ASTRONOMY 9602
COMPUTER PROJECT #1

Mohammed Chamma
250887035
March 1st 2017

The Figure-8

All simulations show here are done with “Gravity Lab”, a C++ program written by Piet Hut and Jun Makino.

The simulations here all span **10 seconds** and are plotted with snapshots taken approximately every **0.01s**.

The first simulation done is the figure 8 motion, using the default initial conditions distributed with the program. These initial conditions are shown in Table I and the units are arbitrary.

Object	x	y	z	vx	vy	vz
1	0.9700436	-0.24308753	0	0.466203685	0.43236573	0
2	-0.9700436	0.24308753	0	0.466203685	0.43236573	0
3	0	0	0	-0.93240737	-0.86473146	0

Table I: Figure 8 initial conditions. The masses of each object are equal and are set to 1

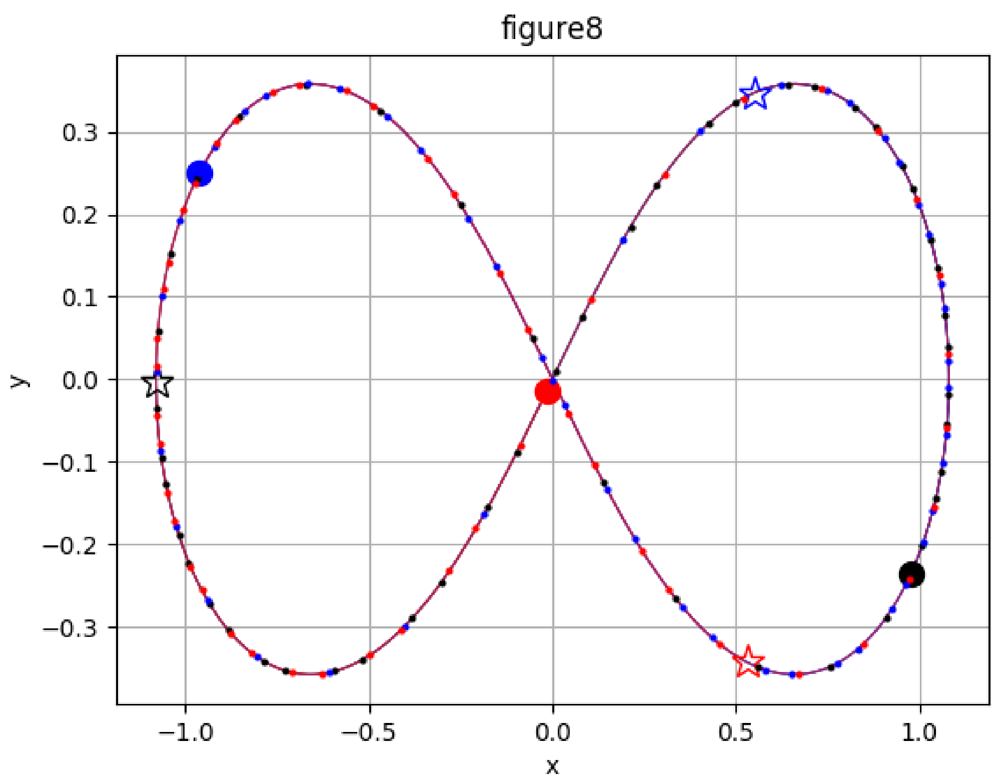

Figure 1: Figure 8 motion, default initial conditions. Large circles denote the starting position. Stars denote the final position of the object. Small dots are spaced at equal time intervals (roughly every 20 points) and give information on the speed of the object.

The plot shown above and those to come have the following features to glean certain information:

- Starting positions are denoted with a large circle
- Final positions are denoted with a star symbol
- The thin solid line denotes the path of the object
- The small dots along the path of an object are spaced at equal time intervals. Dots close together indicate low velocity and dots far apart denote high velocity
- When only three objects are present, their paths are colored black, blue and red.

From Figure 1 we can see that the objects start roughly along a line, with the red object in the center. Based on the initial velocity vectors, the blue and black dots both initially move to the left and upward along the figure-8 curve, while the central object goes downwards and to the left. The objects are chasing each other, and the paths perfectly overlap.

Based on the spacing of the dots the objects never slow down considerably but they do move slower around the bends on either side of the eight and a little quicker when crossing the central region.

Chaos

I made several changes (some small changes, others larger) and plotted the results. This gives us an idea of how stable the figure 8 pattern is. Table II shows all the initial conditions used, varied from the figure-8 initial conditions, and the changes are bolded and italicized. Figure 2 and 3 show the results.

m	**x**	**y**	**z**	**vx**	**vy**	**vz**
chaos: small change in a single x position						
1	***0.9800436***	-0.24308753	0	0.466203685	0.43236573	0
1	-0.9700436	0.24308753	0	0.466203685	0.43236573	0
1	0	0	0	-0.93240737	-0.86473146	0
chaos2: central object placed near right-most object						
1	0.9700436	-0.24308753	0	0.466203685	0.43236573	0
1	-0.9700436	0.24308753	0	0.466203685	0.43236573	0
1	***1***	0	0	-0.93240737	-0.86473146	0
chaos3: double all masses						
2	0.9700436	-0.24308753	0	0.466203685	0.43236573	0
2	-0.9700436	0.24308753	0	0.466203685	0.43236573	0
2	0	0	0	-0.93240737	-0.86473146	0
chaos4: small change in mass of one object						
1	0.9700436	-0.24308753	0	0.466203685	0.43236573	0
1.1	-0.9700436	0.24308753	0	0.466203685	0.43236573	0
1	0	0	0	-0.93240737	-0.86473146	0
chaos5: small, identical change in x of wing objects						
1	***1***	-0.24308753	0	0.466203685	0.43236573	0
1	***-1***	0.24308753	0	0.466203685	0.43236573	0
1	0	0	0	-0.93240737	-0.86473146	0

Table II: Initial conditions for the chaos simulations.

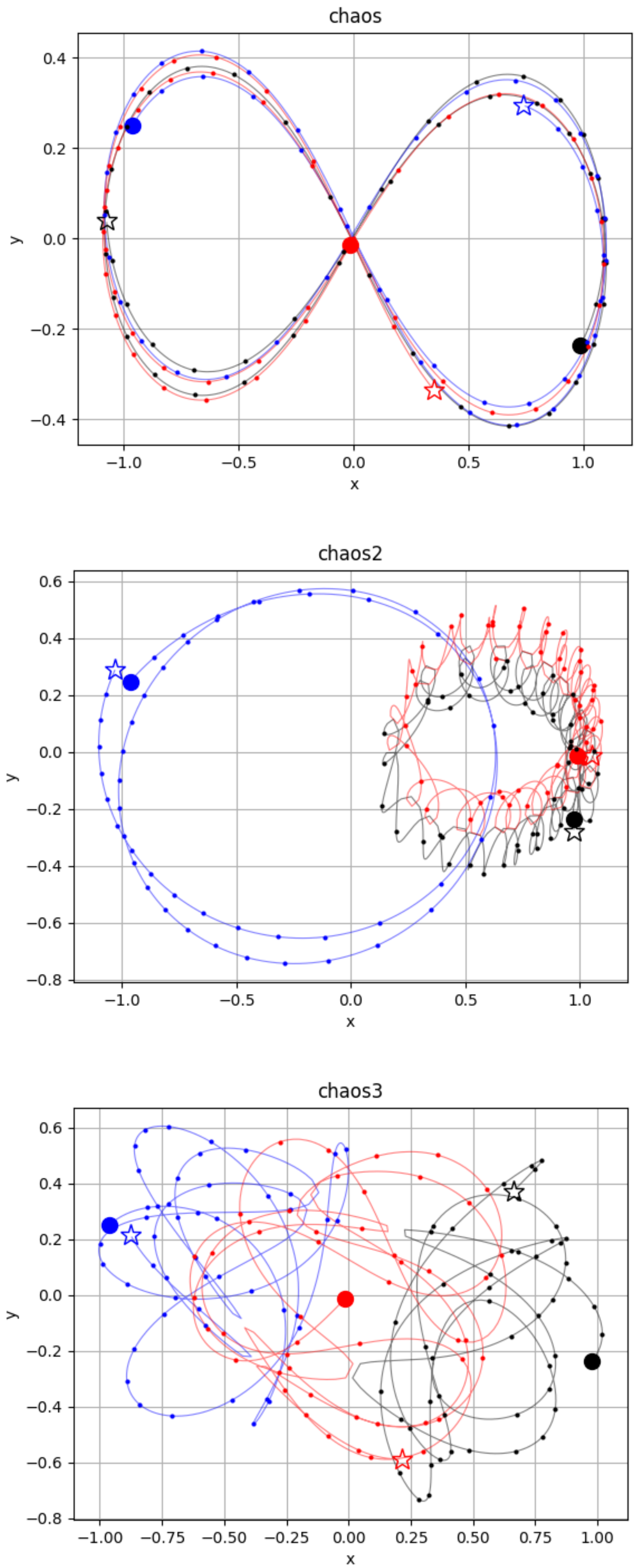

Figure 2: Chaos

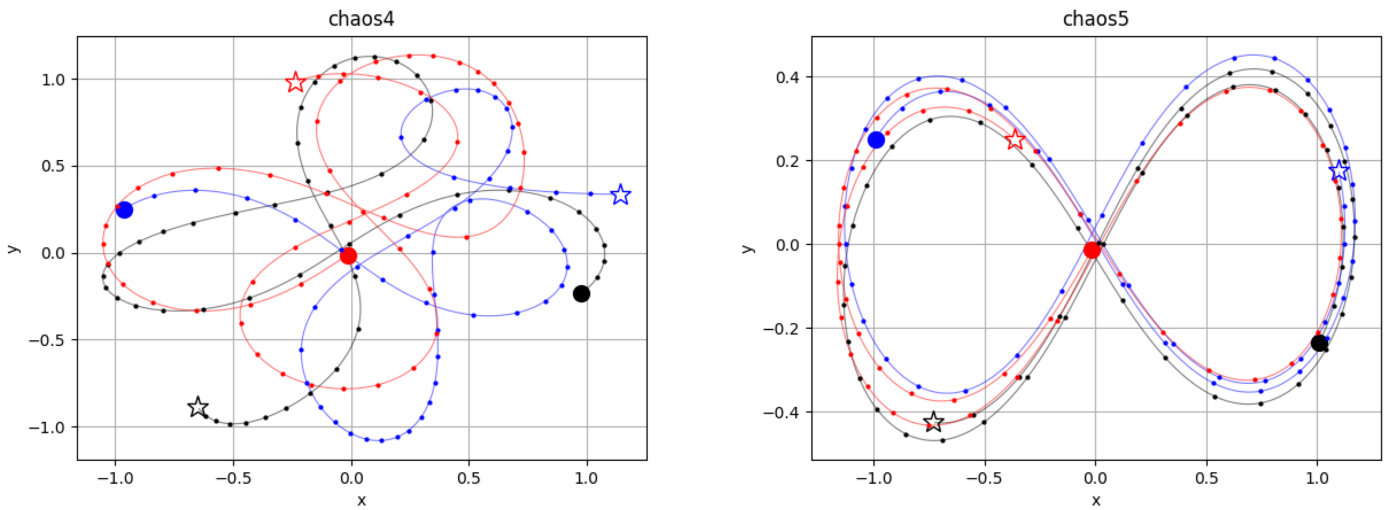

Figure 3: Chaos

We see that in the case of 'chaos' and 'chaos5', small changes in the x-position distort the figure 8 pattern but do not destroy it. Setting the x-positions of the first two objects to 1.9 and -1.9 (a large change in position) leads to a completely unbound system, shown in Figure 4. If we move the central object closer to one of the other objects (the right-most one in 'chaos2'), then we get this messy bound system with noisy orbits. The object far away has a large circularized orbit while the two objects close to each other orbit in a sort of binary while also orbitting around the center of mass of the entire system.

Another relatively large change we can make is to simply double all the masses, in the case of 'chaos3'. The result is a bunch of scribbles.

Finally, just a small change in the mass of one of the objects completely ruins the pattern as well, as in 'chaos4'.

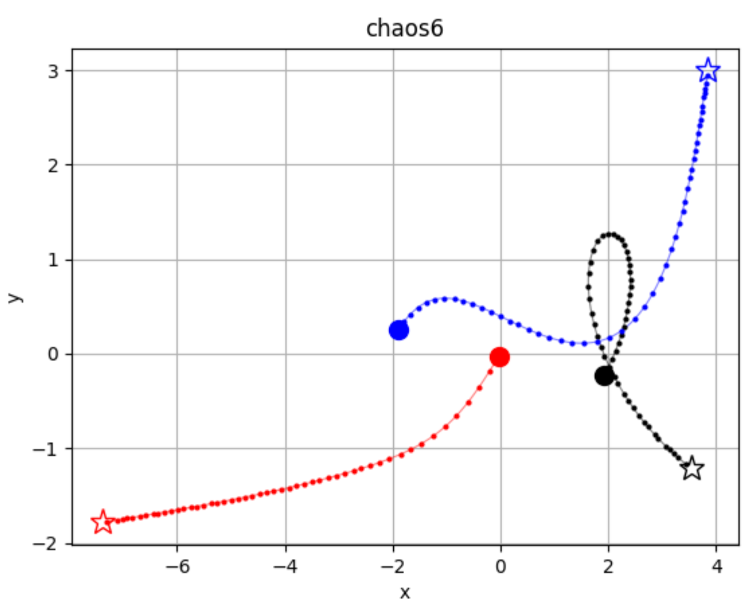

Figure 4: Unbound system when x = 1.9, -1.9 for the two wing objects

Ejection

The large changes made in the previous section left the system bound. Here are two cases where one of the objects is ejected while the other two objects continue to orbit each other. The initial conditions are shown in Table III, and the results are shown in Figure 5.

m	x	y	z	vx	vy	vz
flungout						
2	0.9700436	-0.24308753	0	0.466203685	0.43236573	0
1	-0.9700436	0.24308753	0	0.466203685	0.43236573	0
2	0	0	0	-0.93240737	-0.86473146	0
flungout2						
1	0.9700436	-0.24308753	0	0.466203685	0.43236573	0
2	-0.9700436	0.24308753	0	0.466203685	0.43236573	0
1	0	0	0	-0.93240737	-0.86473146	0

Table III: Initial Conditions for Ejection

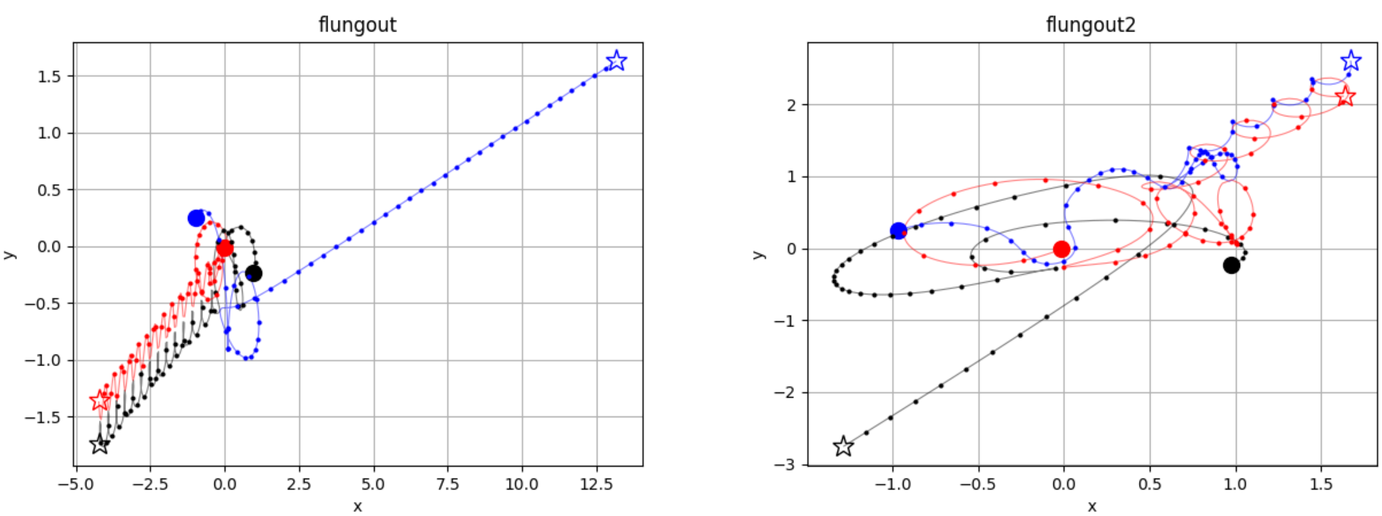

Figure 5: Objects being ejected. Notice the two bound objects pushing each other along after ejecting their companion.

In both cases, just the masses of the object were changed. The 'flungout' case doubled the mass of the blue and red object and the 'flungout2' case doubled just the mass of the black object. The blue object gets ejected in 'flungout' and the black object is ejected in the 'flungout2' case.

Notice in the 'flungout' case the black and red star take an oscillating path together (like a sine wave, as opposed to a closed orbit) and move together further and further away from their starting position. This is counter-intuitive to me and I'm not sure if it's an artifact of the simulation or something physical. One way to rule out whether or not it is an artifact is to plot the energy of the system over time, data that the simulation outputs as an error diagnostic. If it fluctuates wildly in time something nonphysical is probably happening. Another day.

N-Body

It was early in the morning and I was having trouble plotting a simulation of anything with more than 5 bodies because my plotting code wasn't very versatile and I had no energy to fix it. I wanted to make a configuration that resembled a four pointed 'star' with a mass at the center, with initial velocities such that the four pointed star was rotating counter-clockwise. I initially had different masses and different velocities and was getting interesting shapes but overall messy orbits (things like 0.8 for the velocities, 1.1 for the central mass, etc.). I accidentally set all the initial values to 1 like in Table IV and found a closed orbit, shown in Figure 6. Notice the central object has the same mass as the other objects but doesn't move at all. Exciting! And Bedtime.

m	**x**	**y**	**z**	**vx**	**vy**	**vz**
Star!						
1	0	0	0	0	0	0
1	1	0	0	0	1	0
1	0	1	0	-1	0	0
1	-1	0	0	0	-1	0
1	0	-1	0	1	0	0

Table IV: Initial configuration for a star 'rotating' counter-clockwise and with a mass at the center

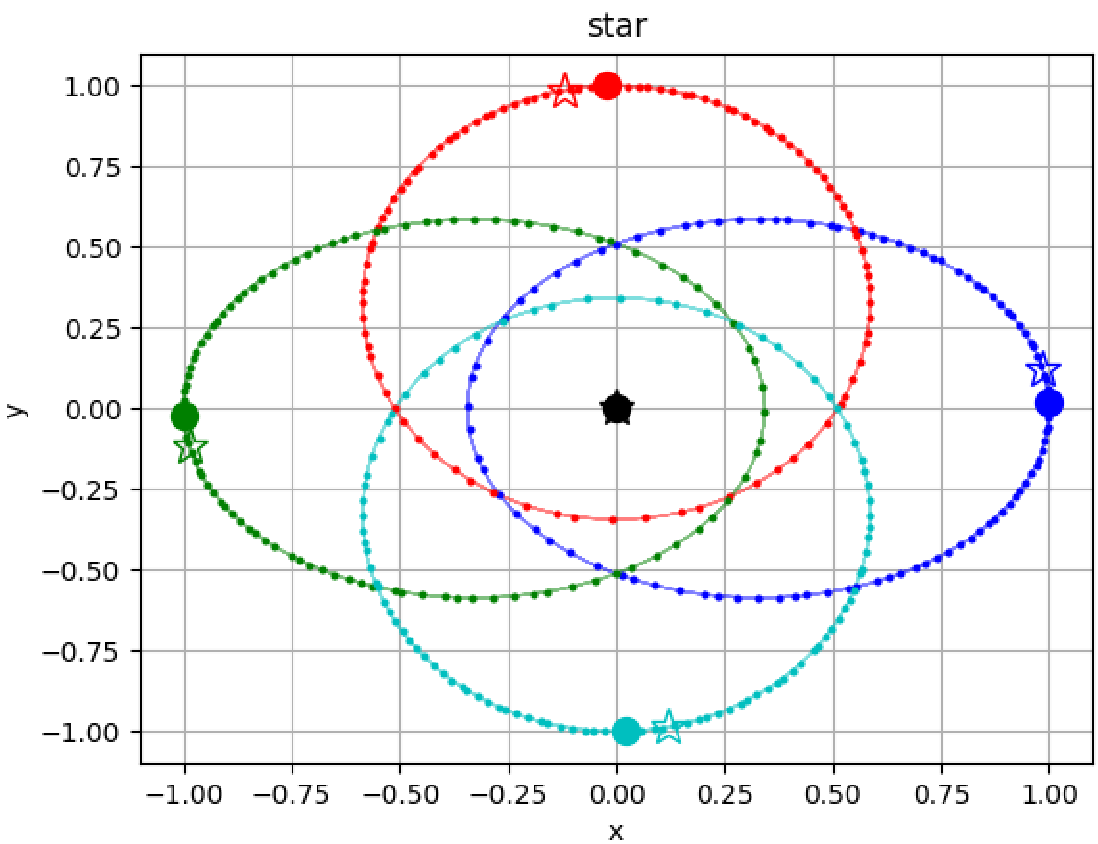

Figure 6: Closed Orbit of 5 Bodies

Code

All code is available at https://github.com/mef51/galacticnbody .

I used Python. I wrote a parser in a file called parsenbody.py which reads the data in output file into some python objects. The second file nbodyplots.py then uses that data to make all the plots in one go. Copies of both are attached at the end.