16.07 Lab I

Issued: October 16, 2009
Due: Monday, October 26, 2009

Introduction

This laboratory is the first in a series that deal with celestial and spacecraft dynamics. In this lab, you will create a MATLAB code that will allow you to numerically confirm Kepler's laws as well as simulate the motion of the earth and the other planets in the solar system as well as spacecraft in orbit about or on the way to one of them. Since this lab will focus on the earth and moon system, the presence of the sun will be ignored. In order to accomplish these tasks, your MATLAB code will integrate the equations of motion in time, starting from an initial condition. In later labs, you will add features like spacecraft thrusting, which may allow you to simulate orbital transfer, rendezvous, descent and ascent from planetary surfaces and other more complicated situations.

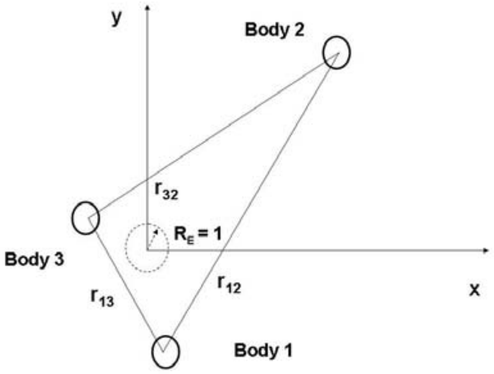

Figure 1: 3 Body coordinate system.

Figure 1 shows the coordinate system for a generic problem involving the three planetary/spacecraft bodies. The bodies are labeled as Body 1, Body 2 and Body 3, but these names are just an example. These bodies will turn into the Earth, the Moon and a spacecraft. The important thing is to choose a convention that you understand and can be consistent with.

The first step is to formulate the equations we would like to solve with the MATLAB code. In the general case, we desire to calculate the positions and velocities of each body as a result of its interaction with the other members of the system. For our purposes, we only consider the gravitational forces resulting from the mutual attraction of body pairs. Thus, the motion of any of the three moving bodies must consider the gravitational influence of the other two moving bodies. For instance, the attractive force that Body 2 exerts on Body 1 is given by,

$$\boldsymbol{F}_{12} = \frac{\mu_2 m_1}{r_{12}^3}\boldsymbol{r}_{12}. \tag{1}$$

Here, $\mu_2 = Gm_2$, the gravitational parameter of Body 2, where G is the universal gravitational constant[1] and m_2 is the mass of Body 2. The mass of Body 1 is denoted by m_1, and r_{12} is the magnitude of the distance vector that points from Body 1 to Body 2, $\boldsymbol{r}_{12} = \boldsymbol{r}_2 - \boldsymbol{r}_1$.

Celestial bodies move in three dimensions, so in general you would write your equations for the x, y,and z positions of each body. However, for this lab, we will take the position of the earth, moon and spacecraft system and their motions as confined to the x, y plane. Therefore, the z equations are not required.

For numerical purposes it is convenient to write the equation of motion $\ddot{\boldsymbol{r}} = \boldsymbol{F}/m$ for each body as a system of first order equations. That is, we write,

$$\dot{\boldsymbol{r}} = \boldsymbol{v}, \tag{2a}$$

$$\dot{\boldsymbol{v}} = \boldsymbol{F}/m. \tag{2b}$$

Therefore, to describe the motion of a body in two dimensions, four state variables are required – two for position and two for velocity. For three moving bodies confined to planar motion in the x, y plane,we need only four for each body for a total of twelve. We will use a Cartesian coordinate system which means that the equations should be separated into Cartesian x,and y components. The state vector $\mathbf{X}$ contains all variables that completely describe the state of the system. For purposes of uniformity, you should order the states as follows,

$$\mathbf{X} = [x_1\ y_1\ x_2\ y_2\ x_3\ y_3\ u_1\ v_1\ u_2\ v_2\ u_3\ v_3]^T\,, \tag{3}$$

where x,and y are position components, u is the x-component of velocity, and v is the y-component of velocity.

You will be solving several different problems in this lab and building on your code in future labs. Thus, it will be beneficial to code in a flexible way. For example, you might want to define variables for the masses of the objects, instead of hard-coding the values. Some problems in this or future labs may only deal with two bodies such a planet in orbit around the sun, or in others, the gravitational pull of one of the bodies may be neglected[2]. *However, your code must have the capability to handle up to three bodies* . An effective way to remove the effect of a body on the other bodies in your code is to set its gravitational parameter to zero.

For later work, we present physical parameters for 5 celestial bodies: the sun, the earth, Mars, Jupiter and the moon. Because of the extremely large mass of the sun, it is usual to consider it to be stationary in our coordinate system. The planets, including the earth, orbit around the sun; and the moon orbits around the earth. When considering planetary orbits around the sun or the moon's orbit around the earth, the effect of other celestial bodies are ignored. These simplifications are referred to as the two-body and the three-body problems. The two body problem has an analytic solution in term of the Kepler planetary orbit problem. The three-body solution has no analytic solution and as we shall later see can exhibit quite wild behavior.

[1] $G = 6.673 \times 10^{-11}$ m^3/kg-s^2

[2] For example, the pull of the space shuttle on the Earth could certainly be neglected.

None of the planetary orbits are circular but are slightly elliptical. In addition, the moon's orbit is slightly out of the plane formed by the earth's orbit around the sun. However, when expressing motion of bodies in orbit around the sun or in the case of the moon in orbit about the earth, we assume these orbits to be **circular** and **co-planer**. Also, the planets are not spheres so that their radius is not constant; we will use the equatorial radius. Because, of these approximations, if you look up these physical parameters from various sources, you may notice slight discrepancies in the 3rd or 4th digit.

These parameters are not independent since the orbital velocity depends on the orbital radius. The orbital velocity is presented only for interest; you should obtain it from Kepler's laws and use it as your boundary condition.

Table 1: Physical and Orbital Data

	mass	semimajor axis	radius	orbital velocity
	(kg)	(km)	(km)	(m/s)
Sun	1.989×10^{30}	–	—	—
Earth	5.975×10^{24}	149×10^{6}	6378	29,790
Moon	7.36×10^{22}	384×10^{3}	1737	1023
Mars	6.441×10^{23}	228×10^{6}	3397	24,140
Jupiter	1.899×10^{27}	778×10^{6}	71,492	13,060

When working with problems involving the sun and the planets, it is usual to non-dimensionalize distances with respect to the distance between the sun and the earth: 149×10^{6} km, defined as 1 AU (astronomical unit.), and time by 24*60*60 sec.=1day. (In some studies, the time scale chosen is 365.25*24*60*60 sec.=1 year.)

When working with problems involving the earth and the moon and a spacecraft traveling between them or in orbit about them, normalize the distances by the radius of the earth E_R=6378 km, and time by 24*60*60 sec = 1 day.

Problems

In the following problems, the items you must turn in – either the written answers to questions, or required plots – are highlighted in **bold**. The ordering of these deliverables in your report should match the ordering that appears below.

1. The Equations of Motion

Making use of the state vector (Equation 3) and the system of first order equations (Equation 2), we can write the equations of motion for our system in the form,

$$\dot{\mathbf{X}} = \mathcal{F}\left(\mathbf{X}\right), \tag{4}$$

where the right-hand side is a non-linear function of the state. It turns out that when expressed in the SI system, the units of the distances and masses in our system have very large exponents if we're considering planetary systems. Because of finite machine precision, trying to do calculations with either very large or very small numbers can result in accumulated round-off errors. Thus, when solving equations numerically, it is standard practice to change variables such that the unit length and unit time are characteristic of your system. We will use the bar to denote non-dimensionalized variables. Thus, $\bar{r}$ and $\bar{t}$ will denote non-dimensionalized distance and time, respectively. Note that in this gravitational problem we do not need to scale the mass since we are working with gravitational parameters, μ. In the general $\boldsymbol{F} = m\boldsymbol{a}$ problem, we would need to scale length, time, *and* mass. We will use the radius of the earth as our length scale, and 1 day (24 hours) as our time scale.

In your report, you need only **write the equations of motion for the three-body earth-moon plus spacecraft system, and their derivation, in the non-dimensionalized bar variables**. You should obtain $\mu_{earth} = 11472$ and $\mu_{moon} = 141.3$ in your non-dimensionalized equations.

2. Observing Kepler's Laws

Once you have the equations of motion, you should write your solver using MATLAB. There is a function in MATLAB called `ode45` which nicely solves differential equations of the form given in Equation 4. For details on the use of this function, type `help ode45` into MATLAB. Another command associated with `ode45` is a command to set the error tolerances. The `ode45` function uses a 4-stage Runge-Kutta integration technique, and has a feature to vary the time step to control the error. For example, if the curvature of an orbit gets very high, `ode45` will reduce the time step in order to carefully resolve the trajectory. These error tolerances are set with the command:

```
options=odeset('RelTol',1e-6,'AbsTol',1e-6,'MaxStep',.001);
```

The values shown here are a good starting point. If your calculation takes too long to run you can loosen the tolerances, and if you need more detail in the solution, you can tighten them.

Your solver should consist of a MATLAB script file that sets the solver options, sets the initial state, and then calls `ode45`. One of the input parameters to `ode45` is a function that calculates the state derivative. In addition to the script file, you will need to write a function to define your system of equations.

Once you have a working solver, test it by simulating the orbit of a geostationary satellite in a planar x, y orbit around a fixed Earth. In simulating the motion of a satellite, we ignore the effect of the satellite on the earth, and take the earth to be stationary at the center of our coordinate system. A geostationary satellite orbits at a radius of approximately 42,042 km from the center of the earth so that it remains fixed above a point on the rotating earth; it rotates in inertial space in a period of 23 hrs, 56 minutes and 4.09 sec (because a the earth rotates about the sun and in a time that is close but not exactly equal to 1 year.) You need to transform these initial values into your non-dimensional framework setting $x(0)$, $y(0)$, $u(0)$ and $v(0)$ equal to the non-dimensionalized form of the geostationary satellite parameters. Use the solution to the satellite orbit problem as your initial conditions.

Simply initialize the satellite to start at $(R_0, 0)$ with velocity $(0, v_0)$, where R_0 and v_0 have been non-dimensionalized. The orbit should be nearly circular (try for good accuracy so that the later parts of the lab make sense; strive for say .001) and if the simulation is run for one day, the planet should complete just about one orbit. This is a good quick check that your solver is working.

Kepler's First Law

> *The orbits of the planets are ellipses, with the Sun at one focus of the ellipse.* (We will use the earth and a satellite to validate Kepler's Laws; Kepler didn't have satellites.)

Once your solver is working, you will simulate the eccentric (elliptical) orbit of a satellite around a fixed Earth. Let's call it object A. Start the object at $(10, 0)$ with an initial velocity of $(0, 20)$. **From your simulation, determine the eccentricity and period of object A's orbit, and describe how you determined these values from your numerical results. Compare these results to the theoretical values you would expect, given the initial conditions used**. The lecture notes from lectures L15 and L16 will give you the details of the theory needed. Additionally, the Appendix at the end of this document gives some key formulas and labels critical parts of an elliptical orbit.

Kepler's Second Law

> *The line joining a satellite and the fixed Earth sweeps out equal areas in equal times as the object follows its orbit.* (Again, we will use a satellite in orbit about the earth.)

Now, using the same simulation of object A, calculate the area swept by the object as it completes one orbit. You can do this as a simple post-processing operation once you have determined the position of the object as a function of time. The function `ode45` will return a vector `t` with time values, and an array `X` with the value of the state variables at each time. From this you should be able to compute a new vector `Area` that contains the value of the area swept by the object for each time. **Plot the integrated area swept as a function of time, and verify that the area increases linearly**.

Kepler's Third Law

> *The ratio of the squares of the orbital periods of two objects is equal to the ratio of the cubes of their semimajor axes.*

Here we consider a different planet, call it object B. The initial velocity for this object is the same as for object A, but its starting position is twice as distant from the earth. **Verify the ratio in Kepler's Third Law, for object A and B using the results of your numerical simulation. Turn in plots showing both orbits, and label the semimajor axes for both.** Your plots should show the results of the independent simulations of objects A and B; you should not simulate them simultaneously.

The ratio of a orbiting object's period squared to its semimajor axis cubed can be shown to equal $4\pi^2/\mu$. **Verify this ratio from your numerical simulations.** Remember to use non-dimensional values.

3. Three-Body Simulation

This question will explore three-body interaction. In general, the three-body problem has no analytical solution. In some cases, the behavior of a three-body system is chaotic. All three bodies have zero initial velocity.

Body 1 is located at $(3, 0)$ and has $\bar{\mu}_1 = 1$.
Body 2 is located at $(2.5, -1)$ and has $\bar{\mu}_2 = 1$.
Body 3 is located at $(1, 0)$ and has $\bar{\mu}_3 = 10$.

Run the simulation for 1 year[3], and **turn in a plot of the trajectories of the three bodies**. Use the following command to scale your axes:

```
axis( [0.0 3.25 -1.25 1.75] );
```

After making your plot, pencil in little arrows along the trajectories that indicate the direction the planets moved in. Discuss the behavior you observed in the simulation. You need to submit your code for this question, but only this question. You will submit it online through the course website. Do not submit a paper copy. When your code is executed, it must run the simulation, and then produce the previously specified plot. The code you submit must have the following naming format:

```
"your_ name"_lab1_filename1.m
"your_ name"_lab1_filename2.m
```

You should only need to **submit two files electronically: the calling script, and your function to calculate the state derivative.**

3. Earth-Moon Simulation

First, develop the solution for the moon rotating about the earth as a "Kepler problem" with the moon orbiting the fixed earth with a modified μ_{eff}. **Obtain the period of the moon rotating about the earth from this solution.**

Now you will simulate the orbit of the moon about the earth as a 2-body problem; this is in fact a study of their relative motion. Since both the earth and the moon, orbit about their center of mass, in order to get a steady repeating orbital solution, you must use the center of mass as the origin of your coordinate system and then must locate the earth and the moon relative to their center of mass. You must give each body an initial velocity appropriate to its behavior as given by the exact solution of the two-body problem by Kepler's law using the transformation of the 2-body problem to a 1-body or "Kepler problem". Non-dimensionalize distances by the radius of the earth; non-dimensionalize time by 1 day. With this choice, the distance between the earth and moon is 60.27 earth radii and the orbital period is about 27.3 days. Run your simulation for 3-4 months to verify that the bodies do not move off course. **Turn in a plot of the trajectory of the earth-moon system run for a period of 4 months.**

[3] one year *simulation time*, not actual time—don't take this as a deadline extension…

4. Earth-Moon Simulation-Rotating Coordinates

Now program the equations for the earth moon relative motion in a rotating coordinate system. You have already derived the expression for the inertial acceleration of a point in rotating coordinates; the gravitation forces between the bodies are added without modification. **Turn in your derivation of these governing equations.** Now, if you equations are correct, you should be able to place the earth and the moon at their initial positions and see no motion: i.e they will be fixed in your coordinate system. You may have to adjust the values of Ω slightly to get them to stay absolutely fixed in the rotating system; shouldn't be a big adjustment.

Turn in the value of Ω that leads to stationary positions of the earth and moon in the rotating coordinate system. Submit both the script file and the function file as a zip file to the course website.

Report Preparation

There is no need to produce a fancy report. You should include all the figures requested and answer all the questions posed. If your write-up is neat, organized, and readable, then there is no need to type it.

Appendix

Figure 2 labels key parts of the elliptical orbit, where π is the periapsis, or point of closest approach, α is the apoapsis, or the furthest point in the orbit, O is the location of the gravitational center, the planet being orbited, a is the semimajor axis, b is the semiminor axis, p is called the parameter of the conic, and e is the eccentricity of the orbit. The following are a few relationships you might need:

$$b = a\sqrt{1-e^2}, \tag{5}$$

$$r_\pi = \frac{p}{1+e} = a(1-e), \tag{6}$$

$$r_\alpha = \frac{p}{1-e} = a(1+e). \tag{7}$$

To find the relationship between the velocity of the orbiting planet and its distance from the gravitational center, the energy equation can be used,

$$\frac{1}{2}v^2 - \frac{\mu}{r} = -\frac{\mu}{2a}. \tag{8}$$

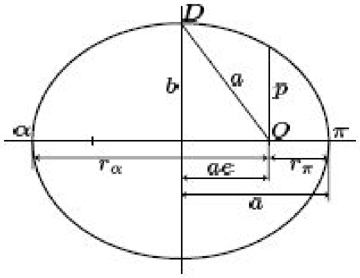

Figure 2: An elliptical orbit with key values labeled.

MIT OpenCourseWare
http://ocw.mit.edu

16.07 Dynamics
Fall 2009

For information about citing these materials or our Terms of Use, visit: http://ocw.mit.edu/terms.