Homework 1: Graphing and Introduction

Caden Gobat

January 27, 2021

1 Introduction

In general, if the forces on a body are known, one can derive equations representing its position by solving differential equations using Newton's second law. In the specific case of springs and their oscillatory motion, the force itself depends on the position of the mass. This is apparent from Hooke's Law: $\mathbf{F} = k|r|(-\hat{\mathbf{r}})$. By treating $\mathbf{r}$ as a vector in (x, y) space, and solving some simple second-order linear differential equations, one can obtain the general solution for the x and y positions of the mass throughout time through the vector equation $\langle x(t),\ y(t)\rangle = \langle A_x \sin(\omega t + \phi_x),\ A_y \sin(\omega t + \phi_y)\rangle$. The two components can be treated essentially independently of one another, and as such we can get a 2D position of the mass for any point t in time, provided we know (or can define) the constants A_x, A_y, ω, ϕ_x, and ϕ_y.

Here I will do exactly this, and plot the results.

2 Results

Question 1

Submitted for the code assignment. ✓

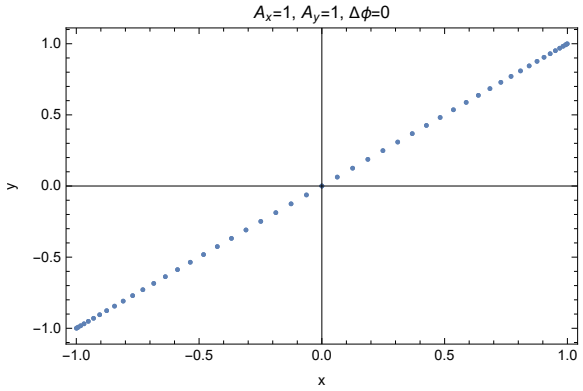
Question 2

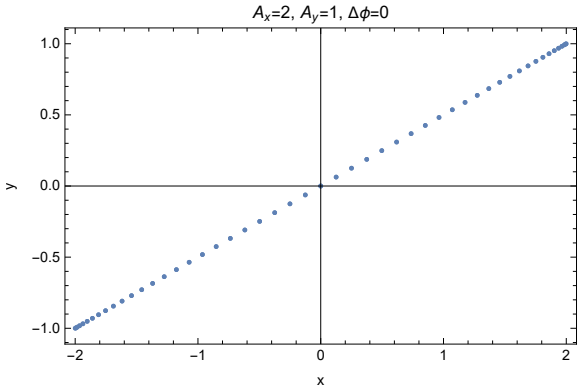
The plots on the following page were generated using the `C` and `Mathematica` code submitted for the code assignment. Each row represents a different phase difference ($\Delta\phi$) requested in the assignment, while the two columns represent $(A_x, A_y) = (1, 1)$ and $(A_x, A_y) = (2, 1)$, respectively. Note the scaling of the axes between plots, as the shapes themselves can appear deceptively similar.

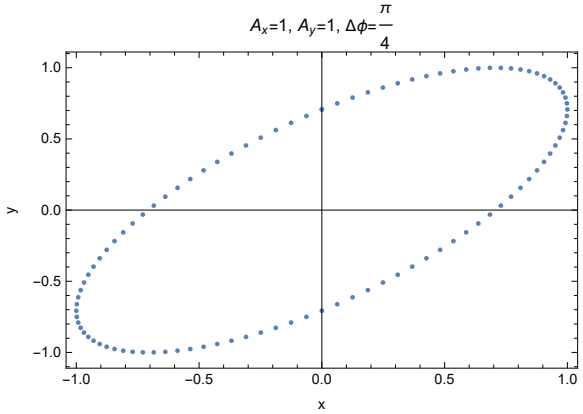
Question 3

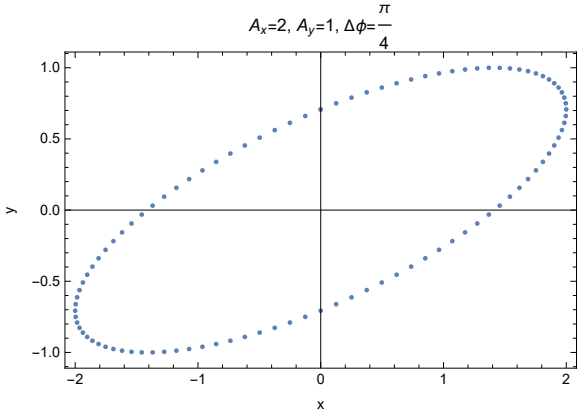
In general, it seems like these equations tend to produce ellipses of various kinds. A_x and A_y affect the scaling in each direction (indirectly, the semi-major and semi-minor axes), and the phase difference $\Delta\phi$ affects the eccentricity and angle of the ellipse. The frequency ω only affects the "speed" at which the mass travels, not the actual shape of its path. As such, it was held constant due to the fact that it has little discernible impact on the appearance of the plots (besides the spacing between each point in time, which is not conveyed in this "strobe light"-style plot). Perhaps an animation would help to illustrate this concept further.

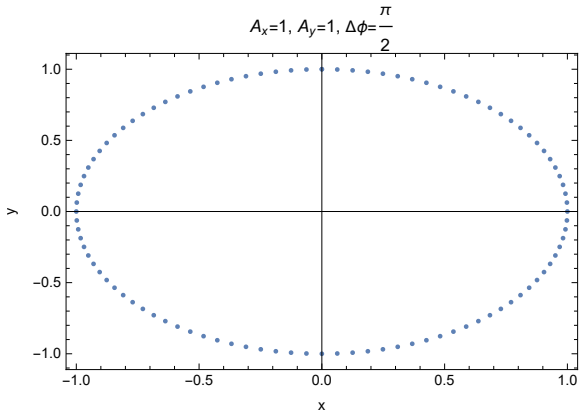
It seems that $\Delta\phi = n\pi$, where $n = 0, 1, 2, 3, ...$ produces a line (essentially an ellipse with eccentricity= 1), while $\Delta\phi = \frac{n\pi}{2}$, whereas $n = 1, 3, 5, ...$ creates circles (assuming $A_x = A_y$, otherwise just an ellipse with its axes aligned with the x- and y-axes). Other phase differences produce ellipses of various eccentricities (and, as long as $A_x = A_y$, the semi-major axis will be aligned with either $y = x$ or $y = -x$).

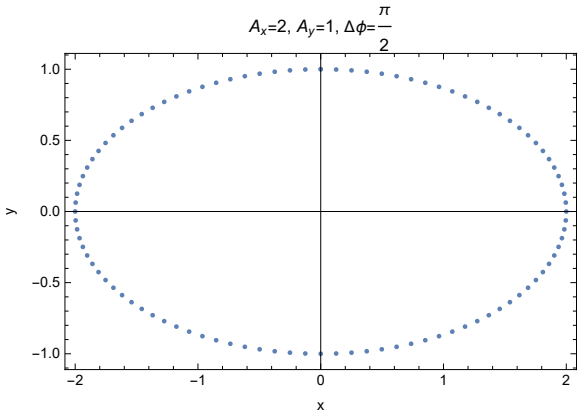

A_x=1, A_y=1, $\Delta\phi$=0
y
x

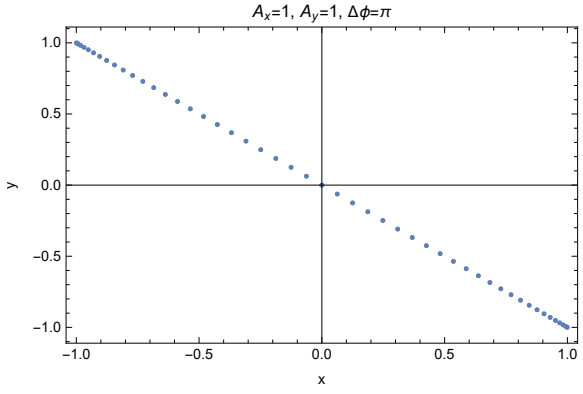

A_x=2, A_y=1, $\Delta\phi$=0
y
x

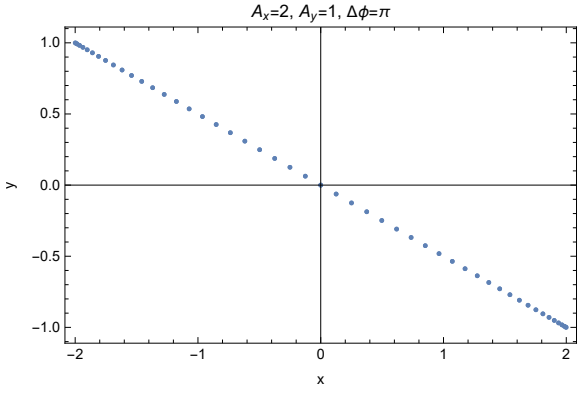

A_x=1, A_y=1, $\Delta\phi=\frac{\pi}{4}$
y
x


A_x=2, A_y=1, $\Delta\phi=\frac{\pi}{4}$
y
x


A_x=1, A_y=1, $\Delta\phi=\frac{\pi}{2}$
y
x

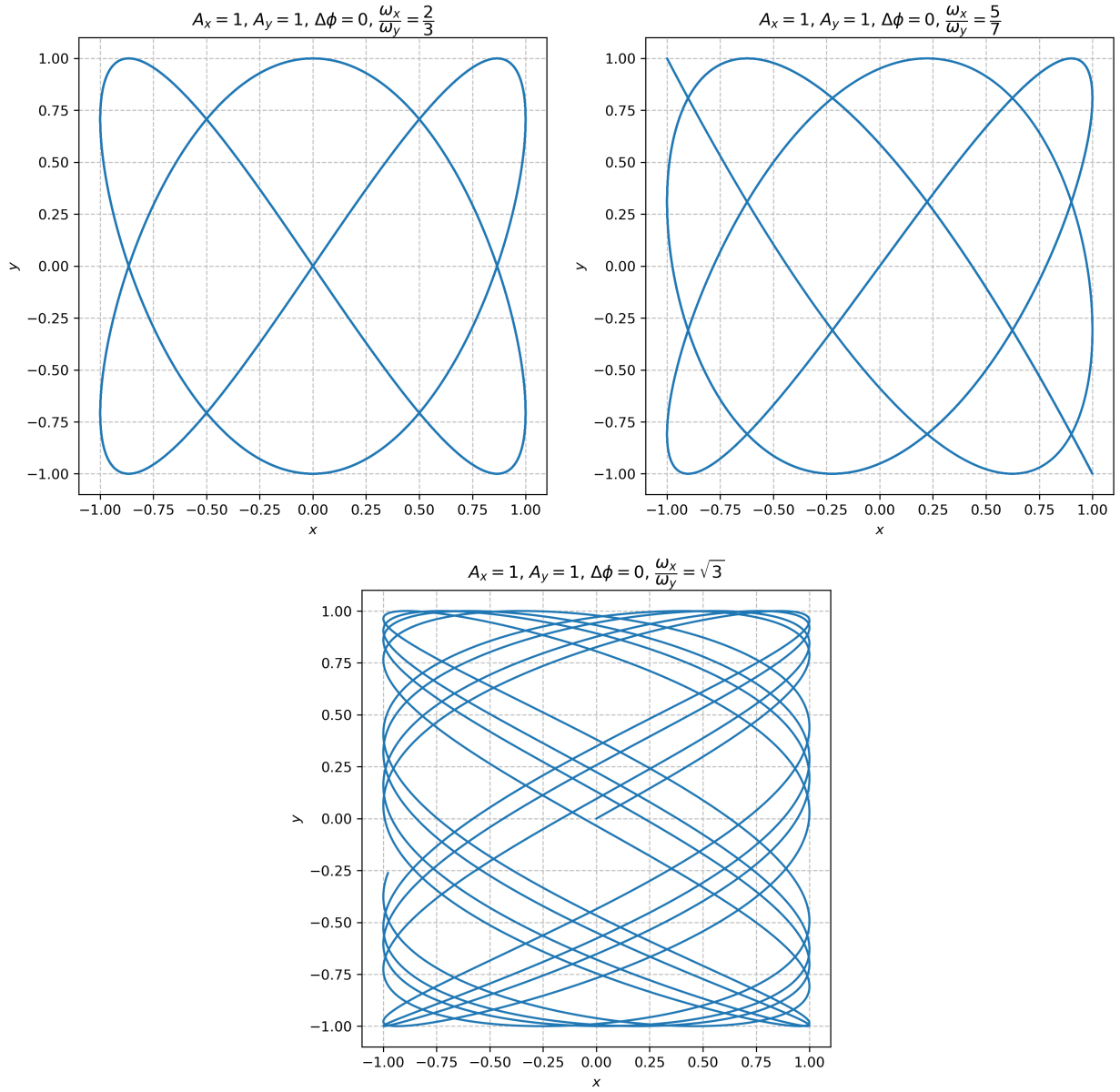

A_x=2, A_y=1, $\Delta\phi=\frac{\pi}{2}$
y
x


A_x=1, A_y=1, $\Delta\phi$=π
y
x


A_x=2, A_y=1, $\Delta\phi$=π
y
x

Bonus problem

The following plots of Lissajous figures were created using a similar setup, but with $\omega_x \neq \omega_y$. The parameters that varied in the previous problems were held constant here, while the ratio $\frac{\omega_x}{\omega_y}$ was manipulated to create the shapes seen here. The time domain was set to run from $0 \leq t \leq \frac{50}{\omega_{\min}}$ in each case.



Rational values of $\frac{\omega_x}{\omega_y}$ (such as $\frac{2}{3}$ and $\frac{5}{7}$) produce figures where the same shape is traced out, period after period. It seems that if the numerator is even, it creates a "closed loop" where there are no apparent endpoints (seen in the first figure above), while odd numerators create more of an open-ended line, as in the second figure.

Irrational values of the frequency ratio, such as $\frac{\omega_x}{\omega_y} = \sqrt{3}$, mean that the traced path is not the same each period. This results in a sort of drift over time, as the last figure above shows.

3 Conclusions

In all, this assignment constituted a good introduction to the principles of computation in `C` and subsequent visualization in `Mathematica`. I ran into some issues initially because it seems that math constants like `M_PI` are not available in the `c99` implementation, so one must either define it manually (something along the lines of `#define PI 4.*atan(1)`) or use a different standard when compiling. Depending on the operating system I was working on, I found that I sometimes had to pass the `-lm` flag to get the math library to link properly as well.

Importing the generated data into `Mathematica` also presented some trouble at first because I was formatting it as comma-separated, which apparently the `ReadList[]` function is unable to parse. Switching to tab-separated values in the `C` code output resolved the issue.

Besides these minor technical issues, the project went well and I was able to complete it successfully and with an improved understanding of the softwares involved.