Assignment #6: C Software Design

Due Date: Friday, March 11 at 5:00 PM

Type of Assignment: Group

Submission Procedure: Upload copies of your assignment components (i.e., source files, compiled C program, command line instructions, Bash script, output, and plots), assignment report, and profile report to both group members' private git repositories in the directory:

`Assignments/Assignment_6/Submission`

Grade Details: This assignment is worth 150 points.

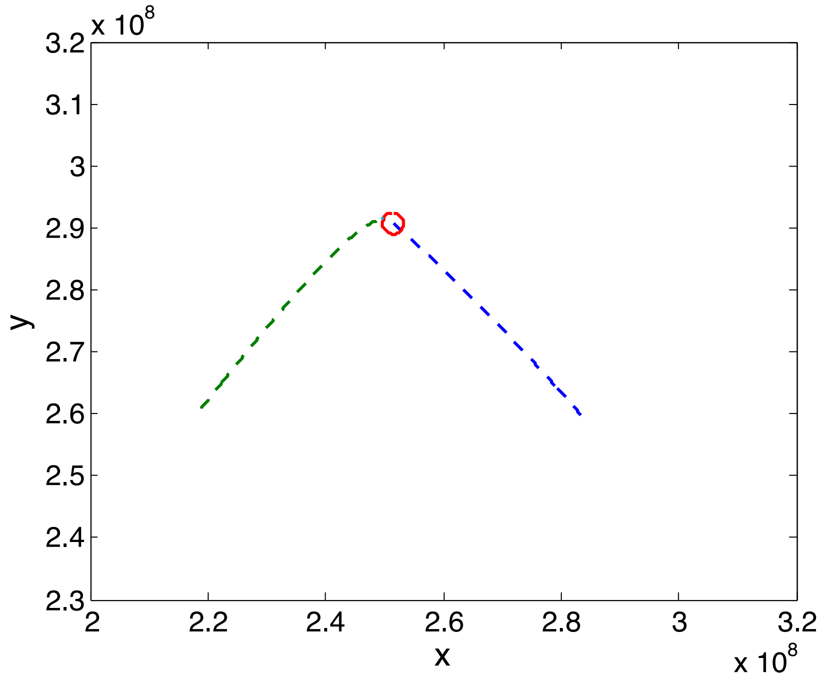
Instructions:

This assignment involves developing a C program for solving the spacecraft-Earth-moon three-body problem. It essentially involves converting the MATLAB program you designed for Assignment #2 to the C programming language, but there are important differences as well. There are several components required in your C program, and these components are described in detail below. **Be sure to document your code well, and make sure your code can compile and run on the CSCI OpenStack Linux instance.**

Background:

Three of our best astronauts were on their way to the moon when a series of explosions knocked out the main power of their spacecraft and wrecked havoc on life support. Now, the spacecraft is on a prime collision course with the moon, as illustrated in the figure on the top of the next page. In the figure, the green dashed line denotes the projected path of the spacecraft, the blue dashed line denotes the projected path of the moon, and the red circle denotes the final position of the moon.

Luckily, there is a chance to use onboard thrusters to change the current velocity of the spacecraft and hence its projected trajectory. Your objective is to determine the velocity change required in order to guarantee the spacecraft's successful return to Earth. Ideally, you would also like to get the astronauts home as fast as possible due to the failing life support system. However, it is not known just how much power can be supplied to the thrusters and how much longer the thrusters will be operational. It is possible to use the moon's gravity field in order to slingshot the spacecraft around the moon and back to Earth. This will allow for a small change in current velocity to be used to get the astronauts safely back home. This is also what was used to get the Apollo 13 astronauts back home.



Gravitational Forces:

In order to determine the required change in current velocity, you will first need to understand how to simulate the moon-Earth-spacecraft system. The combined gravitational pulls of the Earth and the moon alter the path of the spacecraft as a function of time, and, similarly, the combined gravitational pulls of the Earth and the spacecraft alter the path of the moon.

Consider two bodies A and B with masses m_A and m_B respectively. Assume that body A is centered at a position (x_A, y_A) and that body B is centered at (x_B, y_B). Then, the distance between bodies, denoted by d_{AB} or d_{BA}, is given by:

$$d_{AB} = d_{BA} = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2}. \tag{1}$$

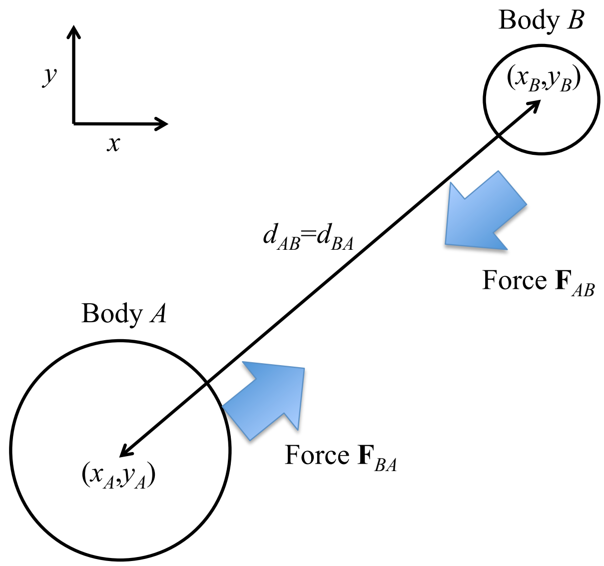
The body A exerts a gravitational force on body B of magnitude

$$F_{AB} = \frac{G m_A m_B}{d_{AB}^2} \tag{2}$$

where G is the gravitational constant

$$G = 6.674 \times 10^{-11} \mathrm{N}(\mathrm{m/kg})^2. \tag{3}$$

The force exerted on B by A is pointed directly in the direction of A. Similarly, the body B exerts a gravitational force on body A of equal magnitude and opposite direction. This situation is depicted in the figure given below.



Note that the force exerted on B by A and the force exerted on A by B are both vectors. We can explicitly state the x- and $y-$ components of the vectors in terms of the positions of the two bodies. Notably, we have:

$$(F_{AB})_x = \frac{Gm_A m_B(x_A - x_B)}{d_{AB}^3} \tag{4}$$

$$(F_{AB})_y = \frac{Gm_A m_B(y_A - y_B)}{d_{AB}^3} \tag{5}$$

$$(F_{BA})_x = \frac{Gm_B m_A(x_B - x_A)}{d_{BA}^3} \tag{6}$$

$$(F_{BA})_y = \frac{Gm_B m_A(y_B - y_A)}{d_{BA}^3}. \tag{7}$$

Note immediately that $(F_{AB})_x = -(F_{BA})_x$ and $(F_{AB})_y = -(F_{BA})_y$. This is because the force vectors $\mathbf{F}_{AB}$ and $\mathbf{F}_{BA}$ are perfectly balanced.

We can repeat the exercises above to precisely find the forces exerted on the spacecraft, moon, or Earth by the other two respective bodies. In what follows, we will use the notation S to denote the spacecraft, M to denote the moon, and E to denote the Earth. For example, x_S will denote the x-coordinate of the center of the spacecraft, d_{SE} will denote the distance between the spacecraft and the Earth, and $(F_{EM})_x$ will denote the x-component of the force exerted on the moon by the Earth.

Equations of Motion:

Now that we know exactly what the gravitational forces are in our moon-Earth-spacecraft system, we can discuss the equations governing the motion of the system. The basis of this discussion will be precisely Newton's second law: $F = MA$. That is, force equals mass times acceleration. Consider the acceleration of the spacecraft in the x-direction which we will denote by a_{S_x}. To determine such an acceleration, we simply need to know the forces exerted on the spacecraft in the x-direction. These are precisely the gravitational forces exerted on the spacecraft by the moon and the Earth. Hence, Newton's second law gives us:

$$m_S a_{S_x} = (F_{MS})_x + (F_{ES})_x. \tag{8}$$

We can repeat this line of thought to obtain equations for the acceleration of the spacecraft in the y-direction as well as the accelerations of the moon and the Earth. ***In what follows, we make the following assumption: the gravitational pulls of the moon and the spacecraft do not alter the path of the Earth.*** The basis behind this assumption is that the mass of the Earth is so great that the gravitational forces exerted on the Earth by the other two bodies is negligible. With this assumption in mind, we obtain the following six equations for the accelerations of the moon-Earth-spacecraft system:

$$a_{S_x} = \frac{(F_{MS})_x + (F_{ES})_x}{m_S} \tag{9}$$

$$a_{S_y} = \frac{(F_{MS})_y + (F_{ES})_y}{m_S} \tag{10}$$

$$a_{M_x} = \frac{(F_{EM})_x + (F_{SM})_x}{m_M} \tag{11}$$

$$a_{M_y} = \frac{(F_{EM})_y + (F_{SM})_y}{m_M} \tag{12}$$

$$a_{E_x} = 0 \tag{13}$$

$$a_{E_y} = 0. \tag{14}$$

The above accelerations are related to the velocities of our bodies through the following derivative relations:

$$a_{S_x} = \frac{dv_{S_x}}{dt} \tag{15}$$

$$a_{S_y} = \frac{dv_{S_y}}{dt} \tag{16}$$

$$a_{M_x} = \frac{dv_{M_x}}{dt} \tag{17}$$

$$a_{M_y} = \frac{dv_{M_y}}{dt} \tag{18}$$

$$a_{E_x} = \frac{dv_{E_x}}{dt} \tag{19}$$

$$a_{E_y} = \frac{dv_{E_y}}{dt} \tag{20}$$

where v_{S_x}, for example, is the velocity of the spacecraft in the x-direction. Finally, the above velocities are related to the center positions of our bodies through the following derivative relations:

$$v_{S_x} = \frac{dx_S}{dt} \tag{21}$$

$$v_{S_y} = \frac{dy_S}{dt} \tag{22}$$

$$v_{M_x} = \frac{dx_M}{dt} \tag{23}$$

$$v_{M_y} = \frac{dy_M}{dt} \tag{24}$$

$$v_{E_x} = \frac{dx_E}{dt} \tag{25}$$

$$v_{E_y} = \frac{dy_E}{dt}. \tag{26}$$

Euler's Method:

We now have equations governing the positions, velocities, and accelerations of our bodies in terms of derivative relations. So, how do we solve them? Unfortuntely, the given equations are rather nasty nonlinear ordinary differential equations that are not susceptible to arithmetic or analytical manipulation. However, we can use a very basic ***numerical method*** to ***integrate*** the given equations in time. The idea is as follows. Suppose we have an equation of the form:

$$\frac{dz}{dt}(t) = g(z(t), t). \tag{27}$$

The function g can be any function of two variables. Now, by definition, we know that:

$$\frac{dz}{dt}(t) = \lim_{\delta \to 0} \frac{z(t+\delta) - z(t)}{\delta}. \tag{28}$$

Hence, a reasonable approximation might be to replace δ in the above relation by some small but finite ***time-step size*** Δt. This yields the following approximation:

$$\frac{dz}{dt}(t) \approx \frac{z(t+\Delta t) - z(t)}{\Delta t}. \tag{29}$$

By combining the above equation with the equation given in (27), we obtain:

$$z(t+\Delta t) \approx z(t) + \Delta t g(z(t), t). \tag{30}$$

The above is Euler's approximation of the value of z at time $t + \Delta t$, and the resulting scheme of approximation is called Euler's method. It is the most basic numerical method for solving ordinary differential equations on a computer.

If we employ Euler's method to find the velocities of the moon, Earth, and spacecraft at time $t + \Delta t$ given the corresponding velocities and gravitational forces at time t by solving the equations given by (9)-(20), we end up with the following approximations:

$$v_{S_x}(t+\Delta t) \approx v_{S_x}(t) + \Delta t \frac{(F_{MS})_x(t) + (F_{ES})_x(t)}{m_S} \tag{31}$$
$$v_{S_y}(t+\Delta t) \approx v_{S_y}(t) + \Delta t \frac{(F_{MS})_y(t) + (F_{ES})_y(t)}{m_S} \tag{32}$$
$$v_{M_x}(t+\Delta t) \approx v_{M_x}(t) + \Delta t \frac{(F_{EM})_x(t) + (F_{SM})_x(t)}{m_M} \tag{33}$$
$$v_{M_y}(t+\Delta t) \approx v_{M_y}(t) + \Delta t \frac{(F_{EM})_y(t) + (F_{SM})_y(t)}{m_M} \tag{34}$$
$$v_{E_x}(t+\Delta t) \approx v_{E_x}(t) \tag{35}$$
$$v_{E_y}(t+\Delta t) \approx v_{E_y}(t). \tag{36}$$

It should be noted that in order to evaluate the gravitational forces listed above, the center positions of the moon, Earth, and spacecraft must be known at time t.

Finally, if we know the positions and velocities of the moon, Earth, and spacecraft at time t, we can employ Euler's method to find the corresponding positions at time $t + \Delta t$. The resulting approximations are:

$$x_S(t+\Delta t) \approx x_S(t) + \Delta t v_{S_x}(t) \tag{37}$$
$$y_S(t+\Delta t) \approx y_S(t) + \Delta t v_{S_y}(t) \tag{38}$$

$$x_M(t+\Delta t) \approx x_M(t) + \Delta t v_{M_x}(t) \tag{39}$$
$$y_M(t+\Delta t) \approx y_M(t) + \Delta t v_{M_y}(t) \tag{40}$$
$$x_E(t+\Delta t) \approx x_E(t) + \Delta t v_{E_x}(t) \tag{41}$$
$$y_E(t+\Delta t) \approx y_E(t) + \Delta t v_{E_y}(t). \tag{42}$$

The above discussion suggests the following methodology for simulating the moon-Earth-spacecraft system. Given the velocities and positions of the three bodies at time $t = 0$, we find the gravitational forces between the bodies at time $t = 0$ using the procedure outlined in **Gravitational Forces**. Then, we use the equations given by (31)-(36) to find the velocities at time Δt and the equations given by (37)-(42) to find the positions at time Δt. Once we know those positions and velocities, we can update the time to $t = \Delta t$ and find the gravitational forces at time $t = \Delta t$ and iterate. In this fashion, we can obtain the positions and velocities of the moon-Earth-spacecraft system at time instances $t = 0, \Delta t, 2\Delta t, 3\Delta t, \ldots, n\Delta t, (n+1)\Delta t, \ldots$.

Termination Conditions:

The only question remaining is when to stop the simulation. In this project, we will consider three possible termination conditions:

1. The spacecraft has come within a user-prescribed clearance of the moon.

2. The spacecraft has returned to Earth.

3. The spacecraft is lost to the vast reaches of outer space.

Termination condition #1 will occur if the distance between the center of the spacecraft and the center of the moon dips below the radius of the moon plus the prescribed clearance. That is, termination condition #1 will occur if

$$d_{MS}(t) \leq r_M + clearance \tag{43}$$

where r_M meters is the radius of the moon and *clearance* meters is the user-prescribed moon-spacecraft clearance. Obviously, if termination condition #1 occurs, the mission will be considered a failure.

Termination condition #2 will occur if the distance between the center of the spacecraft and the center of the Earth dips below the radius of the Earth. That is, the spacecraft will have crashed into the Earth at time t if

$$d_{ES}(t) \leq r_E \tag{44}$$

where r_E is the radius of the Earth. If termination condition #2 occurs, the mission will be considered a success.

There is some uncertainty as to when termination condition #3 will occur. For the sake of this project, assume that termination condition #3 will occur if the distance between the center of the spacecraft and the center of the Earth exceeds twice the distance between the center of the moon and the center of the Earth. That is, we assume the spacecraft is lost to space at time t is

$$d_{ES}(t) \geq 2d_{EM}(t). \tag{45}$$

If the spacecraft is lost to space, the mission will be considered a failure.

Physical Property Details:

In order to conduct the simulation depicted in the flow chart, you will need to know various physical properties of the moon, Earth, and spacecraft. These are listed below.

Mass of moon: $m_M = 7.34767309 \times 10^{22}$ kg
Mass of Earth: $m_E = 5.97219 \times 10^{24}$ kg
Mass of spacecraft: $m_S = 28,833$ kg
Radius of moon: $r_M = 1,737,100$ m
Radius of Earth: $r_E = 6,371,000$ m.

Simulation Details:

Before employing thrusters, the initial conditions (the positions and velocities at time $t = 0$)

of the spacecraft are as follows:

$$x_S(0) = d_{ES}(0)\cos(\theta_S) \tag{46}$$
$$y_S(0) = d_{ES}(0)\sin(\theta_S) \tag{47}$$
$$v_{S_x}(0) = v_S(0)\cos(\theta_S) \tag{48}$$
$$v_{S_y}(0) = v_S(0)\sin(\theta_S) \tag{49}$$
$$d_{ES}(0) = 340,000,000 \text{ m} \tag{50}$$
$$v_S(0) = 1,000 \text{ m/s} \tag{51}$$
$$\theta_S = 50^\circ \tag{52}$$
$$x_M(0) = d_{EM}(0)\cos(\theta_M) \tag{53}$$
$$y_M(0) = d_{EM}(0)\sin(\theta_M) \tag{54}$$
$$v_{M_x}(0) = -v_M(0)\sin(\theta_M) \tag{55}$$
$$v_{M_y}(0) = v_M(0)\cos(\theta_M) \tag{56}$$
$$d_{EM}(0) = 384,403,000 \text{ m} \tag{57}$$
$$v_M(0) = \sqrt{\frac{Gm_E^2}{(m_E + m_M)d_{EM}(0)}} \tag{58}$$

$$\theta_M = 42.5^\circ \tag{59}$$
$$x_E(0) = 0 \text{ m} \tag{60}$$
$$y_E(0) = 0 \text{ m} \tag{61}$$
$$v_{E_x}(0) = 0 \text{ m/s} \tag{62}$$
$$v_{E_y}(0) = 0 \text{ m/s} \tag{63}$$

Your job will be to find a perturbed velocity initial condition for the spacecraft:

$$v_{S_x}(0) = v_S(0)\cos(\theta_S) + \Delta v_{S_x} \tag{64}$$
$$v_{S_y}(0) = v_S(0)\sin(\theta_S) + \Delta v_{S_y}. \tag{65}$$

The size of this change is measured as:

$$|\Delta v_S| = \sqrt{\Delta v_{S_x}^2 + \Delta v_{S_y}^2}. \tag{66}$$

Assignment Objectives:

Part 1.1

Write a C program which accomplishes one of the following two objectives, specified by the user at the command line.

1. **Objective 1:** Find the smallest change in initial velocity which will guarantee the spacecraft will make it safely back to Earth. The user should specify the moon-spacecraft clearance in meters as well as the desired absolute accuracy of the final solution in meters per second.

2. **Objective 2:** Find the change in initial velocity which will result in the fastest return to Earth subject to the constraint:

   $$|\Delta v_S| \leq 100 \text{ m/s}. \tag{67}$$

   The user should specify the moon-spacecraft clearance in meters as well as the desired accuracy of the final solution in meters per second.

After solving the user-specified optimization problem, your C program should write the optimal simulation trajectory in a file which can be opened in MATLAB for visualization.

Your C program should be named `ThreeBody`, and a call to your program should take the form:

```
./ThreeBody Objective Clearance Accuracy
```

where `Objective` is either 1 or 2 depending on which optimization problem should be solved, `Clearance` is the user-specified moon-spacecraft clearance in meters, and `Accuracy` is the user-specified absolute accuracy in meters per second. Your C program should then create an output file with the name:

```
Optimum_Objective_Clearance_Accuracy
```

within the folder `Output/` where any periods are replaced with the letter `p`. For instance, if a user types:

```
./ThreeBody 1 10000 0.1
```

then your C program should solve Objective 1 to an absolute accuracy of 0.1 meters per second using a moon-spacecraft clearance value of $10,000$ meters, and it should save the optimal simulation trajectory in the file `Optimum_1_10000_0p1`.

Part 1.2

Construct a makefile to compile and link the C program you built in Part 1.1. Your makefile should also enable the capability to *clean* your directory of any executable or object files. To run this operation, the user should be able to type `make clean` at the command line.

Part 2

Use your C program to solve both Objectives 1 and 2 to an absolute accuracy of 0.5 meters per second with a user-prescribed moon-spacecraft clearance of $10,000$ meters. Plot the resulting simulation trajectories in MATLAB. Make sure to include proper labels with your plots.

Part 3

Write a Bash script which uses your C program to solve both Objectives 1 and 2 to an absolute accuracy of 0.5 meters per second for the following range of user-prescribed moon-spacecraft clearances: 0, 10, 100, $1,000$, $5,000$, $10,000$, $50,000$, and $100,000$ meters. Make sure to include the output files resulting from running your script with your submission.

Assignment Report

You will need to submit an assignment report along with your source files, compiled C program, command line instructions, Bash script, output, and plots associated with Parts 1, 2, and 3 above. Within this assignment report, you should explain the design process you utilized in arriving at your final program, and you should map out the various source files which are used to create your compiled C program.

Profile Report

You will also need to submit a profile report. In particular, you need to use the GNU C profiler, gprof, to time your code compiles and runs successfully. Document the results of the profiler using your code associated with Part 2 of this assignment. You need to analyze both the flat profile and call graph obtained with gprof.

Output Format

You may elect to store your output in any format you desire. However, to read your optimal trajectories into MATLAB, it is recommended that you store a matrix with thirteen columns in your output file. The first column should correspond to the time instances of the simulated trajectory, while the final twelve columns should correspond to the numerically computed solution (positions and velocities of the satellite, Earth, and moon) at these time instances.

Assessment of Accuracy

In this assignment, you will need to guarantee that your solution is within some user-prescribed accuracy. For this purpose, we will utilize the notion of absolute accuracy. If

$\mathbf{v}$ denotes a vector we are trying to approximate and $\widehat{\mathbf{v}}$ denotes our approximation to $\mathbf{v}$, then we denote the absolute error as $e = \|\mathbf{v} - \widehat{\mathbf{v}}\| = \sqrt{\sum_{i=1}^{n} (v_i - \widehat{v}_i)^2}$. If $e \leq tol$ where tol is some user-prescribed accuracy, we then say we have approximated $\mathbf{v}$ to an absolute accuracy of tol.