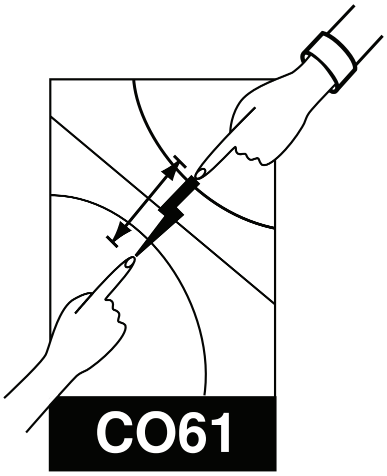
rocket science



revised EG, MK January 2018

Preface

It is common practice in modern software engineering to write programs in a modular and standardised way. To help you get started, appendix A provides you with one or more specific function headers which you **MUST** use to write the functions around which your program should be built and in order to receive a satisfactory mark. Your comments in the header of each function must include:

- author and date,
- purpose (a brief description of what the function does),
- inputs and outputs,
- and, if appropriate, any constraints or limitations of use.

Do not forget that you will also need to keep good records of your progress during this practical in your logbook. If you make plots and/or write any notes electronically, you should print them and affix them into the pages of your logbook. You should have comments in function and script headers of your code, as well as comments within your code. These aspects may all be considered at marking time.

Assessment

You must upload your work electronically via WebLearn as described in the Prelims Handbook **in the week BEFORE** you meet with a demonstrator for marking. This means that in order to come in for marking before the deadline, you must submit your code electronically in the proceeding week. This will be strictly enforced in the final weeks (after week 5), while in earlier weeks, demonstrators may use their discretion (allow marking for electronic submissions in the same week). Demonstrators are available for Prelims Computing marking (and advice!) during Trinity Term Weeks 1–7 on both Thursday and Friday from 10:00–13:00 and 14:00–17:00.

1 Introduction

You are asked to plot the trajectory of a rocket on a mission to land on the moon. You do not have to worry about the affect of the Earth's atmosphere or the changing mass of the rocket as it burns its fuel or loses its booster rockets. The only variables you will need to worry about are the initial velocity of the rocket and the launch angle relative to a chosen reference frame. Figure 1 shows the trajectory that is expected for a typical Earth-moon trip.

2 The physics

You will recall that the gravitational force between two bodies is given by

$$\mathbf{F}(\mathbf{r}) = -GM_1M_2\frac{1}{{\mathbf{r}_r}^2}\frac{\mathbf{r}}{|\mathbf{r}|} \tag{1}$$

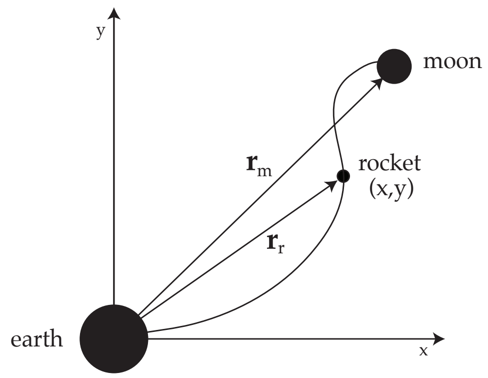


Figure 1: Trajectory for Earth-Moon trip.

In our system of a rocket in the gravitational fields of the Earth and moon we have

$$\mathbf{F}(\mathbf{r}) = -GM_eM_r\frac{1}{\mathbf{r}_r{}^2}\frac{\mathbf{r}}{|\mathbf{r}|} - GM_mM_r\frac{1}{(\mathbf{r}_r-\mathbf{r}_m)^2}\frac{\mathbf{r}_r-\mathbf{r}_m}{|\mathbf{r}_r-\mathbf{r}_m|} \tag{2}$$

where M_e, M_m, M_r are the masses of the earth moon and rocket and $\mathbf{r}_r$ and $\mathbf{r}_m$ are the distances of the rocket and the moon to the earth respectively. The acceleration of the rocket is given by

$$\mathbf{a} = \frac{\mathbf{F}(\mathbf{r})}{M_r} = \mathbf{F}(\mathbf{r}) = -GM_e\frac{1}{\mathbf{r}_r{}^2}\frac{\mathbf{r}}{|\mathbf{r}|} - GM_m\frac{1}{(\mathbf{r}_r-\mathbf{r}_m)^2}\frac{\mathbf{r}_r-\mathbf{r}_m}{|\mathbf{r}_r-\mathbf{r}_m|} \tag{3}$$

If we integrate the acceleration twice with respect to time we can find and plot the position of the rocket at any time on its journey to the moon.

There are many numerical methods of solving differential equations of this kind. We shall look at two of the simpler ones in this project.

3 Numerical method

The problem may be simply stated as: given a force $\mathbf{F}(\mathbf{r},\mathbf{v},t)$ and initial conditions for positions and velocities find the acceleration ($\mathbf{a} = \mathbf{F}/m$) and, from the acceleration determine the trajectory of the particle. We treat extended bodies as point masses i.e. particles.

3.1 Euler's method

The simplest method of solving a differential equation was devised by Euler in 1768.

Given

$$\tfrac{dx}{dt} \approx f(x_n, t_n) \qquad \text{with } x = x_0 \text{ when } t = 0,$$

then the value of x a small time h later is approximately

$$x_1 \approx x_0 + hf(x_0, 0)$$

or in general

$$x_{n+1} \approx x_n + hf(x_n, t_n) \qquad \text{where } t_n = n_h.$$

Alternatively

$$x_{\text{new}} \approx x_{\text{old}} + hf(x_{\text{old}}, t_{\text{old}}).$$

Thus the Euler method of deriving the rocket trajectory would be as follows:

1. Start at an initial position with an initial velocity.
2. Choose a step size Δt
3. Calculate the acceleration on the rocket from equation 3; you will need to resolve the $\mathbf{r}$ vector into $R(x, y)$ giving
$$a_x = -GM_e \frac{1}{|\mathbf{r}_r|^2} \frac{x_r}{|\mathbf{r}_r|} - GM_m \frac{1}{|\mathbf{r}_R - \mathbf{r}_m|^2} \frac{(x_r - x_m)}{|\mathbf{r}_r - \mathbf{r}_m|}$$
where $x_e = y_e = 0$. And similarly for a_y.
4. Use the Euler method to integrate $\mathbf{a}(\mathbf{r})$ to obtain the velocity $\mathbf{v}(\mathbf{r})$
$$v_x = v_x + \Delta t a(x)$$
$$v_y = v_y + \Delta t a(y)$$
5. Use the Euler method again to integrate $\mathbf{v}(\mathbf{r})$ calculated in step 4 to get the position
$$x = x + \Delta t v_x$$
$$y = y + \Delta t v_y$$
6. Increment the time in the calculation by $t = t + \Delta t$
7. Repeat the last four steps until the time reaches a defined end time or until the rocket either lands on the moon or falls back to earth.

This method would be exact if Δt were infinitesimal; since Δt is finite this method is only an approximation and is called a first order method since the error in the result is directly proportional to the step size you choose. To get an accurate result using this method you need to choose very small step sizes. This increases the computation time and also risks the accumulation of rounding errors in the computer.

In general you will never use a first order method for solving an ODE.

3.2 Improved Euler method

A way of improving the Euler method is instead of generating $x_n + 1$ by adding the rectangle $hf(x_n, t_n)$ to x_n, add the trapezium $\frac{1}{2}h[f(x_n, t_n) + f(x_{n+1}, t_{n+1})]$, i. e.

$$x_{n+1} = x_n + \tfrac{1}{2}h[f(x_n, t_n) + f(x_{n+1}, t_{n+1})].$$

Here however, the new value (x_{n+1}) appears on both sides of the equation; we want only old values on the right hand side. We can use the first approximation for $x_{n+1} = x_n + hf(x_n, t_n)$ to replace $f(x_{n+1}, t_{n+1})$ with $f(x_n + hf(x_n, t_n), t_{n+1})$ giving an explicit expression for x_{n+1} in terms of x_n:

$$x_{n+1} = x_n + \tfrac{1}{2}h[f(x_n, t_n) + f(x_n + hf(x_n, t_n), t_{n+1})].$$

In summary

Euler method	**Improved Euler method**
$c_1 = hf(x, t)$	$c_1 = hf(x, t)$
	$c_2 = hf(x + c_1, t + h)$
$x_{\text{new}} = x + c_1$	$x_{\text{new}} = x + \frac{1}{2}(c_1 + c_2)$

The improved Euler method is called a second order method because the error is now proportional to the square of the step size, if you halve the step size the error will decrease by 4. The improved Euler method allows us to use a larger step size than the simple method (thus reducing the computing time involved and reducing the risk of rounding errors) and still achieve greater accuracy.

To derive the rocket trajectory using the improved Euler method follow the steps on the previous page for the Euler method replacing steps 4 and 5 with

4. Use the Euler method to calculate a new (x', y'):

$$x' = x + \Delta t v_x$$

$$y' = y + \Delta t v_y$$

   and now calculate a new (v'_x, v'_y) based on (x, y) and (x', y')

$$v'_x = v_x + \Delta t \left(\frac{a_x(x, y) + a_x(x', y')}{2} \right)$$

$$v'_y = v_y + \Delta t \left(\frac{a_y(x, y) + a_y(x', y')}{2} \right).$$

5. Use the improved Euler method to calculate the new (x, y) coordinate from (v_x, v_y) and (v'_x, v'_y):

$$x' = x + \Delta t \left(\frac{v_x + v'_x}{2} \right)$$

$$y' = y + \Delta t \left(\frac{v_y + v'_y}{2} \right)$$

   and now update the values of vx and vy

$$v_x = v'_x$$

$$v_y = v'_y.$$

4 The problem

Since the distance from the earth to the moon is large in comparison to the radius of the moon, and the masses of the earth and moon are large it makes sense to choose a convenient system of units. Using meters, kilograms and seconds in our calculations may require numbers beyond the range of the computer. We will use the radius and mass of the moon as our units of length and mass, in these units:

M_M (mass of the moon)	1.0 moon-mass
M_E (mass of the earth)	83.3 moon-masses
R_M (radius of the moon)	1.0 moon-radius
R_E (radius of the earth)	3.7 moon-radii
G (gravitational constant)	9.63×10^{-7} moon-radii^3 moon-masses^{-1} s^{-1}

Write a function to simulate the rocket movement using the template given in appendix A.

4.1 The simple stationary case

Assume that the moon is stationary on the y-axis at a distance of 222 moon-radii from the earth. Use the Euler method outlined above to plot the trajectory of a rocket launched from the earth's surface on the y-axis (i.e. at co-ordinates (0, 3.7 moon-radii). Stop the calculation when the rocket lands on the moon (i.e. when the distance to the centre of the moon < radius of the moon).

Use $v_x = v_0 \cos(\theta)$ and $v_y = v_0 \sin(\theta)$ as your values for the initial velocities. You will immediately notice that there are two variables that determine the initial velocities, v_0, the initial velocity and θ the initial launch angle. These are critical for the launch of any rocket or satellite and depend upon the position on the earth the rocket is launched from. For this project we will launch the rocket from the y-axis and a valid set of (v_0, θ) is (0.0066 moon-radii/s, $(89.9\pi)/180.0$ rad). Choose a step size of $\Delta t = 10s$.

Now change the starting position of the rocket to the x-axis of your system i.e. at (3.7 moon-radii, 0). Using the system values as before find a (v_0, θ) that will land a rocket on the moon. You should use $v_0 = 0.0066$ moon-radii/s for simplicity.

Plot your trajectories in each case.

▶ **You may wish to discuss your results and the plot with a demonstrator before proceeding.**

4.2 An orbiting moon

Now that the simple case of a stationary moon has been solved find the initial angle required to land on the moon if the moon is orbiting the earth according to

$$\mathbf{r}_m = R_o[\cos(\Omega t), sin(\Omega t)] \tag{4}$$

where R_o is the mean moon-earth distance that was used in the previous problem (222 moon-radii) and $\Omega = 2.6615 \times 10^{-6}$ rad s^{-1} is the moon's angular frequency.

You should launch the rocket from the y-axis with the same v_0 as in the previous problem and use the improved Euler method to calculate the positions of the rocket. Plot the trajectory.

5 Preparing for assessment

When you have finished writing your code and have tested it completely, you should submit your code electronically to WebLearn as described in the Prelims Handbook. As noted (see above Preface), you must upload your code to WebLearn in the week before you come in to meet with a demonstrator for marking.

At marking time, be prepared with your written report (as described in the Handbook), your logbook (where you wrote extra notes during your program development), and be prepared for the demonstrator to download your code from WebLearn for you to describe it and demonstrate its execution.

A Functions to be implemented

```
function [tout, pos] = simulate_rocket(init_pos, init_vel, moon_pos, t)
% Author: ??? , Date: ??/??/????
% Simulate the rocket trajectory with the earth and moon influence. The coordinate
% used in this function is centred at earth's centre (i.e. earth centre at (0,0) )
% and scaled in moon-radius.
% The simulation finishes when it simulates for the whole t, or the rocket landed
% on the moon.
% Input:
% * init_pos: 2-elements vector (x, y) indicating the initial position of the rocket.
% * init_vel: 2-elements vector (vx, vy) of the initial velocity of the rocket.
% * moon_pos: a function that receives time, t, and return a 2-elements vector (x, y)
%             (see hint) indicating the moon position relative to earth.
% * t: an N-elements vector of the time step where the position of the rocket will be
%             returned.
%
% Output:
% * tout: an M-elements vector of the time step where the position is described,
%         if the rocket does not land on the moon, M = N.
% * pos: (M x 2) matrix indicating the positions of the rocket as function of time,
%        with the first column is x and the second column is y.
%
% Example use:
% >> init_pos = [0, 3.7];
% >> init_vel = 0.0066 * [cosd(89.9), sind(89.9)];
% >> moon_pos = @(t) [0, 222];
% >> t = linspace(0, 10000, 1000);
% >> [tout, pos] = simulate_rocket(init_pos, init_vel, moon_pos, t);
% >> plot(pos(:,1), pos(:,2));
end
```

B Hint: Lambda function

If you want to pass a function as an argument of another function, you need to define it as a 'lambda' or 'anonymous' function. A lambda function is defined by the syntax:

```
func_name = @(arg1, arg2, ...) ...
```

with `arg*` the arguments of the function. The example below shows how to define a lambda function that returns the square value of the argument, and how to call it.

```
>> fsquare = @(x) x.*x; % defining the function
>> a = fsquare(4); % calling the function
>> a
16
```

