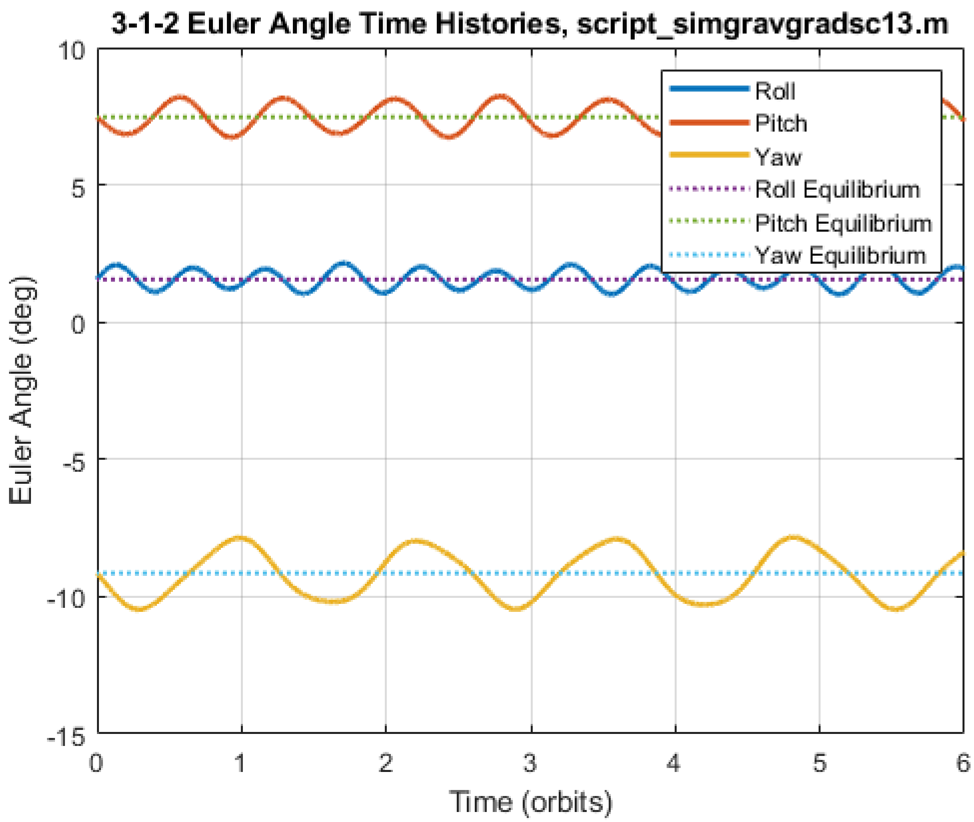
Problem 3

```
%script_simgravgradsc13.m
%
%  Copyright (c) 2019 Mark L. Psiaki.  All rights reserved.
%
%  This Matlab script simulates the motion of a
%  nadir-pointing satellite that is acting
%  under the influence of a gravity-gradient torque
%  in a circular Low-Earth Orbit (LEO) that has an
%  orbital period of 6000 sec.  The initial conditions
%  start with zero roll, pitch, and yaw angles relative
%  to the principal axes, but with non-zero initial roll,
%  pitch, and yaw rates relative to the princpal axes.
%
%  The principal axes are not exactly body axes, which is
%  why the point about which the satellite attitude
%  oscillates is not exactly zero for the roll, pitch, and
%  yaw angles.
%
%  This script also makes plots of the roll, pitch, and
%  yaw attitude time history.
%
%  Clear the Matlab workspace.
%
   clear;clc;close all;
%
%  Load simulation parameters IMoIbody, norbit, omegabody0
%  and q0 from the file simgravgradsc13_data.mat.
%
   load simgravgradsc13_data
%
%  Set up the initial state.  This initial state vector
%  has been chosen with a knowledge of the principal
%  axes coordinates, and q0 has been chosen so that
%  the principal axes are exactly aligned with
%  the local-level orbit-following coordinate
%  system so that, had omegabody0 also been chosen
%  correctly, q(t) and omegabody(t) would have
%  remainded constant.  A slightly perturbed
%  omegabody0 has been chosen to produce motion.
%  This allows the stability of the gravity-gradient
%  system to be tested.
%
   x0 = [q0;omegabody0];
%
%  Define the aircraft dynamics function handle
%  in a form that is suitable for input to ode45.m.
%
   ffunctode45 = @(tdum,xdum) ...
             ffunctgravgradsc02(tdum,xdum,IMoIbody,norbit);
%
```

```
%  Define the time span of the simulation, computing outputs
%  1000 times per orbit for 6 orbits.
%
   Torbit = 2*pi/norbit;
   tspan = (0:6000)'*(Torbit/1000);
%
%  Set up numerical integration options for ode45.m
%  in a way that uses a tighter relative tolerance than
%  is normally used.
%
   optionsode45 = odeset('RelTol',1.e-10);
%
%  Call ode45.m in order to perform numerical integration.
%
   tic
   [thist,xhist] = ode45(ffunctode45,tspan,x0,optionsode45);
   timetosim = toc;
%
%  Determine the 3-1-2 Euler angle time histories.  The function
%  yawpitchrollcalc02.m has been designed with the 3-1-2
%  assumption specifically in mind.  Note that the
%  computation of qknorm via normalization is included
%  in order to remove any small errors in the quaternion
%  unit normalization that may have built up due
%  to numerical intergration error of the quaternion
%  kinematics.
%
   N = size(thist,1);
   phihist = zeros(N,1);
   thetahist = zeros(N,1);
   psihist = zeros(N,1);
   for k = 1:N
      qk = xhist(k,1:4)';
      qknorm = qk*(1/sqrt(sum(qk.^2)));
      [phik,thetak,psik] = yawpitchrollcalc02(qknorm);
      phihist(k,1) = phik;
      thetahist(k,1) = thetak;
      psihist(k,1) = psik;
   end
   clear k qk qknorm phik thetak psik
%
%  Plot the roll, pitch, and yaw time histories.
%  An analysis of this case indicates that there should
%  be 8.05 pitch angle oscillations, 4.61 oscillations
%  of one of the coupled roll-yaw modes, and 11.4
%  oscillations of the other coupled roll-yaw modes.
%  Of course, the actual roll, pitch, and yaw
%  principal axes differ slightly from the
%  body axes used in this simulation.  Therefore,
%  the roll, pitch, and yaw angle time histories plotted
%  below are not pure principal axes quantities.
%  This fact causes slight additional coupling
%  between modes beyond the theoretical coupling of the
%  two roll/yaw modes of oscillation.
```

```
%
   figure(1)
   hold off
   plot(thist*(1/Torbit),[phihist,thetahist,psihist]*(180/pi),...
        'LineWidth',2)
   hold on
   plot(thist*(1/Torbit),ones(N,1)*...
        ([phihist(1,1),thetahist(1,1),psihist(1,1)]*(180/pi)),...
        ':','LineWidth',1.5)
   hold off
   set(get(gcf,'CurrentAxes'),'FontSize',10)
   grid
   xlabel('Time (orbits)')
   ylabel('Euler Angle (deg)')
   title(['3-1-2 Euler Angle Time Histories,',...
          ' script\_simgravgradsc13.m'])
   legend('Roll','Pitch','Yaw',...
          'Roll Equilibrium','Pitch Equilibrium',...
          'Yaw Equilibrium')
%
%  Save the results.
%
   textcommands = ['These data have been generated by the',...
                   ' commands in script_simgravgradsc13.m'];
   save simgravgradsc13
   format long
   xfinal = xhist(end,:)'
   qfinalmag = norm(xfinal(1:4,1))

   disp('The system produces a neutrally stable response, evident
 from the time histories showing non incresing crest and troughs in
 attitude over 6 orbits')


xfinal =

   0.021414146457538
   0.062360917783887
  -0.071803040606404
   0.995237366114879
   0.000136192378780
  -0.001052654303743
   0.000062483571553


qfinalmag =

   1.000000270643444

The system produces a neutrally stable response, evident from the time
 histories showing non incresing crest and troughs in attitude over 6
 orbits
```



Published with MATLAB® R2019a