Example to plot directly into latex

19-10-2019

1 Introduction

2 Horshoe Orbit

To find the initial conditions for the asteroid in a horseshoe orbit, one can initialise the orbit without a velocity, in the vicinity of L3. The coordinates of L3 for the given configuration, are: $[-1, 000, 0]$[1]. A time-span of 500000 iterations was used. This results in the following horseshoe orbit:

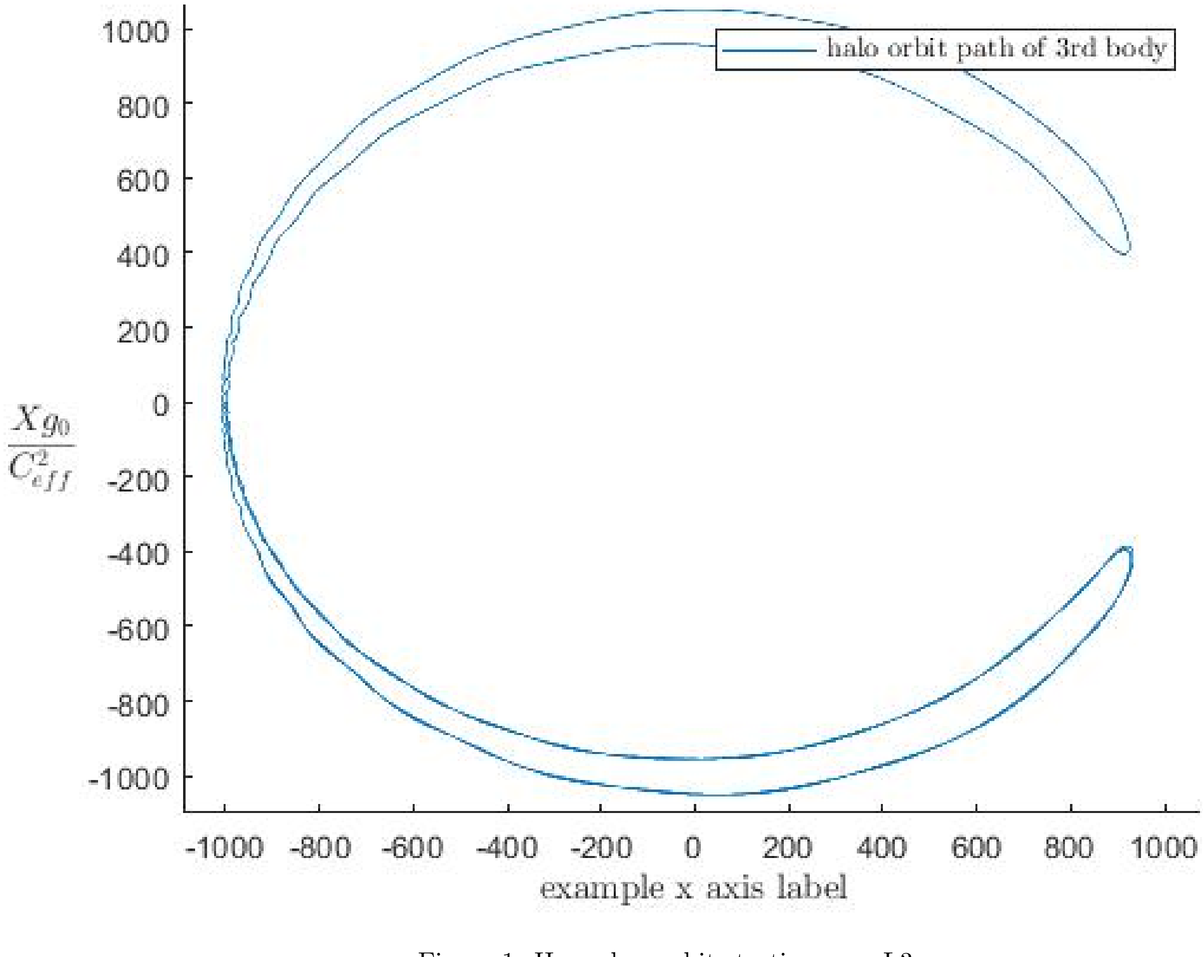

Figure 1: Horseshoe orbit starting near L3

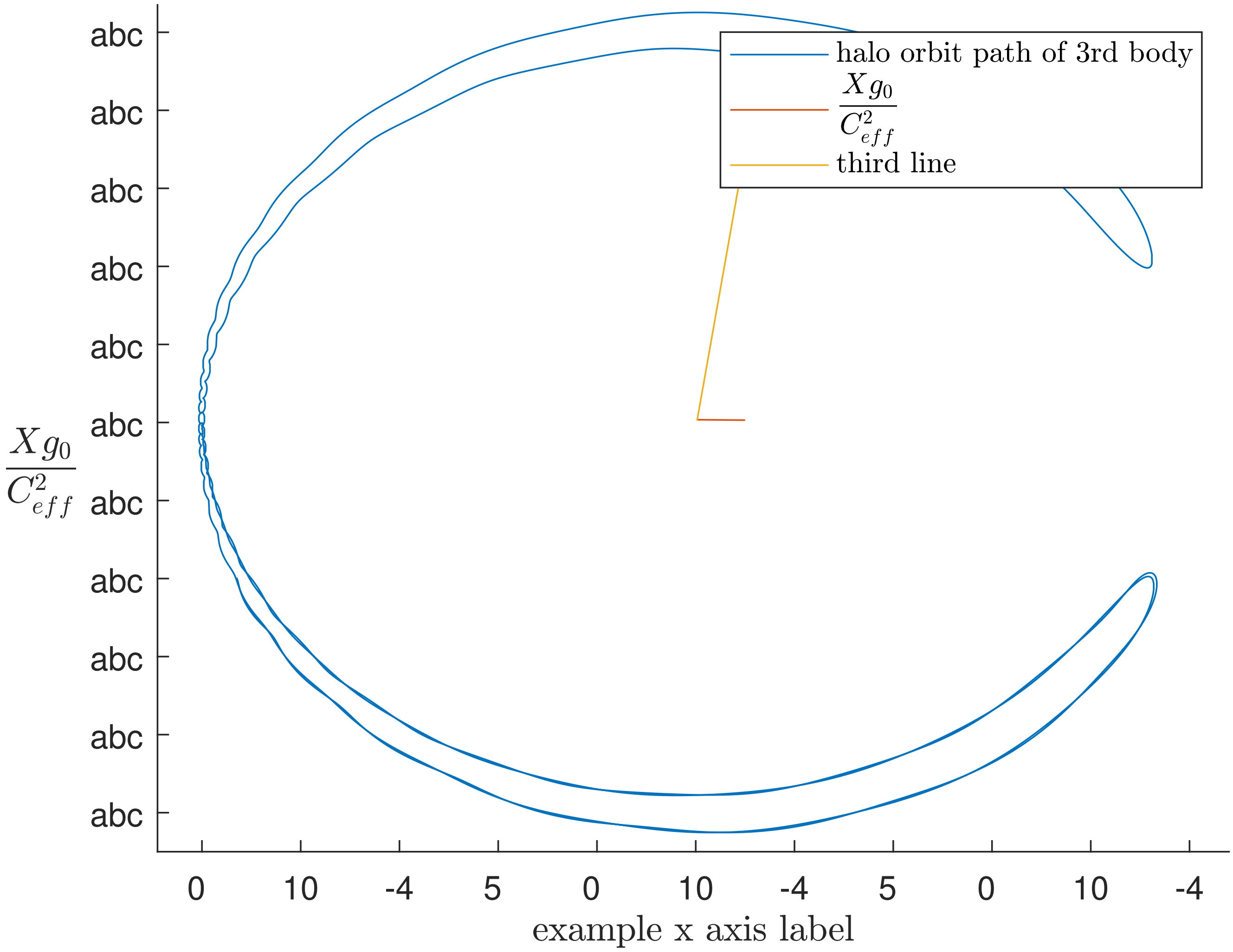

Figure 2: Figure consisting of multiple dataseries.

References

[1] Dr. Ir. E. Schrama. Lecture notes on planetary sciences and satellite orbit determination, September 2019.

Appendix 3: Code for Section D

```
%% clear console and data
clear all
close all
clc

% declare and initialise parameters
mp = 1000;
mq = 1;
alpha_p = [-1; 0];
alpha_q = [1000; 0];
G = 1;
mu_p = G * mp;
mu_q = G * mq;
d_p = 1;
d_q = 1000;
n = sqrt((mu_p + mq)/(d_p + d_q)^3);

% write down integration span
t_span = [0:1:500000];

% create initial state (a_1,a_1_dot,a_2,a_2_dot):
alpha_init = [-1001; 0; 0;0];

% call differential equations with ODE45
[t,alpha] = ode45(@(t,alpha) odefcn18_3(t,alpha,alpha_p,alpha_q,mu_p,mu_q,d_p,d_q,n), t_span, alpha_init);

%% plot to latex example
dataseries_1 = alpha(:,1);
dataseries_2 = alpha(:,3);


%% it configures and creates the plot
% Put this below your code (above your functions though)
% declare dataseries (in this case 3, can be more)
x_series = java.util.ArrayList(); %omg can has java in matlab
y_series = java.util.ArrayList();
z_series = java.util.ArrayList();

% declare axis scales
axisScales = java.util.ArrayList();

% declare axis domains
axis_domains = java.util.ArrayList();

% declare and initialise plot parameters
currentFolder = ”/code/”;
latexDestination =”latex/images/”;
fileName = 'plot_1d';
relativePath = '../latex/Images/'; % the ../ goes up one folder
exportType = 'eps'; % can be eps or jpeg
lineColours = 'blue';
nrOfDimensions = 2;
y_axis_label = '$\displaystyle\frac{X{g_0}}{C_{eff}^2}$';
axisLabels = [”example x axis label”, y_axis_label];

% set custom axis domains:
setAxisDomain = true;
x_axis_domain = [-1090 1080];
y_axis_domain = [-1100 1070];
```

```
axis_domains.add(x_axis_domain)
axis_domains.add(y_axis_domain);

% create custom axis scales
setCustomScales = true; % set to false to disable custom axis scales
x_axis_scale = [0,10,-4,5];
y_axis_scale = ['a';'b';'c'];
z_axis_scale = [0:1:10];
axisScales.add(x_axis_scale);
axisScales.add(y_axis_scale);
axisScales.add(z_axis_scale); % can also do this in loop

% create x-series (can be as much as you like)
x_series1 = dataseries_1;
x_series2 = [2,3,100];
x_series.add(x_series1);
x_series.add(x_series2);

% create y-series (can be as much as you like)
y_series1 = dataseries_2;
y_series2 = [6,7,6];
y_series3 = [6,7,700];
y_series.add(y_series1);
y_series.add(y_series2);
y_series.add(y_series3);

% put data series in java ArrayList() object
dataSeries = java.util.ArrayList();
dataSeries.add(x_series);
dataSeries.add(y_series);

% create legends for dataseries
y_series1_label = ”halo orbit path of 3rd body”;
y_series2_label = '$\displaystyle\frac{X{g_0}}{C_{eff}^2}$';
y_series3_label = ”third line”;

legend = [y_series1_label;y_series2_label;y_series3_label];
legendLocation = 'best'; % left doesn't work yet
plotType = ”lines”; % scatter doesnt work yet

% create plot object containing all info for plot
plotData = PlotData(fileName,relativePath,exportType,...
        dataSeries,lineColours, nrOfDimensions,axisLabels,legend,...
        legendLocation, plotType,axisScales,currentFolder,...
        latexDestination,setAxisDomain,axis_domains, setCustomScales);

% plot the dataseries automatically to latex
obj_mult = PlotMultipleLines;
plot_altitudes(obj_mult,plotData);

%% Create a quick 2nd figure:
x_series.clear(); % java
x_series.add(dataseries_1) % java
y_series.clear(); % java
y_series.add(dataseries_2) % java
filename = ”different_picture”
exportType = 'jpeg'; % can be eps or jpeg
setCustomScales = false; % set to false to disable custom axis scales

plotDataTwo = PlotData(fileName,relativePath,exportType,...
        dataSeries,lineColours, nrOfDimensions,axisLabels,legend,...
        legendLocation, plotType,axisScales,currentFolder,...
        latexDestination,setAxisDomain,axis_domains, setCustomScales);

plot_altitudes(obj_mult,plotDataTwo);

%% ODE equations
function dalphadt = odefcn18_3(t,alpha,alpha_p,alpha_q, mu_p,mu_q,d_p,d_q,n)
    % declare and initialise parameters
    dalphadt = zeros(4,1);
    r_p = sqrt((alpha(1)-alpha_p(1))^2+(alpha(3)-alpha_p(2))^2);
    r_q = sqrt((alpha(1)-alpha_q(1))^2+(alpha(3)-alpha_q(2))^2);

    % Implement ODE
    dalphadt(1) = alpha(2);
    dalphadt(2) = -mu_p*((alpha(1) + d_p)/(r_p)^3) - mu_q*((alpha(1) - d_q)/((r_q)^3)) + 2*n*alpha(4) + n^2*alpha(1);
    dalphadt(3) = alpha(4);
    dalphadt(4) = -mu_p*((alpha(3))/(r_p)^3) - mu_q*((alpha(3))/((r_q)^3)) - 2*n*alpha(2) + n^2*alpha(3);
end
```