Posting Date: Monday Oct. 28th.
Due Date: Monday Nov. 4th.

1. Complete the MATLAB template file `ffunctgravgradsc02_temp.m` by completing the parts of the code where ???? appears. The result will be the MATLAB function `ffunctgravgradsc02.m`. This function must compute the state rate dynamics model for a rigid-body spacecraft undergoing forcing by gravity-gradient torque while flying in a circular orbit above a body with a $1/r^2$ gravity field. The first 4 elements of the state vector are the attitude quaternion that parameterizes the rotation from local-level orbit-following coordinates to spacecraft body coordinates. The last 3 elements of the state vector are the angular velocity of the rigid body relative to inertial coordinates and given in the body-axis frame. Hand in your completed template file.

   Note that this same function can be used to model the motions of a rigid-body spacecraft that is not subject to gravity-gradient torques due to its being far from any gravitating body. This is accomplished by setting the mean-motion orbital angular velocity input `norbit` to 0. In this case, the attitude quaternion in the first 4 elements of the state vector parameterizes the rotation from inertial coordinates to spacecraft body coordinates.

2. Complete the MATLAB template file `script_simgravgradsc11_temp.m` by completing the parts of the code where ???? appears. The result will be the MATLAB script `script_simgravgradsc11.m`. Use your function `ffunctgravgradsc02.m` from Part 1, your function `rotmatquaternion.m` from Assignment 3, and your completed script `script_simgravgradsc11.m` to compute the angular velocity vector time history in body-fixed coordinates, both from numerical integration of the nonlinear differential equations and from theory. This script will also compute the angular momentum time histories in body-fixed coordinates and in inertial coordinates. Hand in your completed template file along with two plots: The plot of the inertial angular momentum time history, which should show constant values for all 3 components, and the plot of the body-axes angular velocity vector component time histories, both from ode45.m and from your completed theoretical formulas.

   As a way of checking your work, the following operations

   ```
   >> format long
   >> xfinal = xhist(end,:)'
   >> qfinalmag = norm(xfinal(1:4,1))
   ```

   produce the following final state vector and quaternion magnitude for the ode45.m time history that is calculated by `script_simgravgradsc11.m`:

   ```
   xfinal =
      0.076778204795434
     -0.041341864928188
     -0.984339512015987
      0.153186692608211
     -0.013613444068028
   ```

0.007330204591270
0.104719755119660

qfinalmag =
0.999997240113923

Numerical integration errors are what cause this final quaternion magnitude not to be 1.

Also hand in your answers to these question: How well is inertial angular momentum conserved by this torque-free motion? How well do the ode45.m and the theoretical angular velocity time history components match?

3. Use the provided MATLAB script `script_simgravgradsc13.m`, the provided data file `simgravgradsc13_data.mat`, your functions `ffunctgravgradsc02.m` and `rotmatquaternion.m`, and the provided functions `qrtmul.m` and `yawpitchrollcalc02.m` to simulate the attitude motions of a rigid-body spacecraft that is circling the Earth in a Low-Earth Orbit (LEO) and that is influenced by gravity-gradient torques. Hand in the resulting plot of the roll, pitch, and yaw angle time histories.

   As a way of checking your work, the following operations

```
>> format long
>> xfinal = xhist(end,:)'
>> qfinalmag = norm(xfinal(1:4,1))
```

   produce the following final state vector and quaternion magnitude for the ode45.m time history that is calculated by `script_simgravgradsc13.m`:

xfinal =
0.021414146457538
0.062360917783886
-0.071803040606404
0.995237366114880
0.000136192378780
-0.001052654303743
0.000062483571553

qfinalmag =
1.000000270643444

   Numerical integration errors are what cause this final quaternion magnitude not to be 1.

   Also hand in your answer this question: Does this spacecraft design produce a neutrally stable response that will keep the spacecraft close to being aligned with the orbit-following local-level coordinate system indefinitely?

4. Consider a spin-stabilized spacecraft that is not axially symmetric. It has the body-fixed principal axes moment-of-inertia matrix

$$I_{MoI} = \begin{bmatrix} I_{tr1} & 0 & 0 \\ 0 & I_{tr2} & 0 \\ 0 & 0 & I_{spin} \end{bmatrix}$$

where the $\hat{k}$ body-fixed principal axis is the nominal spin axis and where the two transverse principal moments of inertia I_{tr1} and I_{tr2} are unequal. Use the torque-free rigid-body equations from lecture in order to prove that the nutation frequency of this system is

$$\omega_{nut} = \sqrt{\frac{(I_{spin} - I_{tr1})(I_{spin} - I_{tr2})}{I_{tr1} I_{tr2}}}\,\omega_{3avg}$$

and that the time histories for the other two body-axes spin vector components are

$$\omega_1(t) = \omega_1(0)\cos(\omega_{nut} t) + \omega_2(0)\,\mathrm{sign}(I_{tr2} - I_{spin})\sqrt{\frac{I_{tr2}(I_{spin} - I_{tr2})}{I_{tr1}(I_{spin} - I_{tr1})}}\,\sin(\omega_{nut} t)$$

$$\omega_2(t) = \omega_2(0)\cos(\omega_{nut} t) - \omega_1(0)\,\mathrm{sign}(I_{tr2} - I_{spin})\sqrt{\frac{I_{tr1}(I_{spin} - I_{tr1})}{I_{tr2}(I_{spin} - I_{tr2})}}\,\sin(\omega_{nut} t)$$

if $\omega_1(t)$ and $\omega_2(t)$ are sufficiently small.

Hint: The third component of Euler's equations for torque-free rigid-body motion implies that $\omega_3(t)$ is nearly constant if $\omega_1(t)$ and $\omega_2(t)$ are both sufficiently small. Assume that $\omega_3(t) = \omega_{3avg}$, a constant. Under this assumption, you should be able to generalize the analysis done in lecture of the first 2 components of Euler's equations for torque-free rigid body motion. The result should be the nutation frequency and the $\omega_1(t)$ and $\omega_2(t)$ time history formulas shown above. Be careful to show all of the significant steps in these derivations.

5. Complete the MATLAB template file `script_simgravgradsc12_temp.m` by completing the parts of the code where ???? appears. The result will be the MATLAB script `script_simgravgradsc12.m`. You will need to use the formulas in Part 4 in order to complete this template. Note that the moment-of-inertia matrix in the original body-axes coordinate system is not diagonal. Thus, the original coordinate system is not the principal-axes system that is assumed in Part 4. Therefore, this script uses MATLAB'S `eig` function in order to determine a transformation to a new body-fixed coordinate system that is a principal-axes system. The analysis results from Part 4 are applied in this new coordinate system. Use the provided data file `simgravgradsc12_data.mat` and your functions `ffunctgravgradsc02.m` and `rotmatquaternion.m` in order to run your completed script `script_simgravgradsc12.m` to compute the angular velocity vector time history in principal-axes coordinates, both from numerical integration of the nonlinear differential equations and from the theory. This script also computes the angular momentum time histories in the original body-fixed coordinates and in inertial coordinates. Hand in your completed template file along with two plots: The plot of the principal-axes angular velocity component time histories, both from ode45.m and from your completed theoretical

formulas, and the plot of the difference between these two solutions. Also hand in your final state vector given to 15 significant digits by executing the commands

```
>> format long
>> xfinal = xhist(end,:)'
```

Also hand in your answer to these questions: There is a significant qualitative difference between the body-axis angular momentum component time histories plot generated by `script_simgravgradsc11.m`. and the corresponding plot generated by the present script. What is it? What causes this qualitative difference?

6. Use the provided MATLAB script `script_simgravgradsc14.m`, the provided data file `simgravgradsc14_data.mat`, your functions `ffunctgravgradsc02.m` and `rotmatquaternion.m`, and the provided functions `qrtmul.m` and `yawpitchrollcalc02.m` to simulate the attitude motions of a rigid-body spacecraft that is circling the Earth in LEO and that is influenced by gravity-gradient torques. Hand in the resulting plot of the roll, pitch, and yaw angle time histories. Also hand in your final state vector given to 15 significant digits by executing the commands

```
>> format long
>> xfinal = xhist(end,:)'
```

Also hand in your answer this question: Does this spacecraft design produce a neutrally stable response will keep the spacecraft close to being aligned with the orbit-following local-level coordinate system indefinitely?

7. Use the provided MATLAB script `script_simgravgradsc15.m`, the provided data file `simgravgradsc15_data.mat`, your functions `ffunctgravgradsc02.m` and `rotmatquaternion.m`, and the provided functions `qrtmul.m` and `yawpitchrollcalc02.m` to simulate the attitude motions of a rigid-body spacecraft that is circling the Earth in LEO and that is influenced by gravity-gradient torques. Hand in the resulting plot of the roll, pitch, and yaw angle time histories. Also hand in your final state vector given to 15 significant digits by executing the commands

```
>> format long
>> xfinal = xhist(end,:)'
```

Also hand in your answer this question: Does this spacecraft design produce a neutrally stable response will keep the spacecraft close to being aligned with the orbit-following local-level coordinate system indefinitely?