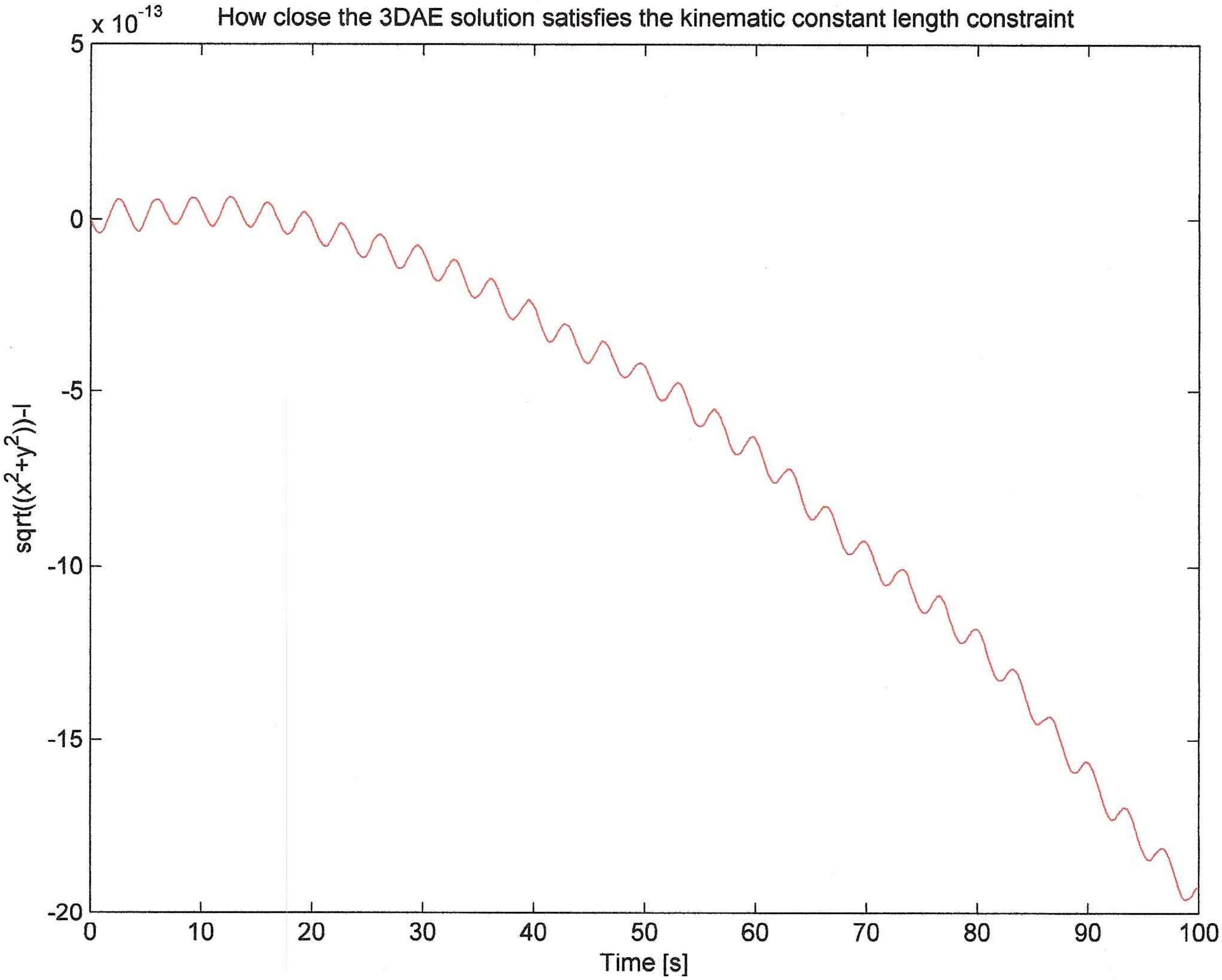

Y-Position of Pendulum
3 DAEs
Simple Pendulum Equation
y-position [m]
Time [s]
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
10
20
30
40
50
60
70
80
90
100



Difference between simple pendulum equation and the 3DAE solution
x/y simple pendulum - x/y
Time [s]
x simple pendulum - x [m]
y simple pendulum - y [m]
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
0
10
20
30
40
50
60
70
80
90
100

12) (continued)

As seen in the previous plots, the solution computed using the 3 DAEs is close to the result from the simple pendulum equation, but only for small time spans. As the time increases, however, the errors in the x-position and y-position both increase. The errors in the y-position seem to increase more rapidly than those in the x-position, however. The following plot shows the drift away from satisfying the kinematic constraint of a constant length pendulum.


How close the 3DAE solution satisfies the kinematic constant length constraint
x 10^-13
5
0
-5
-10
-15
-20
sqrt((x^2+y^2))-l
0
10
20
30
40
50
60
70
80
90
100
Time [s]

12) (continued)

Additionally, the previous plot shows the drift away from satisfying the kinematic constant length constraint, namely,

$$x^2 + y^2 = l^2$$

As the time span of the calculation is increased, the value of $\sqrt{x^2+y^2}$ slowly becomes smaller than l, which is shown by the value $\sqrt{x^2+y^2} - l$ becoming increasingly negative as the time span increases. The difference is on the order of 10^{-13} m; however, it is still present.

Can also be fixed by "projection"