Computational Physics: Assignment 2

Sarah Roberts

November 13, 2018

Part 1. This portion of the assignment was written in C#. The user is asked to input values for $M1$, $M2$, and the distance (d) between them. If the user enters a value for $M2$ that is greater than the value for $M1$, the program with reverse them. In the internal coordinate frame, $M1$ is located at the origin, and $M2$ is located at $(d, 0)$.

The program then calculates the net force on an mass along a grid defined for $x, y \in [-2, 2]$. The effective gravity is returned as vectors at each point, with the x components written to `ax.txt` and the y components written to `ay.txt`. Then, the potential at each point is calculated, and this is written to `potentials.txt`. Note that the path to these files in hard-coded in the main program and each path is defined separately.

The effective gravity and potential are calculated using

$$\mathbf{g} = \frac{Gm_1}{r_1^2}\mathbf{r_1} + \frac{Gm_2}{r_2^2}\mathbf{r_2}$$

$$\Phi = \frac{Gm_1}{r_1} + \frac{Gm_2}{r_2}$$

where $\mathbf{r_1} = x_1/\sqrt{x_1^2 + y_1^2}$ and $\mathbf{r_2} = (x_2 - d)/\sqrt{(x_2 - d)^2 + y_2^2}$.

Part 2. Using the data files from Part 1, I generated vector and contour plots with Python for the cases:

(1) $m_1 = 3$, $m_2 = 1$, and $d = 1$
(2) $m_1 = 100$, $m_2 = 1$, and $d = 1$.

These plots can be found in `2a.png` and `2b.png`. The Python script reads files `ax.txt, ay.txt,` and `potentials.txt`. The path to these files is hard-coded in the script.

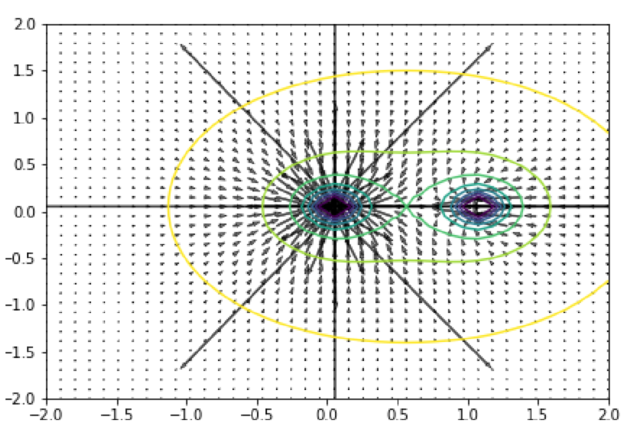

FIGURE 1. `2a.png` This shows the potential as contours and acceleration due to gravity as vectors for $m_1 = 3$, $m_2 = 1$, and $d = 1$.

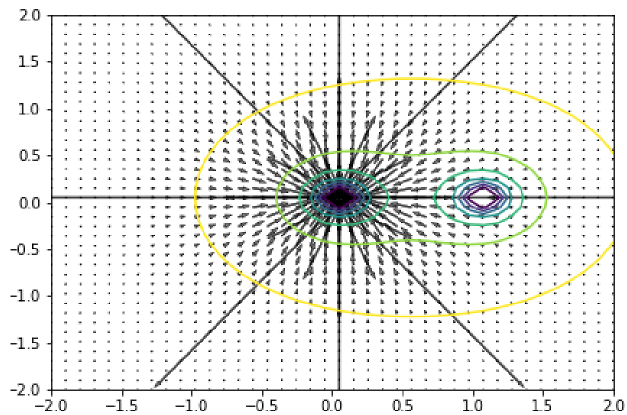

FIGURE 2. `2b.png` This shows the potential as contours and acceleration due to gravity as vectors for $m_1 = 100$, $m_2 = 1$, and $d = 1$.

Part 3. For Part 3 and 4, I changed my coordinate system so that the origin is the two body system's center of mass. This makes equations (1) and (2) on the assignment sheet valid.

Using the equation

$$g = \frac{-m_1}{|\mathbf{r} - \mathbf{r_1}|^3}(\mathbf{r} - \mathbf{r_1}) - \frac{m_2}{|\mathbf{r} - \mathbf{r_2}|^3}(\mathbf{r} - \mathbf{r_2}) + \frac{m}{d^3}\mathbf{r}$$

where $\Omega^2 = m/d^3$ and $m = m_1 + m_2$, the five Lagrange points were calculated for the two systems in Part 2. The secant method (defined in `doSecant(min, max)`) was implemented to search for the roots of g for $x \in (-2, 2)$. "Intelligent" guesses for min and max were chosen by plotting $g(x)$ in excel for varying values of x. This way, the approximate locations of the Lagrange points could be read from the plot. They are

(1) $m_1 = 3$, $m_2 = 1$, and $d = 1$
$L_1 = (0.1650555, 0), L_2 = (1.2382835, 0), L_3 = (-1.4754764, 0)$,
$L_4 = (0.250, 1.0)$, and $L_5 = (0.250, -1.0)$
(2) $m_1 = 100$, $m_2 = 1$, and $d = 1$.
$L_1 = (0.4463525, 0), L_2 = (1.0985547, 0), L_3 = (-1.7438812, 0)$,
$L_4 = (0.490, 1.0)$, and $L_5 = (0.490, -1.0)$.

When the main C# program is run, these results are printed to the console.

Part 4. A circular orbit is plotted to represent the orbit of the JWST. This is an approximation; the telescope's orbit will be perturbed by the gravitational fields of the sun, planets, and other masses. The ideal placement for the JWST is at L2, where the Sun and Earth aligned behind the telescope, which will point away from the Earth. Since the aperature of the telescope will face away from the nearest planet and star it will be protected from any light they emit or reflect. Note that the telescope will orbit Earth as Earth orbits sun.

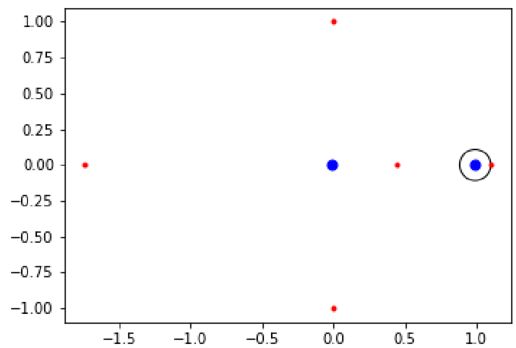

FIGURE 3. Proposed orbit of JWST, at $L2$. The Sun and Earth are shown as large blue points, and the Lagrange points are in red.