ASEN 5070-Statistical Orbit Determination

Homework 1

Zach Dischner

September 4, 2012

1.0 Cartesian Coordinates to Keplarian Elements

For this problem, we had to convert Cartesian position and velocity vectors into Keplarian elements. To do this, I simply followed the handout included in the assignment and wrote a conversion script in MATLAB.

Using the following initial conditions:

$$R = -2436.45\imath - 2436.45\jmath + 6891.037k \text{ km}$$

$$V = R' = 5.088611\imath - 5.088611\jmath + 0.0k \text{ km/s}$$

I was able to get the following Keplarian orbit elements:

Table 1: Problem 1 Solutions

a	7.7122e+03	[Km]
e	9.9944e-04	[]
i	1.1071	[Rad]
Omega	2.3562	[Rad]
w	1.5708	[Rad]
nu	0	[Rad]

2.0 Keplarian Elements to Cartesian Coordinates

This problem is the opposite procedure as in problem 1.0. Again, for this problem I followed the procedure given in a course handout to write a MATLAB script to perform the conversion. Using the Keplarian elements listed in Table 1, the following Cartesian position and velocity vectors were solved for:

Table 2: Problem 2 Solutions

R	$-2436.45\imath - 2436.45\jmath + 6891.037k$	[Km]
V	$5.088611\imath - 5.088611\jmath$ – $1.053e^{-15}k$	[Km/s]

The results match the initial conditions given in Problem 1 to around 8 decimals working with the units given.

3.0 2-Body Acceleration Due to Gravity Alone

Given the gravity potential function, I was able to successfully differentiate the gravity potential function in a manner that follows.

Given:

$U = \frac{\mu}{R}$	$R = \sqrt{x^2 + y^2 + z^2}$	$\nabla U = \frac{\partial U}{\partial x}\hat{\imath} + \frac{\partial U}{\partial y}\hat{\jmath} + \frac{\partial U}{\partial z}\hat{k}$

The partial derivative of each vector component can be treated as follows:

$$\frac{dU}{dX} = \frac{dU}{dR} * \frac{dR}{dX} ==> -\frac{1}{2} * u/(\sqrt{x^2 + y^2 + z^2}^3) * 2 * x$$

When done for each component, the resulting acceleration due to gravity in this simple 2-body problem becomes:

$$\Delta U = -\frac{\bar{r} * u}{\overline{r^3}}$$

4.0 Integration of Equations of motion

For this problem, I used the above relation to craft a function that returned the derivative of position and velocity vector inputs. I.E. Given R and V, it returns V and A. This function was integrated over using MATLAB’s ode45. The numerical integration was carried out over 2 orbit periods, with a 20-second interval.

5.0 Numerical Integration Results

The results of the previous numerical integration are shown in the radius, velocity, and acceleration magnitudes over the integration period.

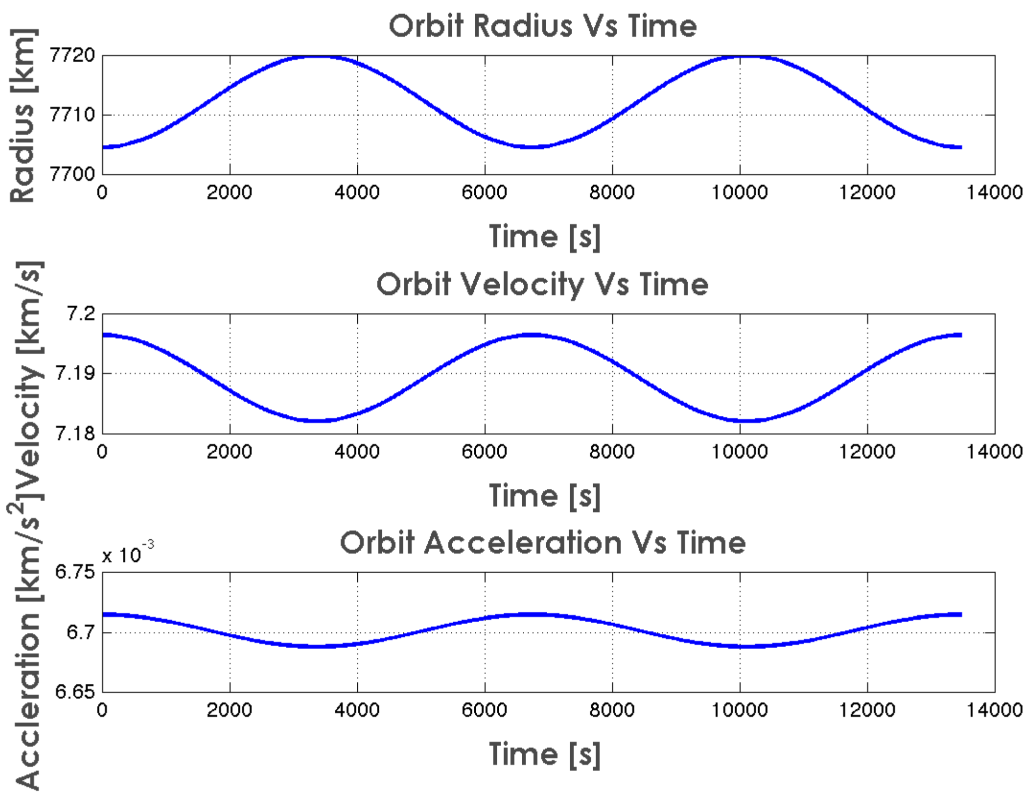

Figure 1: Orbit Radius, Velocity, and Acceleration with Time

As is to be expected in an orbit integration plot, the position and velocities are orthogonal in nature.

6.0 <u>Total Energy Check</u>

To check the validity of the integration, I computed total energy over the orbit periods.

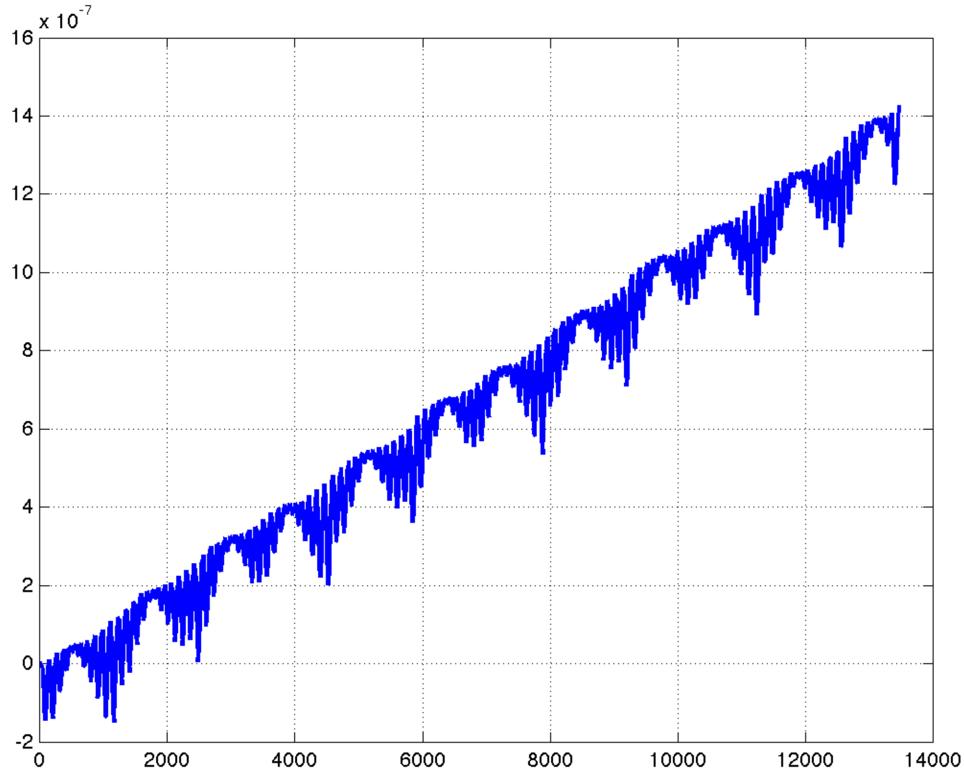

Figure 2: Orbit Total Energy

As can be seen above, the orbit total energy was nearly zero throughout the orbit. Its value remains slightly larger than zero due to integration errors. Lowering the integration tolerance on ode45 will change the shape of the above plot, and lower the magnitude of total energy. See below, where the integration tolerance was cut by 3 orders of magnitude.

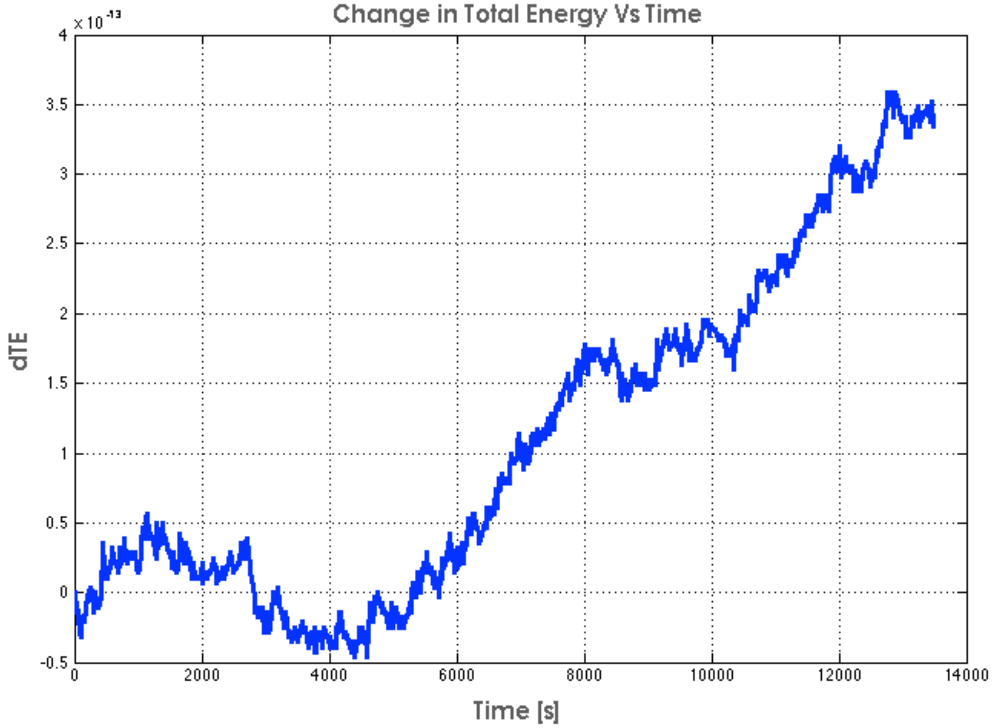

Figure 3: Total Energy Vs Time

7.0 Book Problem

Problem 1, chapter 1 was a book problem, where I used the Newton iteration scheme to determine the initial conditions for a perturbed satellite observation. The iteration was written up in MATLAB, so that initial perturbed conditions were used to determine true initial conditions as:

Parameter	Initial Value	Final Value
X0	1.5	1.00007364093016
Y0	10	7.99999999961264
Xp0	2.2	1.99998473372831
Yp0	0.5	0.999982038956773
g	0.3	0.499992815931914