MATH299M/CMSC389W – *Visualization Through Mathematica*

Spring 2019 – Ajeet Gary, Devan Tamot, Vlad Dobrin

Model H7.2: Gravitational Field populated by several bodies

Assigned: Friday March 15th, 2019

Due: Monday April 1st, 2019 11:59PM

Note: Between models H7.1, H7.2, H8.1, H8.2, H9.1, and H9.2 (Group 2) you need only complete 3 assignments.

For this assignment you can use either ContourPlot or DensityPlot. A *Gravitational field* is a vector field caused by the force of gravity exerted by different bodies. What this means is that if you have a body in space, like a planet, then everywhere else in space is the domain of a vector-value function that describes the direction and magnitude of the gravitational field at that point. The *Gravitational Force* equation you may remember from high school physics is:

$$F = G\frac{m_1 m_2}{r^2}$$

where F is the force, G is the *Gravitational Constant* 6.674 x 10^{-11} m^3 kg^{-1} s^{-2}, m_1 is the mass of the first body, m_2 is the mass of the second body, and r is the distance between the bodies. If you were to model the magnitude of the gravitational force exerted by a body in the plane, you would be doing so from the perspective of some mass m_2 existing at that point in the plane, so m_2 would be a constant. As for the distance between them, assuming that the position of the first mass is constant, then the distance between any point (x,y) and mass 1's position (x_1,y_1) is of course modeled by the Euclidian distance function:

$$dist\big((x_1, y_1), (x_2, y_2)\big) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

G is also a constant. Since we would just be looking at the magnitude of this force, this is all we need to do in the case of one body. Making a ContourPlot or DensityPlot for the magnitude of the gravitational force in the plane exerted by a single body with fixed location and mass shouldn't be hard – you could even add 2D Sliders in a Manipulate command to give the user control of the bodies.

Now, if you wanted multiple bodies, it's not as simple as making the first argument of the Contour/DensityPlot a sum of the magnitudes of the forces exerted by each body. The bodies exert gravitational forces in a certain direction, namely the direction of the body. So, for each point (x,y) in the plane for each body we would need to use the magnitude of the force calculated above multiplied by a 2-vector, the difference vector of the position of the body minus (x,y), so that we have a weighted vector in the correct direction – *then* we add all of those vectors together to get a resultant vector and take the magnitude of *that*. This is important – for example if you're at the point (0,0) and there are bodies of equal mass at (-1,0) and (1,0), if you just add the magnitudes of the gravitational force from each then your plot will show that this magnitude is greatest right were you're standing on the y-axis, however the *opposite* is actually true, in fact along the y-axis the gravitational pulls of these two bodies would completely cancel out.

You can use Show to superimpose a Graphics onto your plot so that you can draw points where the bodies are. It should be obvious from the coloring where the bodies are, so this isn't necessary. Also it may slow down your model.

Here are some cool extra things you could do if you if you like this assignment so far:

Lagrangian Points
Once your model is working, it would be cool to try using actual values, that is, make two bodies, the Earth and Sun, with the masses and distances you find online, and set the mass m_2 to that of a human or a satellite (this number doesn't actually matter – the shading on the plot will use relative magnitude, so if you scale the value on the whole plot by m_2, which is what you're doing since it's in every sum in the sum of forces, it should have no effect on the coloring). Then you'll hopefully be able to spot the *Lagrangian Points*, points for which the pull of gravity of two bodies completely cancels out, meaning that theoretically a satellite placed exactly at that point would stay totally still. Extremely interestingly, the Sun-Earth system has FIVE such Lagrangian Points! Check it out on Wikipedia: https://en.wikipedia.org/wiki/Lagrangian_point
Seeing this would be *awesome.*

Supermassive Black Hole (SMBH)
Add a black hole to see how much it dominates the system! You could look up the mass of the supermassive black hole at the center of the Milky Way Galaxy, Sagittarius A^* (https://en.wikipedia.org/wiki/Sagittarius_A*), and place it as an object at the proper distance away from Earth and see how close to the Sun you need to be for Sagittarius A^* not to overpower the Sun. How would you know when you get to this point if you're just using magnitude in your model?

Zero Gravity in a Shell
If you really want to push your model, you can use the function CirclePoints which takes one positive integer argument n to generate the (x,y) coordinates of n equally spaced points around the unit circle. Multiplying this set of points by some scalar r will make them n equally spaced points around the circle of radius r centered at the origin. If this is your list of bodies, then something interesting happens inside the circle. If you hollow out the Earth and then go inside the empty shell, you become *weightless*, because the gravity from the surrounding shell in all directions *completely* cancels out at *all* points inside the shell (this is not obvious)! With your model you can test if this works in the 2D case by making n quite large and forcing your model to use the vector sum of all the forces exerted by all n of those bodies. This would be *really cool* to see! It would ideally have no color anywhere inside the circle as the circle points approach a full circular perimeter.

Arbitrarily Many Bodies Sandbox
The coolest way to do this project would be if you could move around the bodies on the plot with Locators, but we don't cover that until Week 10, which you're welcome to look ahead to once I post the notes. LocatorAutoCreate allows the user to place arbitrarily many such bodies. Keep this in mind – when we get to Week 10 I'm going to ask you to come back to this model and update it with this awesome feature!