Name: _____ , _____

PHYSICS 401 : SPRING SEMESTER 2018

Project #3: Symplectic integrators and on-screen plottings

Please note: Answer questions, do not just hand in graphs without stating which part of the assignment you are doing.

Reference Reading: 5.1-5.3.

Downloads:

1) Go to http://www.compadre.org/portal/items/detail.cfm?ID=7147, click "download 12084kb.zip". Unzip this and it will create a folder, `workspace_compadre`, at your unzipping location. Let this location be where you will be doing most of your Java programming, with a name, say, `javawork`.

2) Go into `workspace_compadre\osp\src` and COPY the entire folder `org` and paste it back into the folder `javawork`. The `org` folder contains the OpenSourcePhysics Java source codes.

3) Download `PlotApp.java` from my website to the same folder `javawork`.

4) Now compile and run `PlotApp`. (You will get an error message if you do not have the folder `org` in the same location as `PlotApp.java`.)

Congratulation! You have produced your first on-screen plot!

1. Examine `PlotApp.java` and see how this on-screen plot is produced. The program creates a `PlotFrame` object, called `frame`, just like we created a `MovingObject` object called `planet`. `frame` has three labels, the x-axis label"position", the y-axis label "amplitude", and the title "Plot". `PlotFrame` is imported from the `org` folder. There's a for-loop which appends the two functions onto `frame` and the two functions are plotted when we make `frame` visible.

a) The two graphs are produced by `frame.append(0, ...)` and `frame.append(1, ...)`. How would you insert two more graphs $\exp(-0.5*x*x)$ and $\exp(-0.5*x)*x$? (You need to close the plotting window, modify and save a new copy of `PlotApp.java` recompile and re-run `PlotApp`).

b) Uncomment all the `frame.set....` attributes, recompile and re-run `PlotApp` and see how these attributes modify the default plot. Chose the attributes of your liking, recompile and re-run `PlotApp`.

c) On the plotting window, click on the File menu, print it out directly, or save the plot in a format (eps or jpg) that can be inserted into your assignment. Hand in this plot.

2. Now comment out all attributes, keep only two functions and save `PlotApp.java` as `KeplerMovePlotApp.java`. (You also need to rename the class name inside to `KeplerMovePlotApp` so that they match.) The idea is replace the function evalutaions in the for-loop, by the Kepler trajectories produced in Project 2. Very, very carefully, take the `KeplerMoveApp.java` file you have developed in Project 2.3, and move all the steps essential for producing the trajectories into `KeplerMovePlotApp.java`. You don't need the `try{ and catch{...` stuff and statements for writing a text file. Instead of the two function evaluations, replace them in the for-loop by

```
frame.append(0,planet.x, planet.y);
frame.append(1,asteroid.x, asteroid.y);
```

(You must have first compiled `MovingObject.java` and have it in the same location `javawork` with `org` and `KeplerMovePlotApp.java`.) Run the for-loop for 5 orbits as in Project2, planet uses `sym2astep` and asteroid uses `RK2step`. Hand in this plot.

3. Create another `PlotFrame` object say, `eframe`, for plotting the energy-error of both as a function of t/P. When compiled and ran, you will see both plotting windows simultaneously on-screen. Be sure to do everything for `eframe` as for `frame`. Hand in the energy error plot.

4. Implement RK4 and the symplectic integrator FR4 and hand in a plot of their energy-error. Run only one period so that you can see the structure of the energy error up-close. **Optional**: Can you output only the error-plot from $t/P = 0.45$ to $t/P = 0.55$?