

Student name: **Enter your name here**

Homework Assignment 11

Runge-Kutta Integration to Study Horseshoe and Tadpole Orbits of Asteroids

The goal of this homework assignment is to generate tadpole and horseshoe orbits of asteroids. In case you were not able to finish this week's computer lab and do not have a working Runge-Kutta code yet, please take a peek at the lab solution program that we will post to bCourses later this week.

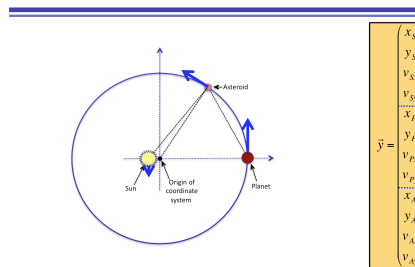
(1) Please start writing a new Matlab code that sets constants and initial positions. For this exercise, let us use the following masses $m_S=10^7$, $m_P=10^4$, and $m_A=1$. Set the gravitational constant $G=1$. Declare these constants global before you use them. The initial sun-planet distance should be $R=|r_S-r_P|=20$. Assign the initial positions $r_S=(x_S,0)$ and $r_P=(x_P,0)$ such that the center of mass

$$\vec{r}_{CM} = \frac{\vec{r}_P m_P + \vec{r}_S m_S}{m_P + m_S}$$

is located at the origin of the coordinate system. This can all be achieved by

- a) preliminarily setting $x_S=0$,
- b) working out x_P
- c) computing r_{CM} and then shifting the sun and the planet positions by $-r_{CM}$.

Initial velocity vectors



Now place the asteroid on the planet's circular trajectory but $\theta = 45^\circ$ ahead of it. The figure above shows the initial position given by $x=R \cdot \cos(\theta)$ and $y=R \cdot \sin(\theta)$

(2) Before we can set the initial velocities, we must work out the planet's angular velocity, ω , that also dictates the angular velocity of the asteroid. ω is obtained by equating the centrifugal force and the gravitational force (R is the distance from the sun to the planet):

$$\frac{Gm_1 m_2}{R^2} = \frac{\mu v^2}{R} = \mu \omega^2 R$$

$$\mu = \frac{m_1 m_2}{m_1 + m_2}$$

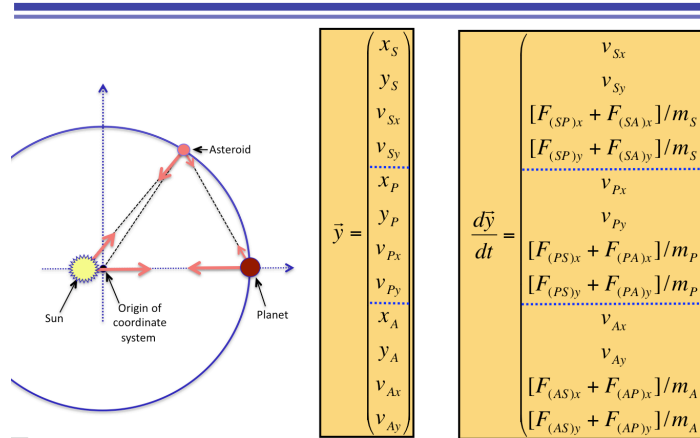
For each of the three objects, sun, planet, and asteroid, the modulus of the velocity vector is obtained from $|\vec{v}| = \omega |\vec{r}|$ where $|\vec{r}|$ is the object's distance from the center of mass. For the directions of the velocity vector please consult the figure above and make sure that \vec{r} and \vec{v} are at a right angle. (Putting in the velocity directions by hand avoids the cross product, $\vec{v} = \vec{\omega} \times \vec{r}$.) Now construct the initial vector \vec{y} with the twelve elements listed above.

(3) Now we need to write a Matlab functions that determines how the vector \vec{y} changes with time. Please generalize the case below with one object moving $\vec{y} = (x, y, v_x, v_y)$ to three objects.

```
function ydot = kepler_ode(t,y)
%obtain 'r' and 'v' from 'y'
r = ...
v = ...
% use the following three variables from the main worksheet
% must declare them 'global' here and there
global m M G
% Now put your formula for the force and acceleration here
F = ...
a = ...
% set the column 'ydot' that will be returned to the calling routine
ydot = ...
```

Please note that you will need to compute six forces shown here:

Set up of the vector y and its time derivative



(4) Now combine your Runge-Kutta code from our computer lab 11 with your new routine to compute $d\vec{y}/dt$. Determine the time, T , that it takes the planet to orbit the sun once from Kepler's third law:

$$T^2 = \frac{4\pi^2}{G(m_1+m_2)} a^3,$$

where a is the semi-major axis, that is identical to R for circular orbits.

Adjust dt so that you use 1000 time steps per orbit. Now run your code from $t=0$ until $t=T/4$ and plot the trajectories of all three objects. Which trajectory will appear as a point? How can you distinguish between the planet and the asteroid in your plot? Enter your plot here. Congratulations! The most difficult part has been accomplished.

(5) Now run your code for 20 full orbits. If you again plot trajectory of the planet and asteroid, you will see that they are hard to tell apart. Let us switch into a rotating frame where the planet is at rest. At every time step, let us perform a transformation of the asteroid's coordinates. Then we combine all transformed coordinates to plot the asteroid's trajectory in the frame of orbiting planet, which will yield the tadpole and horseshoe orbits.

At every step, we need to determine the angle, θ , in the orbit the planet is currently at. θ follows from inverting

$$\begin{aligned} \theta &= \arctan(y_P/x_P) && \text{if } x_P > 0 \text{ and} \\ \theta &= \arctan(y_P/x_P) + 180^\circ && \text{if } x_P < 0. \end{aligned}$$

As you can see, 180° needs to be added for $x_p < 0$. The transformation for asteroid coordinate is

$$\vec{r}'_A = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix} \cdot \vec{r}_A$$

Now see you get a tadpole orbit when you plot \vec{r}'_A .

(6) From now on, we only want to modify the initial conditions to see what kind of orbit we get. Increase the simulation time to 100 orbits. Change the initial angle of 45° in any way you like and see what kind of orbits you get. Enter two plots of reasonably distinct asteroid orbits in the rotating frame and specify the initial angle in each case. Which angle yields the smallest tadpole (and is therefore closest to the Lagrange point L_4)?

Submit you your two Matlab files that produced one to of the tadpole orbits.

The rest is optional:

(7: extra 20%) Now let us again use the original angle of 45° and the distance R but increase the initial speed in steps of 0.5%. At what percentage do you lose the asteroid? Include again two reasonably distinct plots and specify the parameters.

(8: extra 20%) Finally, we want to make some horseshoe orbits. Place the asteroid at distance R from the center of mass at an angle of 180° , which is exactly across from the planet. Increase its velocity in steps of 0.5% and compare the resulting orbits. Include two plots and describe what trend you observed with increasing velocity.