Code user guide

# Folder

The code folder received will contain another 2 subfolders. Each subfolder corresponds to a case of certain boundary conditions type. One subfolder corresponds to Boundary conditions of: Vmag, Rmag, e, I, a, and tp(period). The other corresponds to final vector, and final r vector of the target orbit for instance. Let’s call the subfolder corresponding to Vmag, Rmag, e, I, a, and tp(period) boundary conditions subfolder 1. The other subfolder will be called subfolder 2.

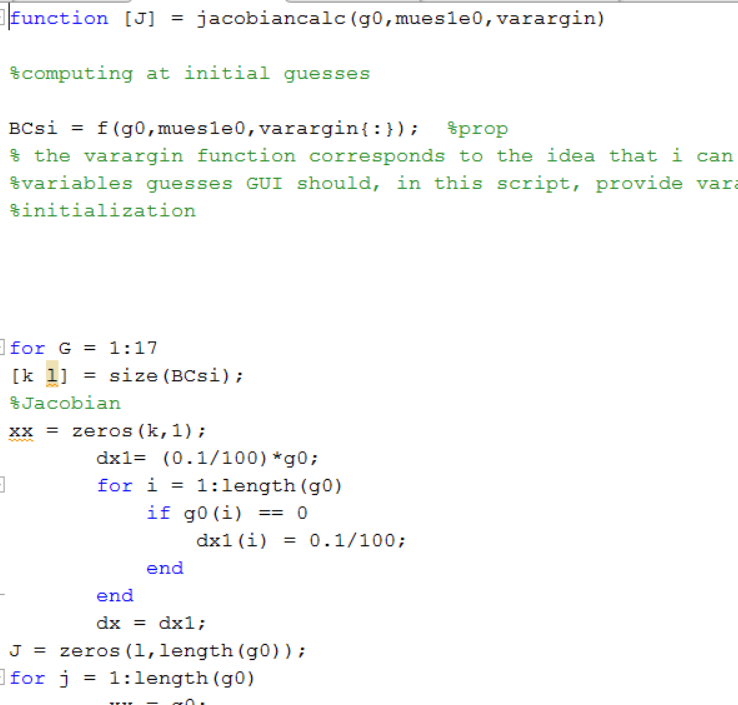
# Functions: Subfolder 1

Let’s start with the functions of the subfolder 1. We have the following functions:

* F
* Findmue
* Jacobiancalc
* Plot\_solve
* R\_V\_to\_parameters2
* Solve\_plot
* Trans
* Two\_body\_earth
* Two\_body\_earth

Jacobiancalc function:

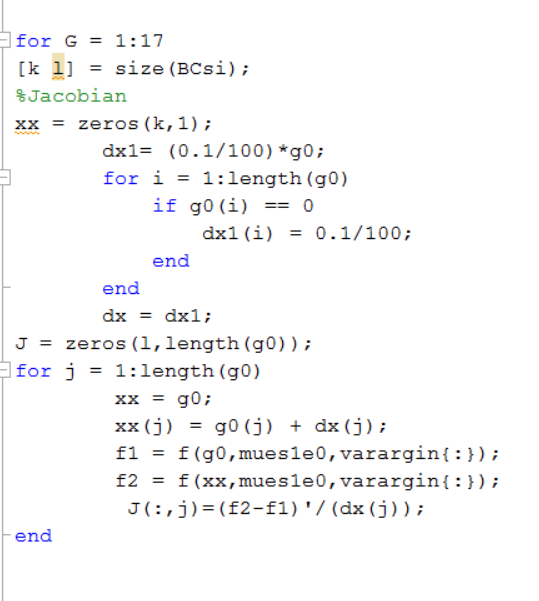
This function simply calculates the jacobian using the forward difference method.



It takes three inputs :

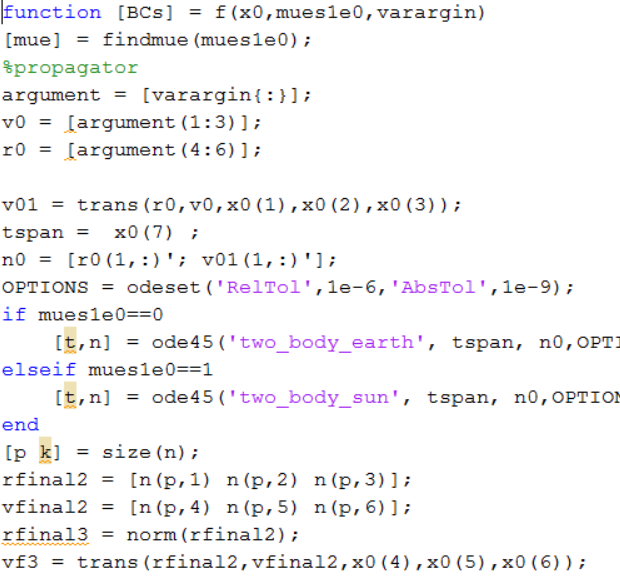
* g0: all initial guesses even the constrained ones
* mues1e0: a zero or a one depending on the user. If zero then we will use mue earth, if one, we will use mue sun.
* varargin: it is a matab operator that takes any number of inputs. Here the user will input r0 and v0 of the initial orbit.

As can be seen from the previous image, this function calls another function called f. We will explore it next

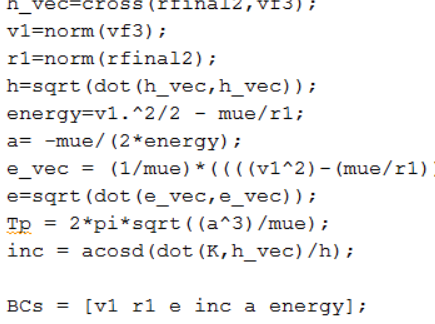
* 

We finaly calculate the jacobian as an output.

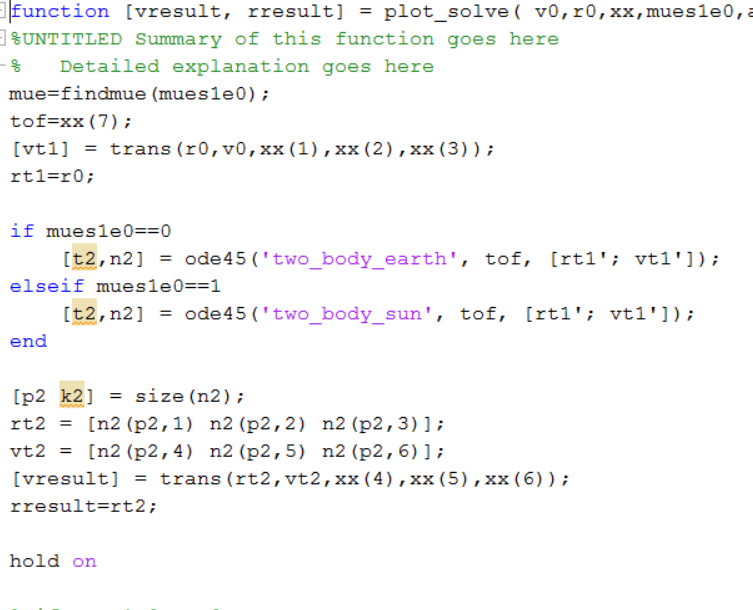
f function:



As we talked in jacobiancalc, x0 corresponds to the initial guesses, varargin corresponds to ro and vo and mues1e0 for the mue value. This function propagates on all the initials, even the constraints, and gets my r and v after the propagation. Next it calculates the elements we need for our residuals and boundary conditions of vmag, rmag, i, e, a, and energy.



Plot\_solve function:



This function takes my v0 and r0 of the initial orbit and the mues1e0 we explained before. The xx is the vector of each guess we used in our solve\_plot function that we will explain next. Briefly speaking, this function takes in those variables just mentioned and outputs a graph of this specific iteration. And, it outputs the graph of this specific iteration.

trans function:



This function takes in r0, v0, dvi, and the two angles in the VUW frame and then transforms them into eci and then adds them to the inputted velocity vector.

Findmue function:

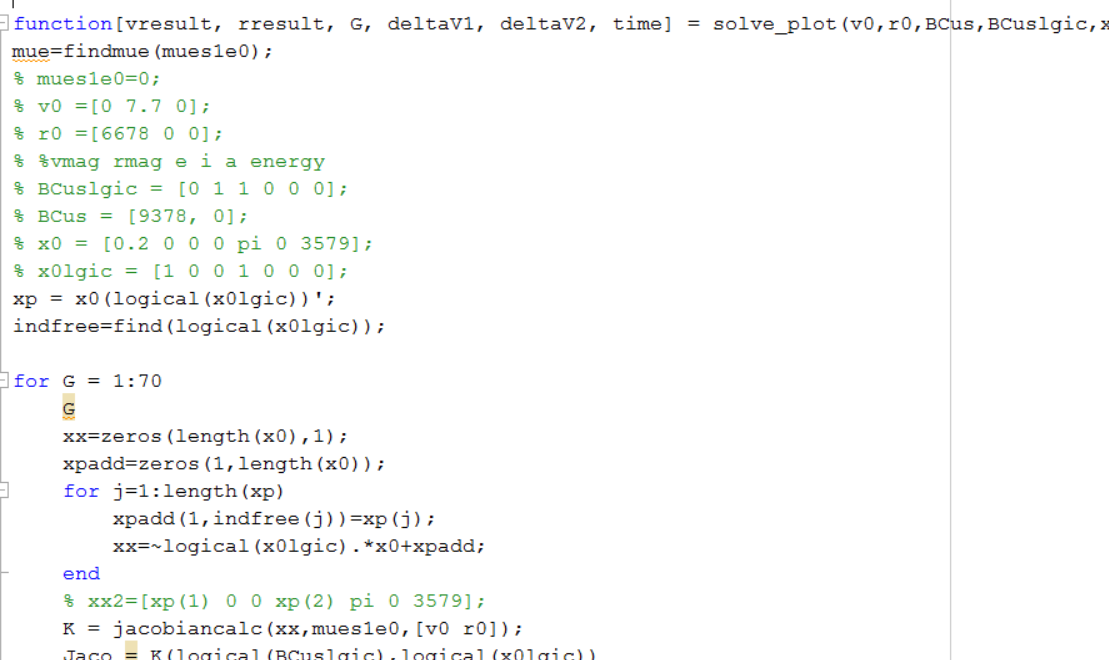
This function, throughout the code, changes the mue according to the user input. If the user input corresponded to 1 (checkbox in the gui), mue will equal mue sun. If zero, mue will correspond to mue earth.

Two body earth/sun function:

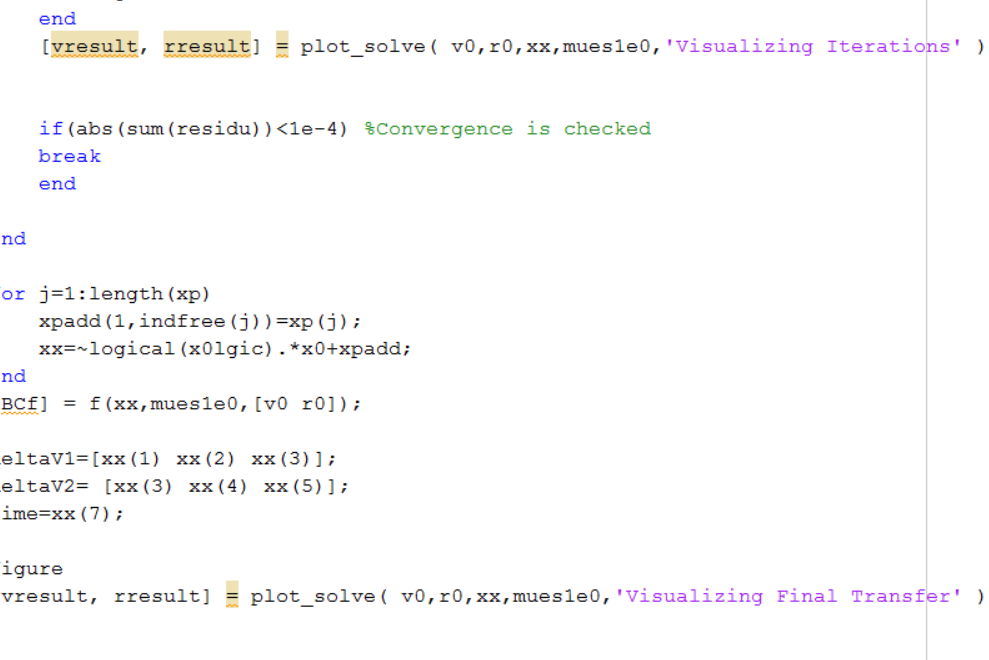
Corresponds to the findmue function weather it is sun or earth. This is the function will go to the ode 45. This function contains the equation of motion of the two body problem.

Solve\_plot function:

This is our main function in the code.



This function takes all the inputs from the gui directly. It takes the user BCs, the initial guesses, the v0 and r0 of the initial orbit, and their places that can map directly to the jacobian needed. In other words, with some gui configurations, the BCs and the initial guesses map inside this code to the jacobian I will calculate finally. That is why they are put in the GUI in a constrained order. This function outputs my final result after calculating and adding the jacobian and adding it to the initial and final r and v.



# Functions: Subfolder 1

In the second subfoder we have:

* F4
* Findmue
* Jacobiancalc4
* Plot\_solve4
* R\_V\_to\_parameters2
* Solve\_plot4
* Trans
* Two\_body\_earth
* Two\_body\_earth

The only changed thing is the boundary conditions of the residuals in the previous f function. And I only changed the name for differentiation.