ASE 366K Problem Set #5

Posting Date: October 11, 2010
Due Date: October 15, 2010

Turn in homework at the beginning of class. If you have any doubts about your ability to make it to class on time, you may turn in your problem solutions by sliding them under my office door (WRW 411C) the day before they are due. Be prepared to answer any of these problems—or similar problems—on an upcoming exam. You may discuss your solutions with classmates, but do not swap work. Only one of the homework problems, chosen at random, (or one part of one problem, if that part is sufficiently involved) will be graded thoroughly. The grade given to the homework set will be based primarily on this problem. The other problems will be graded on a credit/no-credit basis with credit depending on whether a solution was attempted.

(1) This problem will be an exercise in manipulating the canonical units DU and TU defined in lecture. Consider an earth-orbiting satellite with eccentricity $e = 0.1$ and semi-major axis $a = 1.2$ DU, where we take $DU = r_\oplus$ (one Earth radius). At the instant when the satellite intersects the semi-minor axis, it experiences an impulsive velocity change (called a "delta v") as a result of a near-instantaneous burn of its thrusters. The impusive velocity change puts the satellite on a parabolic trajectory.
  - (a) Calculate the velocity of the satellite, expressed in DU/TU, just prior to burn.
  - (b) Calculate the required velocity change Δv, expressed in DU/TU. Assume that the direction of the Δv is aligned with the orbit velocity just prior to burn so that the inertial velcity of the satellite increases by exactly Δv.
  - (c) Convert the Δv to units of km/s.

(2) Write a Matlab script that solves Kepler's equation for a given M and e. Solve Kepler's equation by all three different methods shown in lecture: (1) graphical method, (2) successive iterations, and (3) Newton-Raphson method. You may either write the script from scratch or fill in code where you see the token "????" in the file keplerEq_temp.m, available on the course website. Turn in a printout of your script. Use your script to answer the following questions:
  - (a) For $e = 0.1$ and $M = 0.1$, how many iterations are required for each of the two iterative methods (successive iterations and Newton-Raphson methods) to converge? Provide a plot showing each method's solution as a function of iteration index i.
  - (b) Same as above but for $e = 0.7$ and $M = 0.1$.
  - (c) Same as above but for $e = 0.7$ and $M = 3$.
  - (d) Describe the behavior of the successive iterations method as $e \to 1$.

(3) Write a Matlab function called `pv2orbitalElements` that takes in as arguments the vectors $\boldsymbol{R}(t)$ and $\dot{\boldsymbol{R}}(t)$, which are the position and velocity vectors of an orbiting body expressed in an inertial reference system at time t, the time t, and the gravitational parameter μ. The function should return the corresponding classical orbital elements a, e, i, ω, Ω, and τ, and the true anomaly f. A call to your function should look like

```
[aOrb,eOrb,iOrb,omegaOrb,OmegaOrb,tauOrb,fOrb] ...
```

```
= pv2orbitalElements(R,Rdot,t,mu);
```

All units should be in radians, meters, kilograms, and seconds. Use your function to solve for the classical orbital elements and f corresponding to

$$\boldsymbol{R} = \begin{bmatrix} 7378 \\ 0 \\ 0 \end{bmatrix} \text{km} \tag{1}$$

$$\dot{\boldsymbol{R}} = \begin{bmatrix} 0 \\ 8 \\ 3 \end{bmatrix} \text{km/s} \tag{2}$$

for an earth-orbiting satellite at a time $t = 10000$ seconds. Turn in a printout of your function.

Extra credit: Write `orbitalElements2pv`, the functional inverse of `pv2orbitalElements`. A call to it should look like

```
[R,Rdot,t] = ...
   orbitalElements2pv(aOrb,eOrb,iOrb,omegaOrb,OmegaOrb,tauOrb,fOrb,mu);
```