Coding Physics

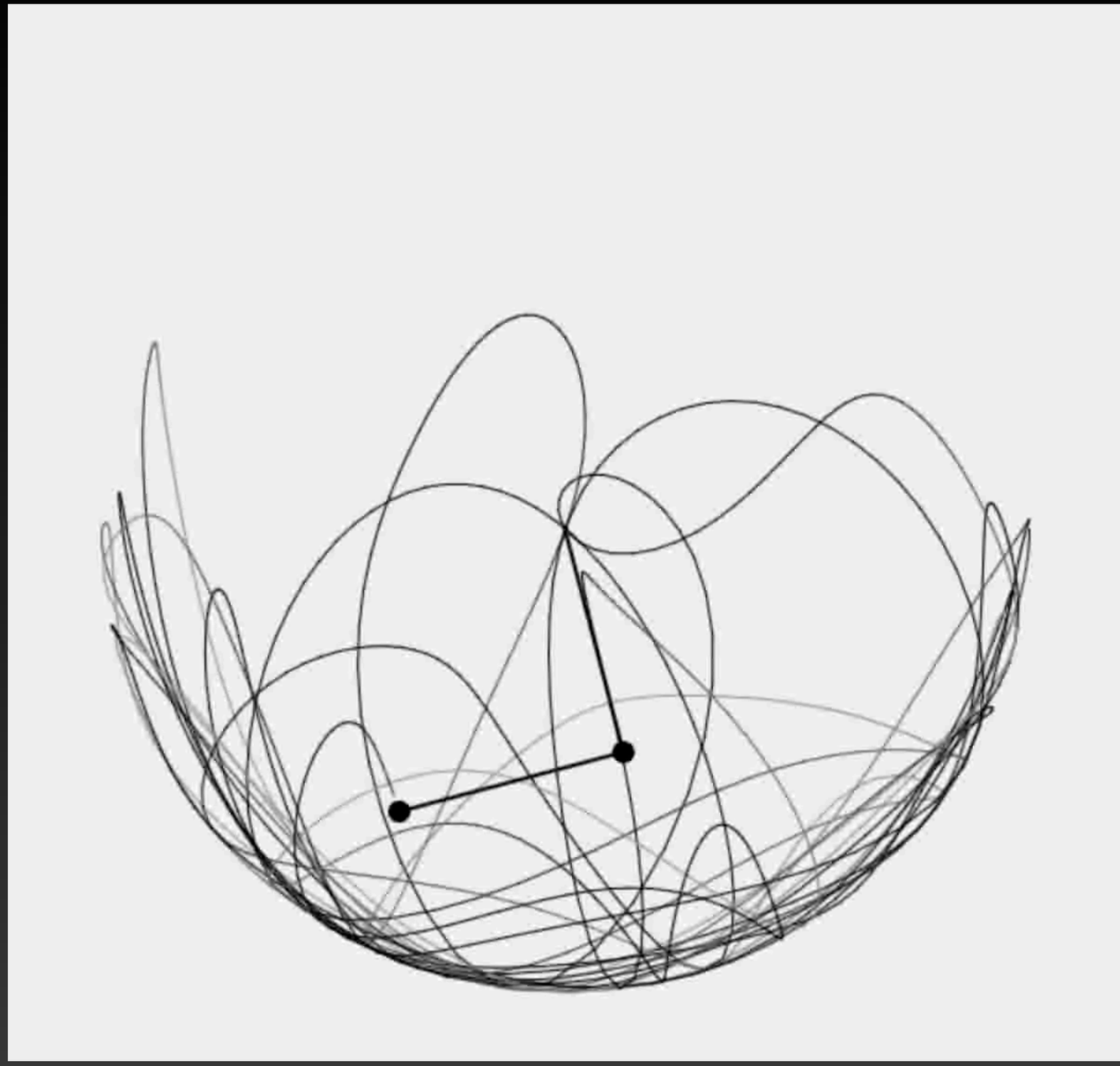
Solving physics problems with computers

Garett Brown 2020

UNIVERSITY OF
TORONTO

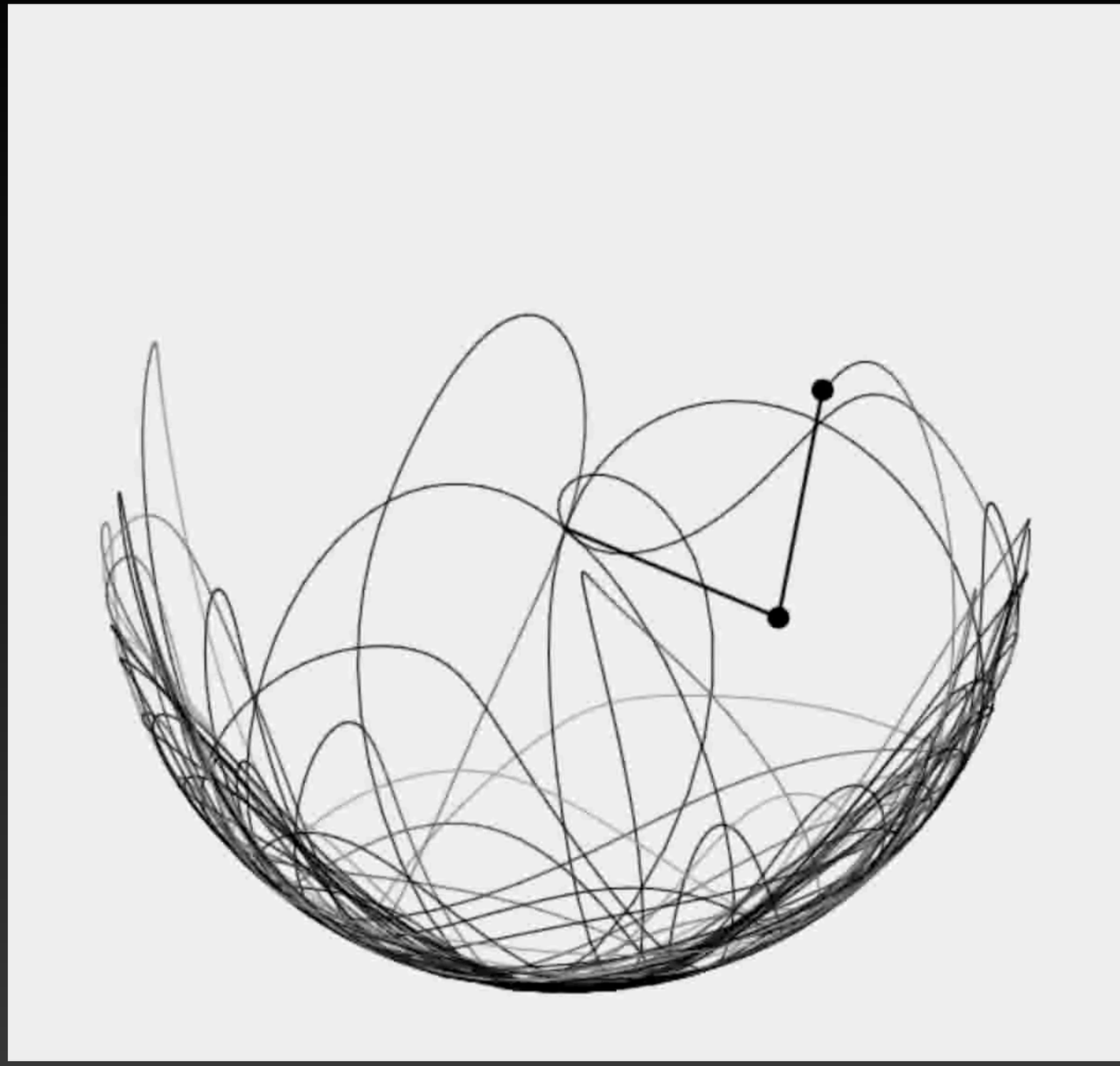
Why should we solve Physics Problems with Computer Code?

- Some experiments cannot be done in a physical lab.
  - Simulating the universe or the planets.
- Some experiments are too costly to practically run thousands of times.
- Machine Learning methods and AI can help us see new patterns



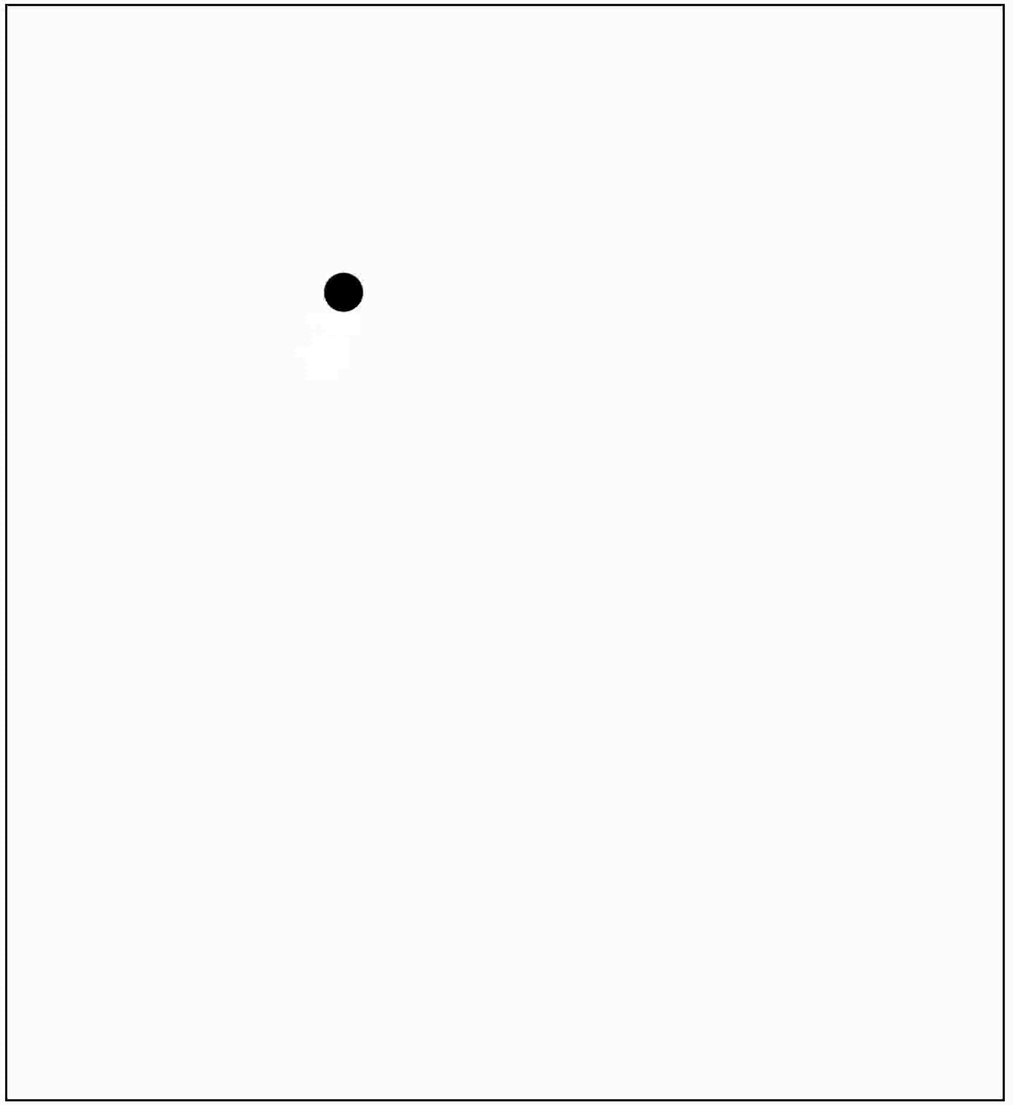
How do we solve Physics Problems with Computer Code?

- First, we need to be able to describe the motion.
- For many problems this can be done with Newton's 2nd law.
  - $F = ma$ and $\tau = I\alpha$
- We will do this for 2 examples:
  - The simple harmonic oscillator
  - The simple pendulum



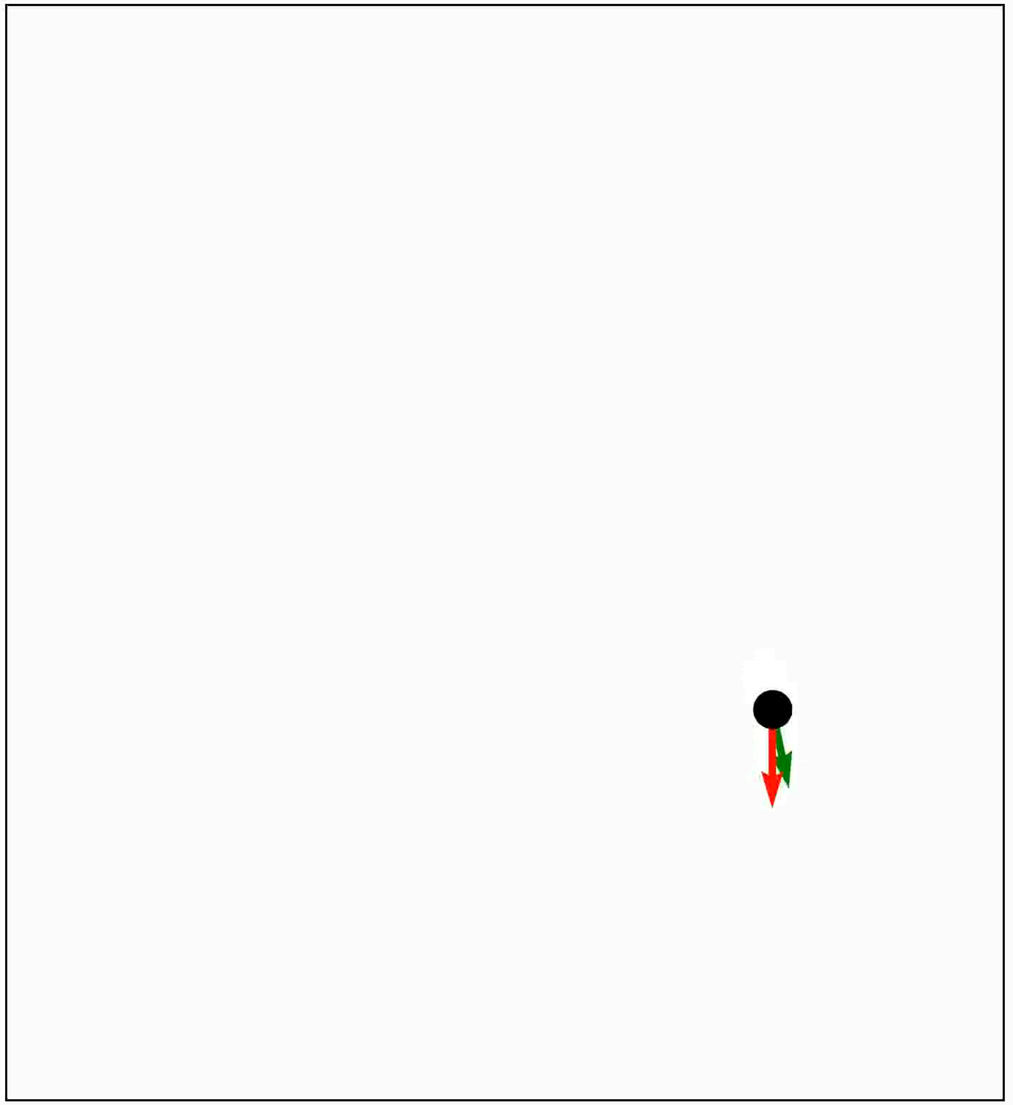
How do we solve Physics Problems with Computer Code?

- If we know the location (position) of an object, how can we know where it will go next?
- We can determine where it will be next if we know the velocity of the object.
- But what if the velocity is changing?



How do we solve Physics Problems with Computer Code?

- If the velocity is also changing, then we can determine how the velocity changes based on the acceleration.
- Velocity is the change in position over the change in time.
- $$v = \frac{x_f - x_i}{t_f - t_i} = \frac{\Delta x}{\Delta t}$$
- $$a = \frac{v_f - v_i}{t_f - t_i} = \frac{\Delta v}{\Delta t}$$

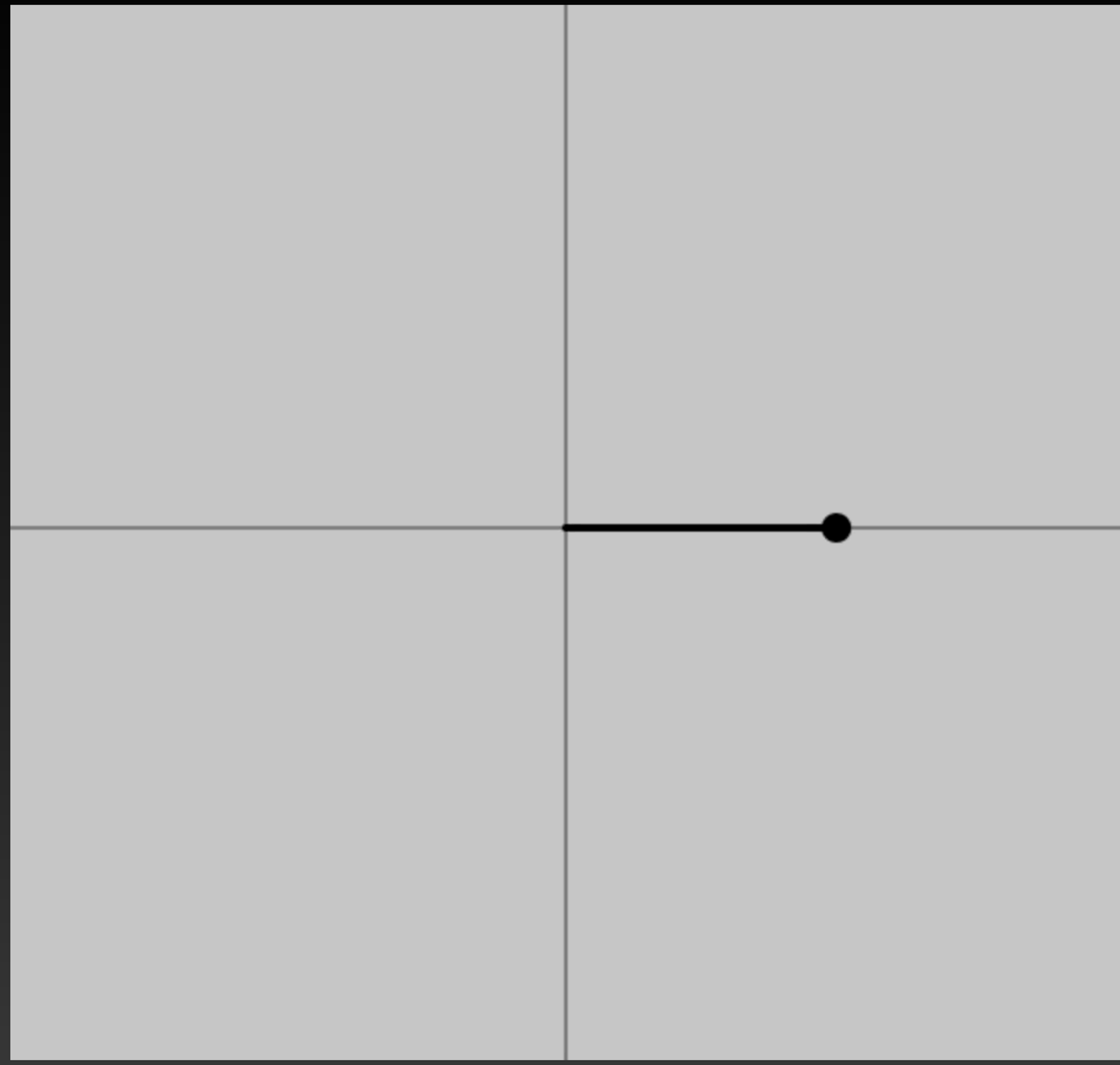


Simple Harmonic Oscillator

- Many physics models are based on the motion of the simple harmonic oscillator (SHO)
- We consider the motion of the mass as described by Hooke's law.
  - $F = -kx$

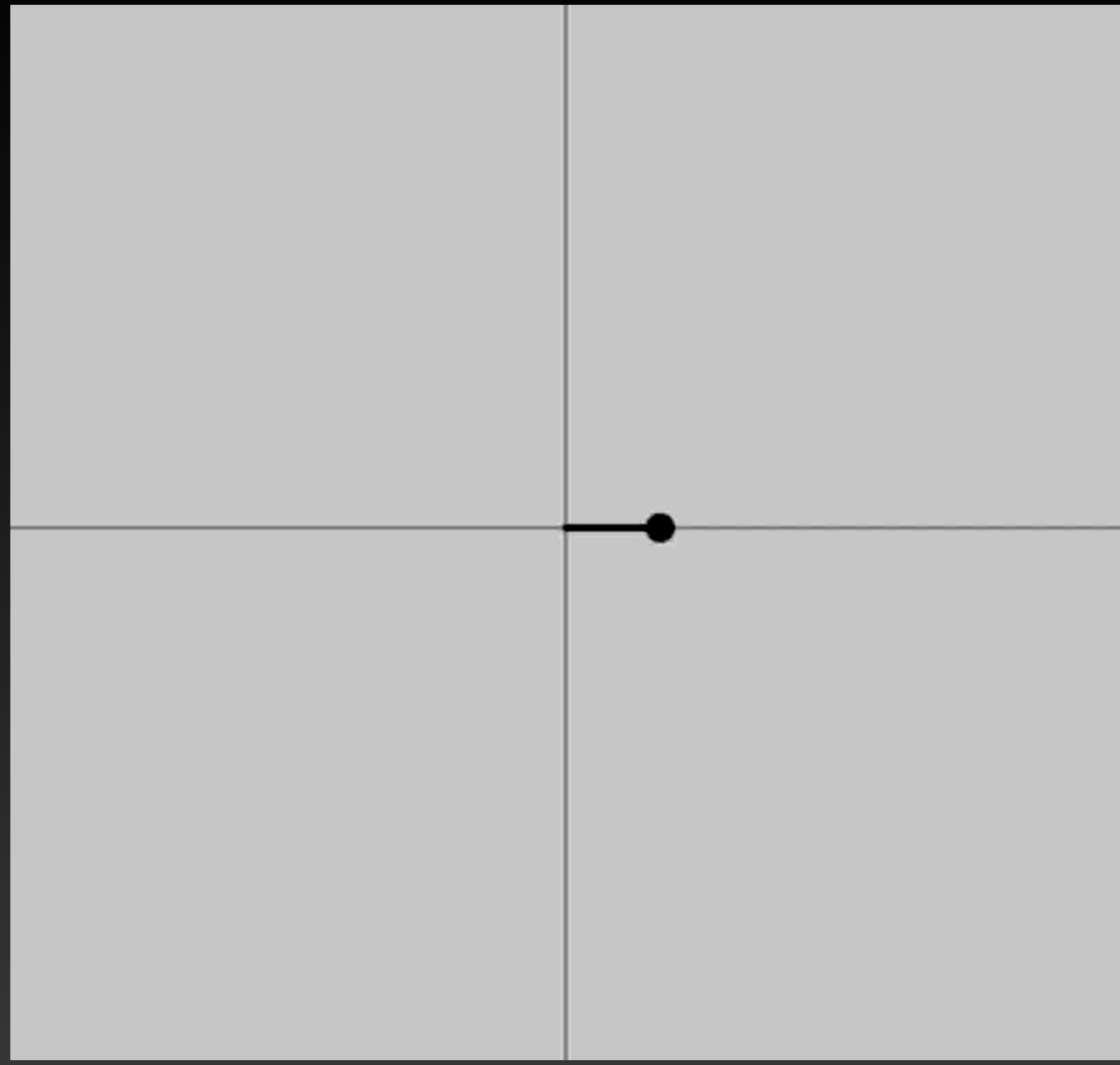
Follow Along:

- JavaScript: https://editor.p5js.org/zyrxvo/sketches/iYYZnF6Q6
- Python: https://trinket.io/library/trinkets/a030bc7979



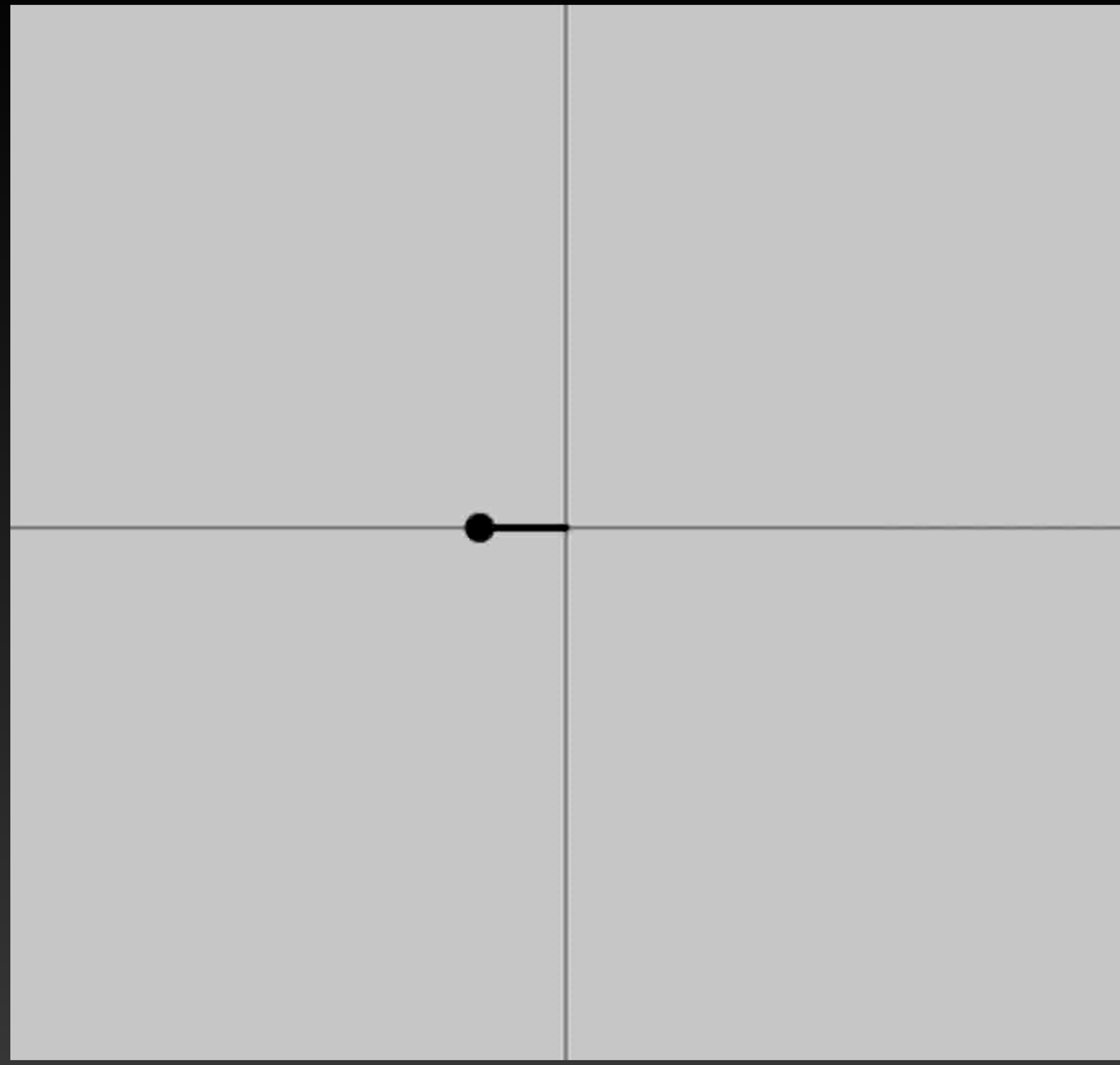
Simple Harmonic Oscillator

- Thus, combining Newton's 2nd law with Hooke's law we have
  - $F = ma$, $F = -kx$
  - $ma = -kx$
  - $a = -\frac{k}{m}x$
- This gives us an equation for how the acceleration is related to the position of the mass.



Simple Harmonic Oscillator

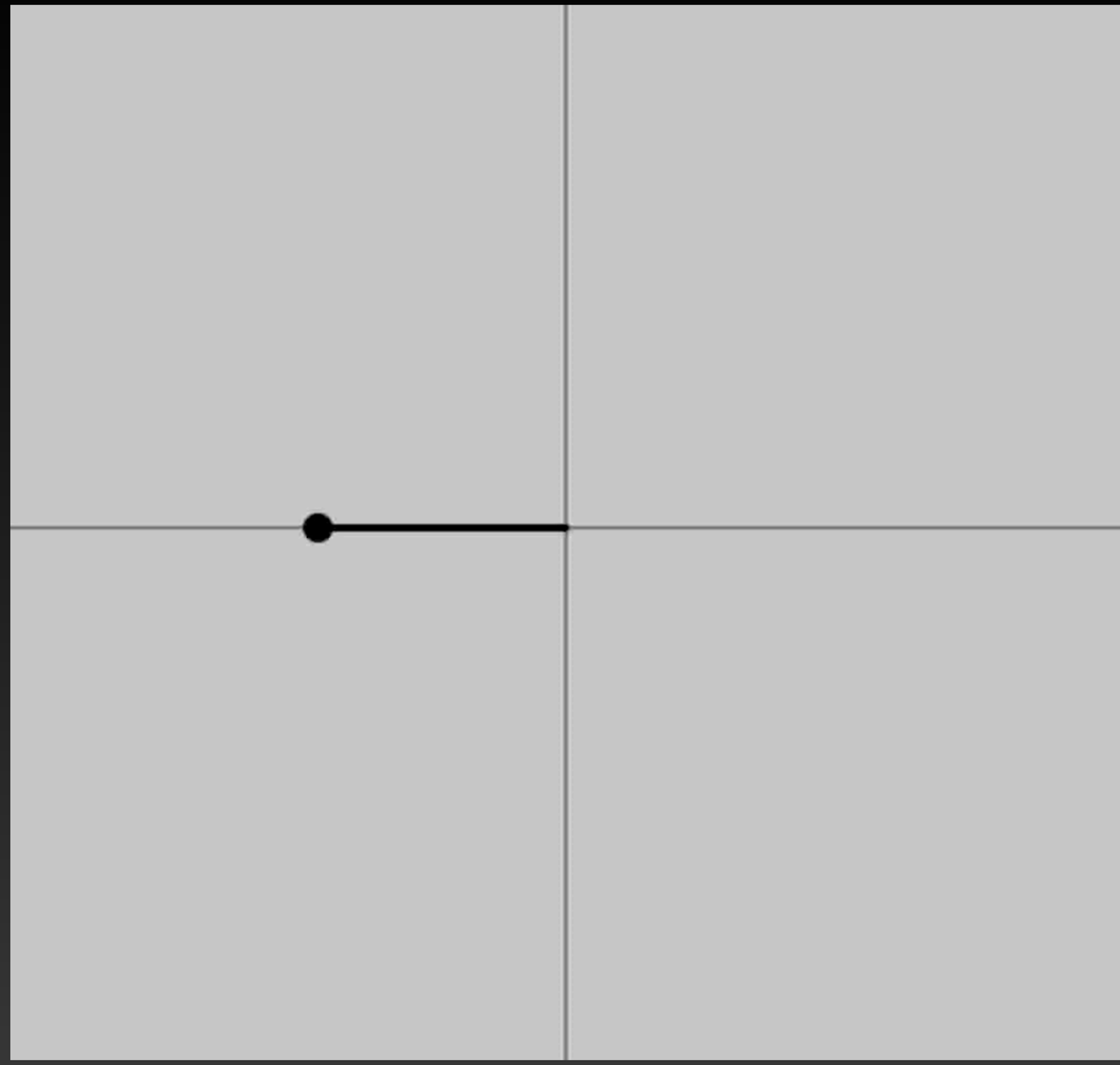
- Recall how the position of a mass changes due to the acceleration.
- We know that the position and velocity are related by $v = \Delta x / \Delta t$, so that the velocity is the change in position divided by the change in time.
- And similarly for the acceleration, $a = \Delta v / \Delta t$.



Simple Harmonic Oscillator

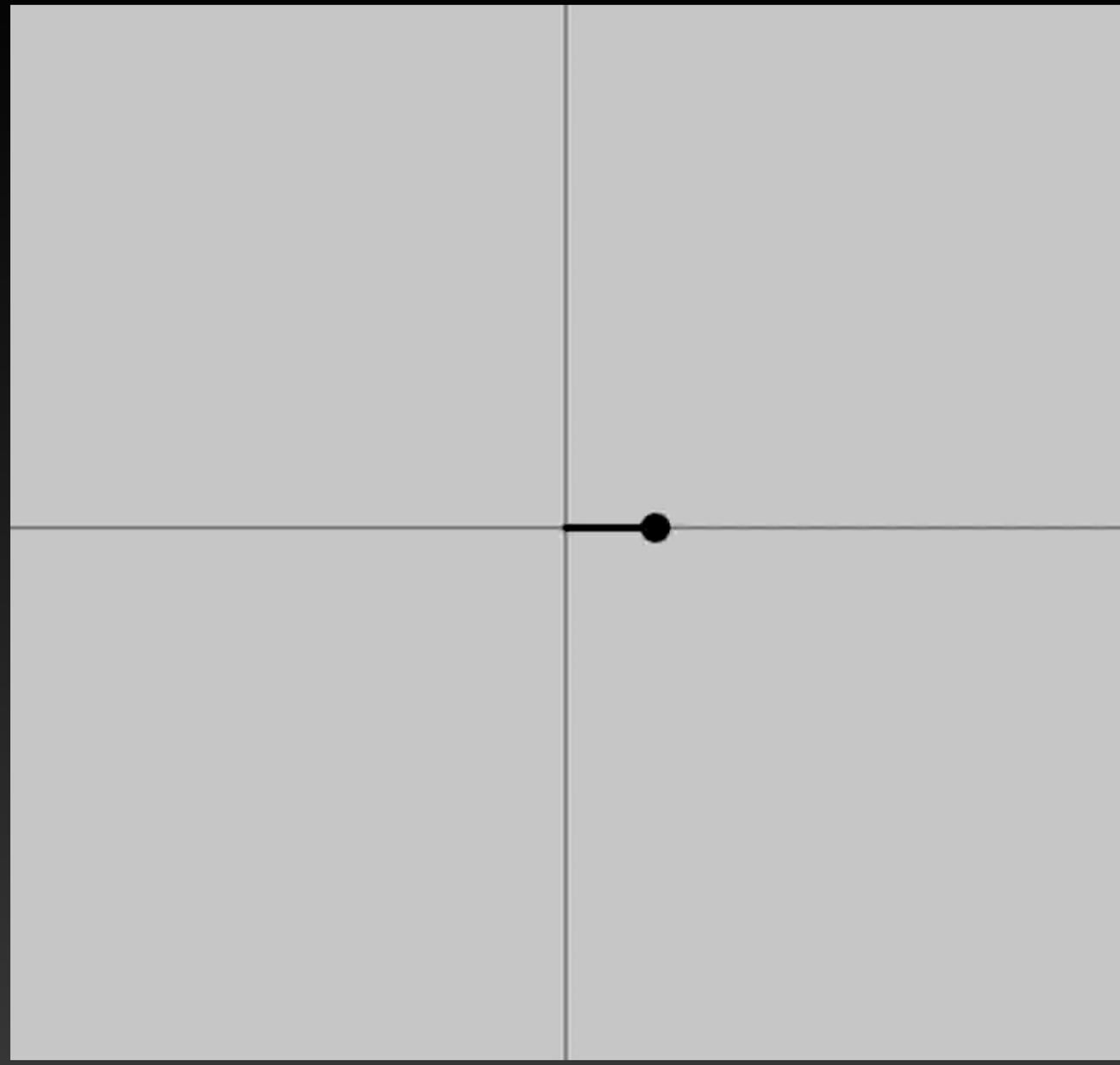
- These 3 equations enable us to describe the position of the mass and how it changes with time.
- First we can calculate what the acceleration of the mass will be based on its position (the amount the spring is stretched).

$$a = -\frac{k}{m}x$$



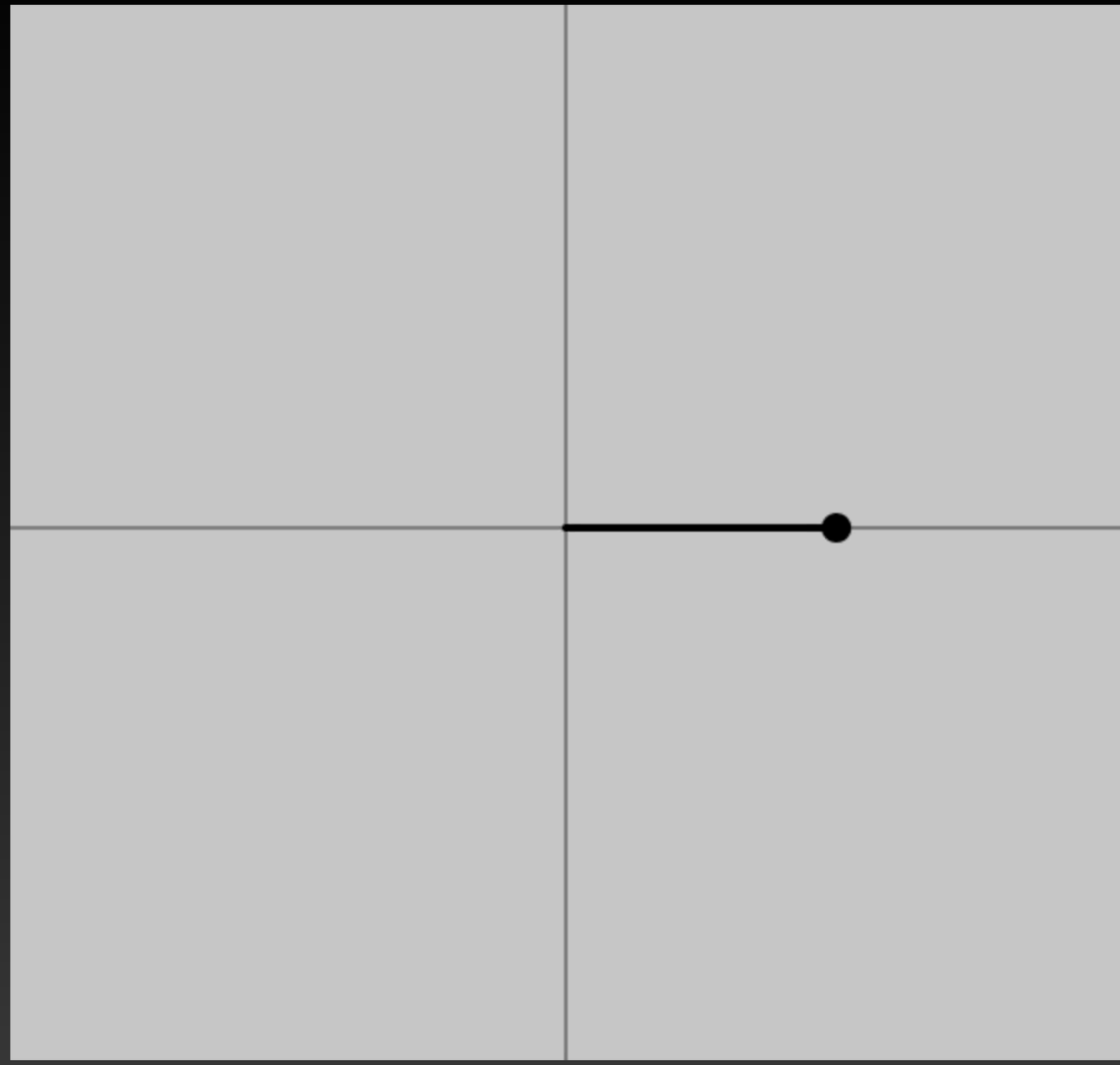
Simple Harmonic Oscillator

- Then we can update the velocity based on the calculated acceleration.
  - Using $a = \Delta v / \Delta t$, we can rearrange the equation for v_f
    - $\Delta v = a \Delta t$
    - $v_f - v_i = a \Delta t$
    - $v_f = v_i + a \Delta t$



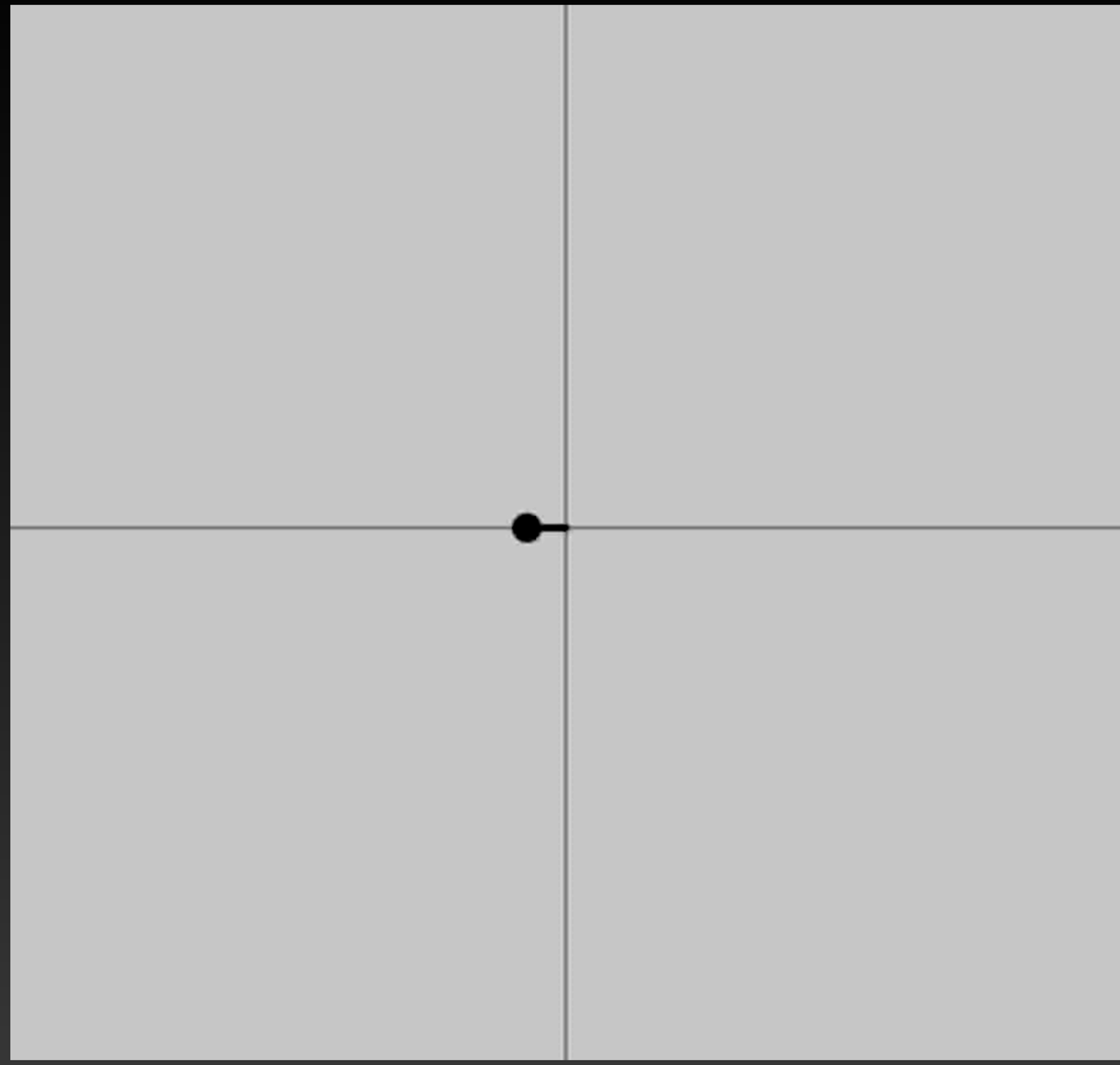
Simple Harmonic Oscillator

- Now we can update the position based on the updated velocity.
  - Using $v = \Delta x / \Delta t$ we can rearrange the equation for x_f
    - $\Delta x = v\Delta t$
    - $x_f - x_i = v\Delta t$
    - $x_f = x_i + v\Delta t$



Simple Harmonic Oscillator

- The final 3 equations are:
  - $a = -\frac{k}{m}x$
  - $v_f = v_i + a\Delta t$
  - $x_f = x_i + v\Delta t$
- If we calculate these 3 equations over and over again we can animate the position of the mass over time.



Simple Harmonic Oscillator

Coding Time!

- For JavaScript go to https://editor.p5js.org/zyrxvo/sketches/iYYZnF6Q6
- For Python go to https://trinket.io/library/trinkets/a030bc7979
- Don’t worry about understanding how the animations are done right now, all of the code used to animate the simple harmonic oscillator is already there. Just find the part that says,

```
**********************************
      Code Solution Here:
**********************************



**********************************
```

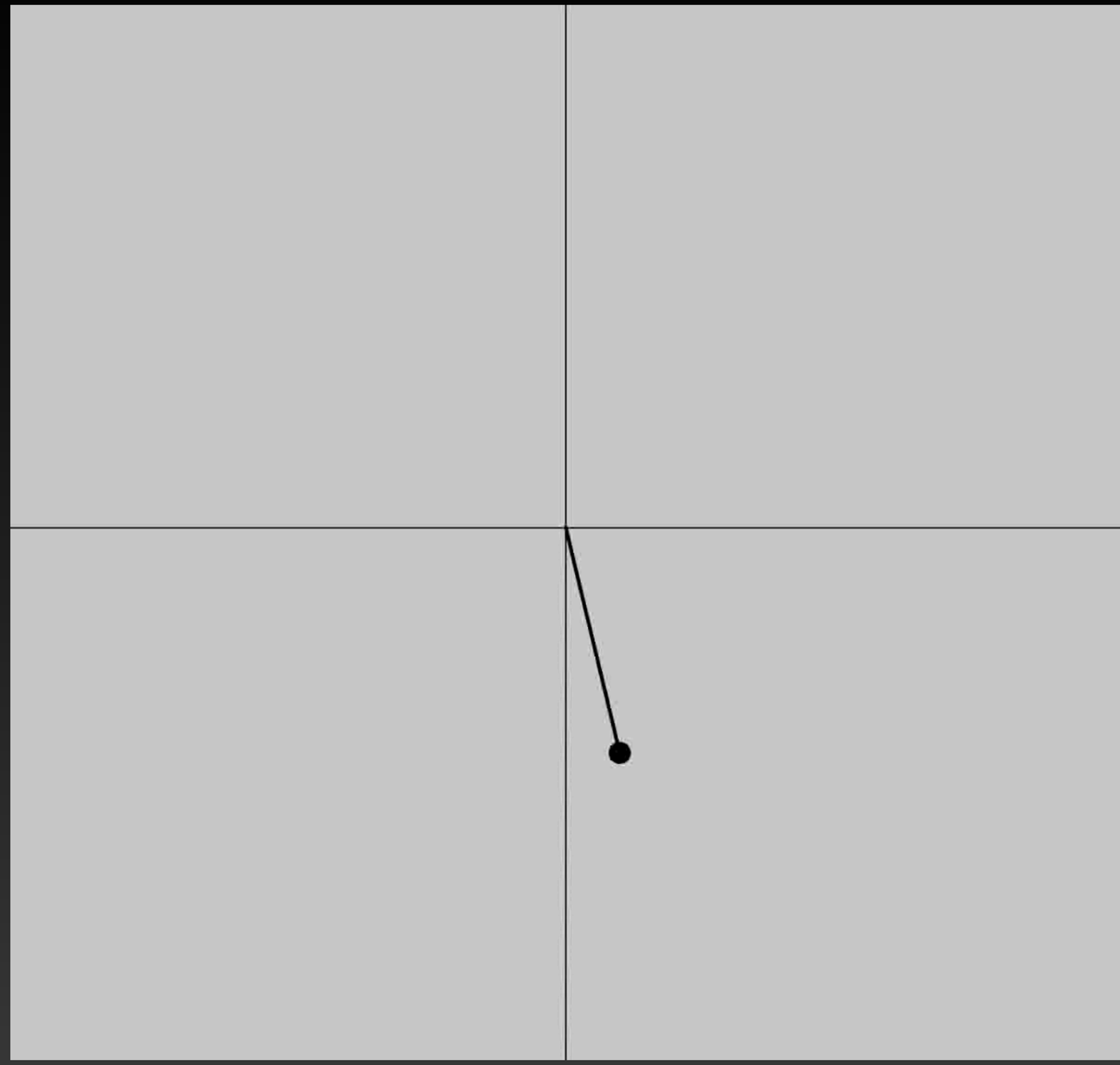
- and code up your solution between the lines. When you think you’ve got it, hit the play button at the top of the page and see if it worked! You can also play around with the initial conditions and physical constants.
- If you get stuck, the solutions can be found at:
  - JavaScript: https://editor.p5js.org/zyrxvo/sketches/ruSYnSzXS
  - Python: https://trinket.io/library/trinkets/04931fe5c8

The Pendulum

- The simple pendulum is often considered the next step in difficulty beyond the simple harmonic oscillator.
- As such, it is also often used to model many other physical phenomena, and is used by analogy to describe many more.

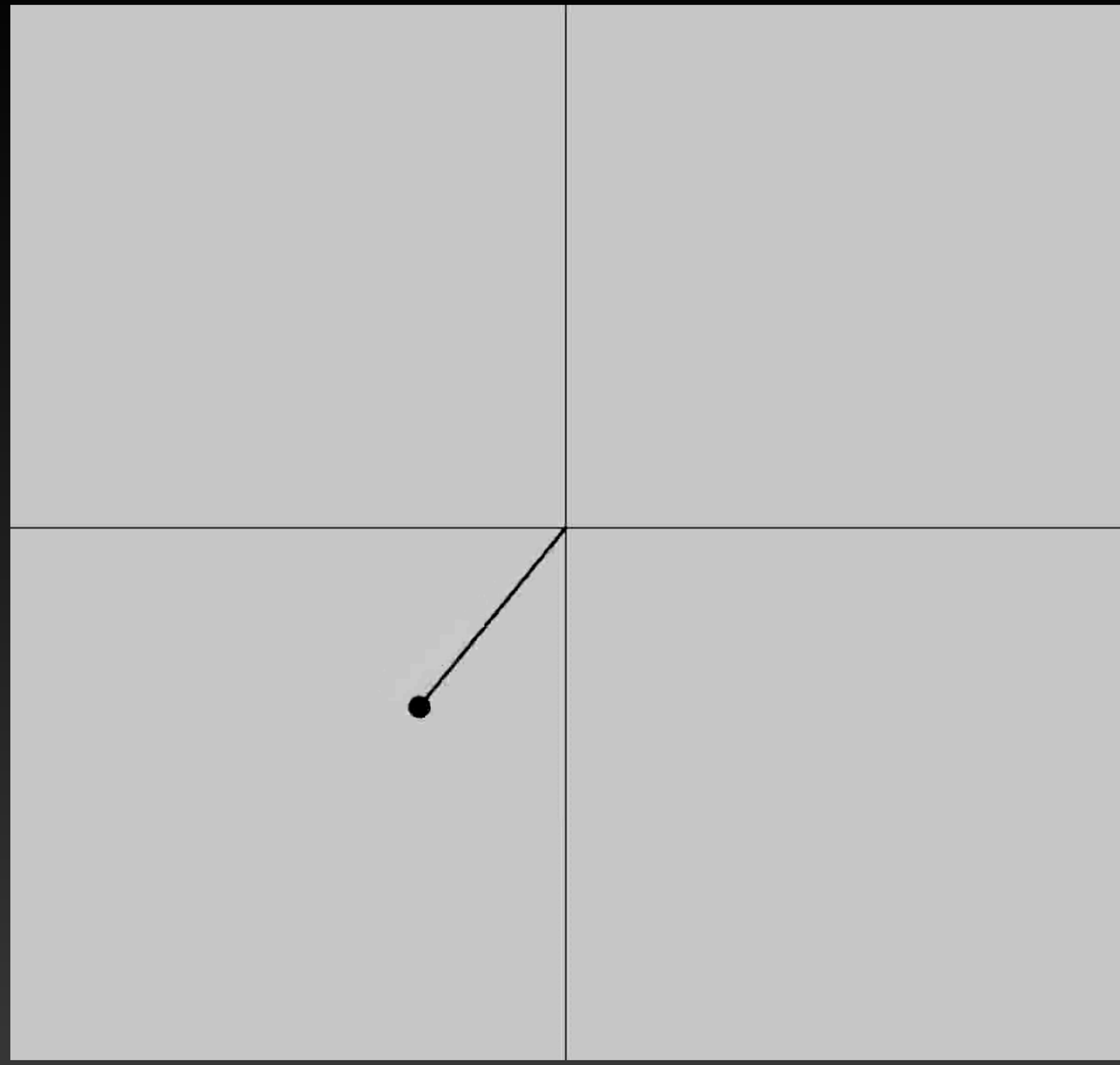
Follow Along:

- JavaScript: https://editor.p5js.org/zyrxvo/sketches/VNttrXjWP
- Python: https://trinket.io/library/trinkets/e455871d49



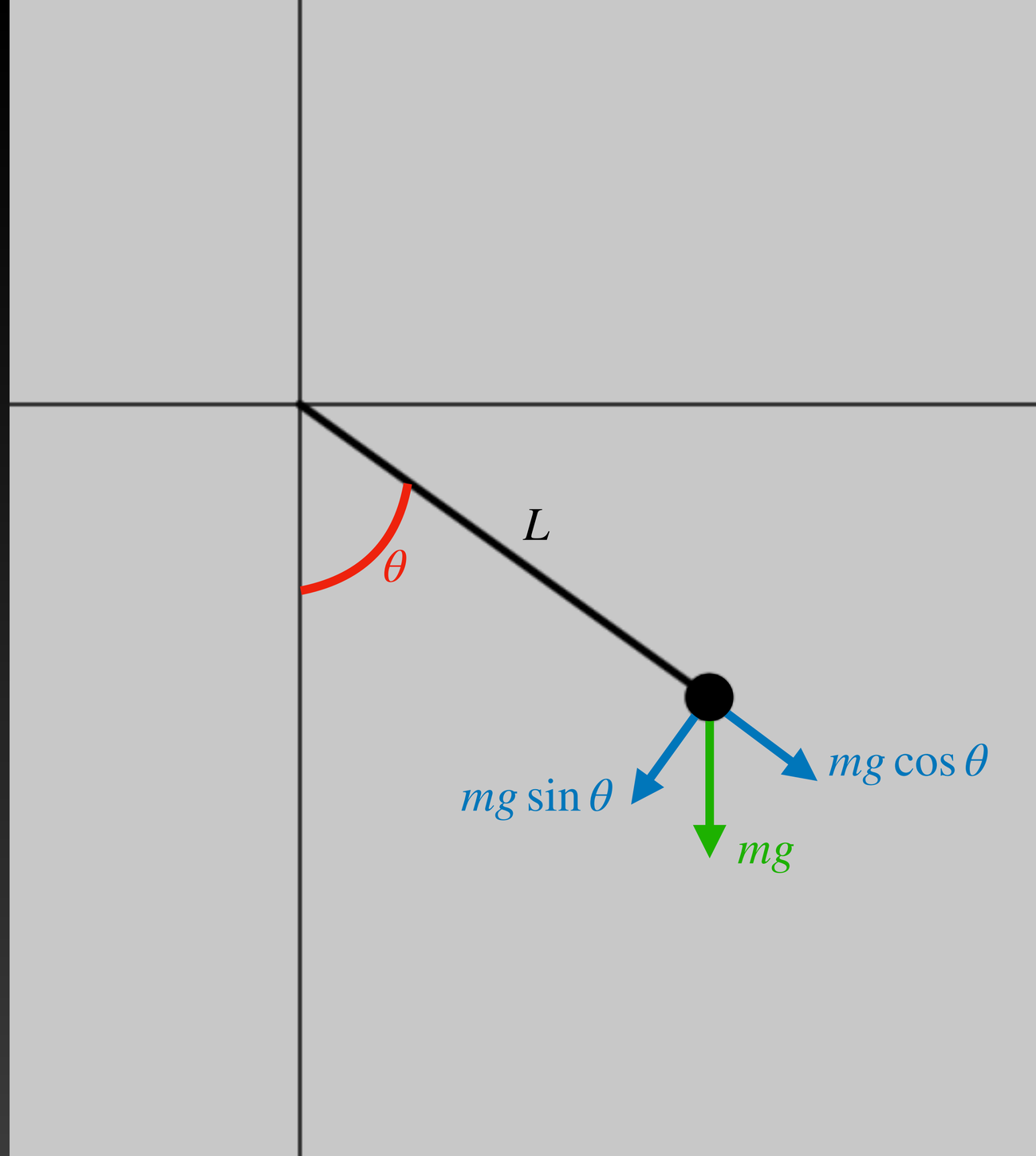
The Pendulum

- Once again, we begin with Newton's 2nd law, this time for rotational systems.
- Often this is written as:
  - $\tau = I\alpha$
- where τ is the torque, I is the moment of inertia, and α is the angular acceleration.



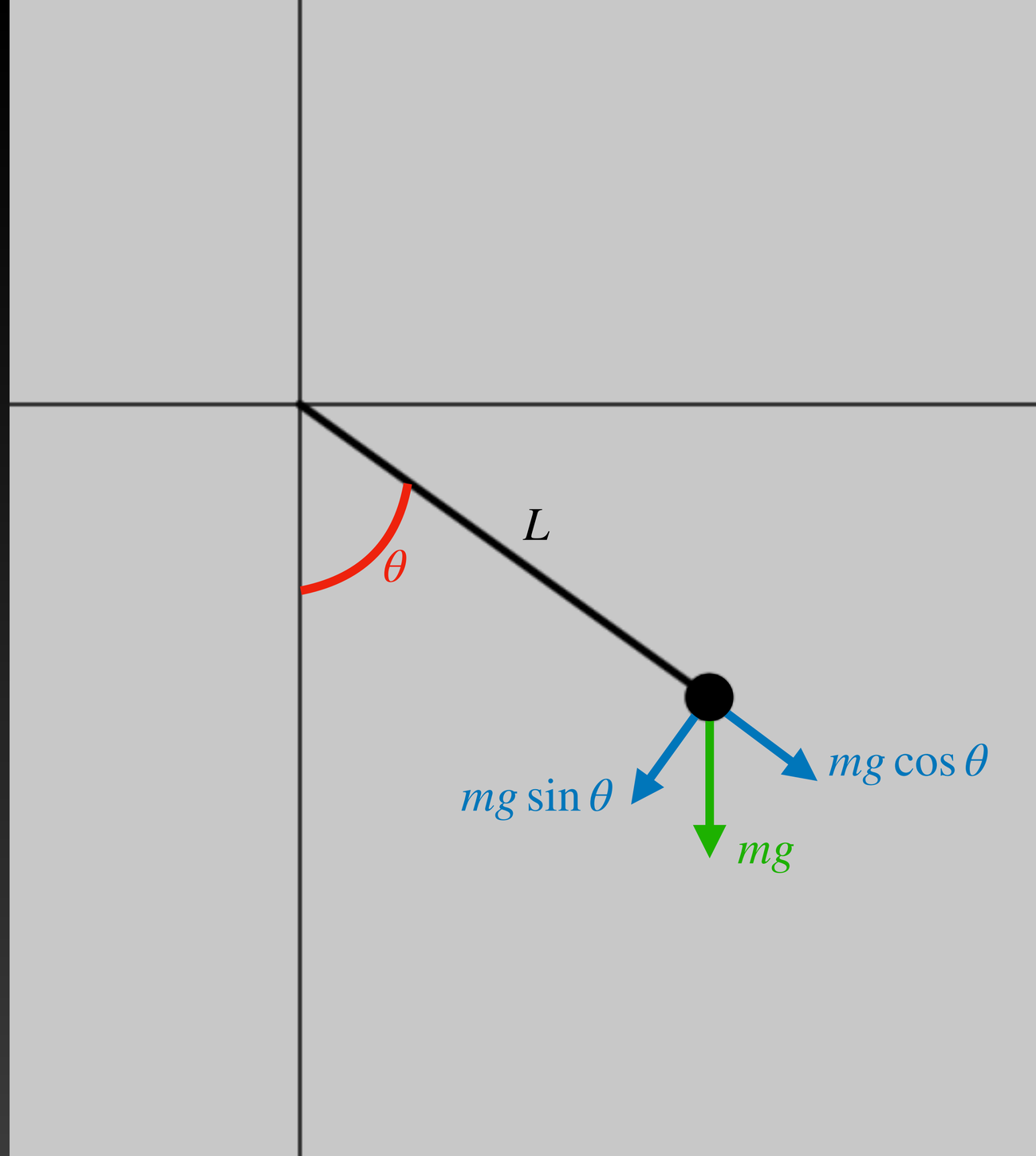
The Pendulum

- The position of the mass is determined only by θ, and gravity is the only force causing a torque on the mass, which is given by $\tau = -mgL\sin\theta$.
- The moment of inertia for the mass is $I = mL^2$
- The angular acceleration of the mass is simply α.



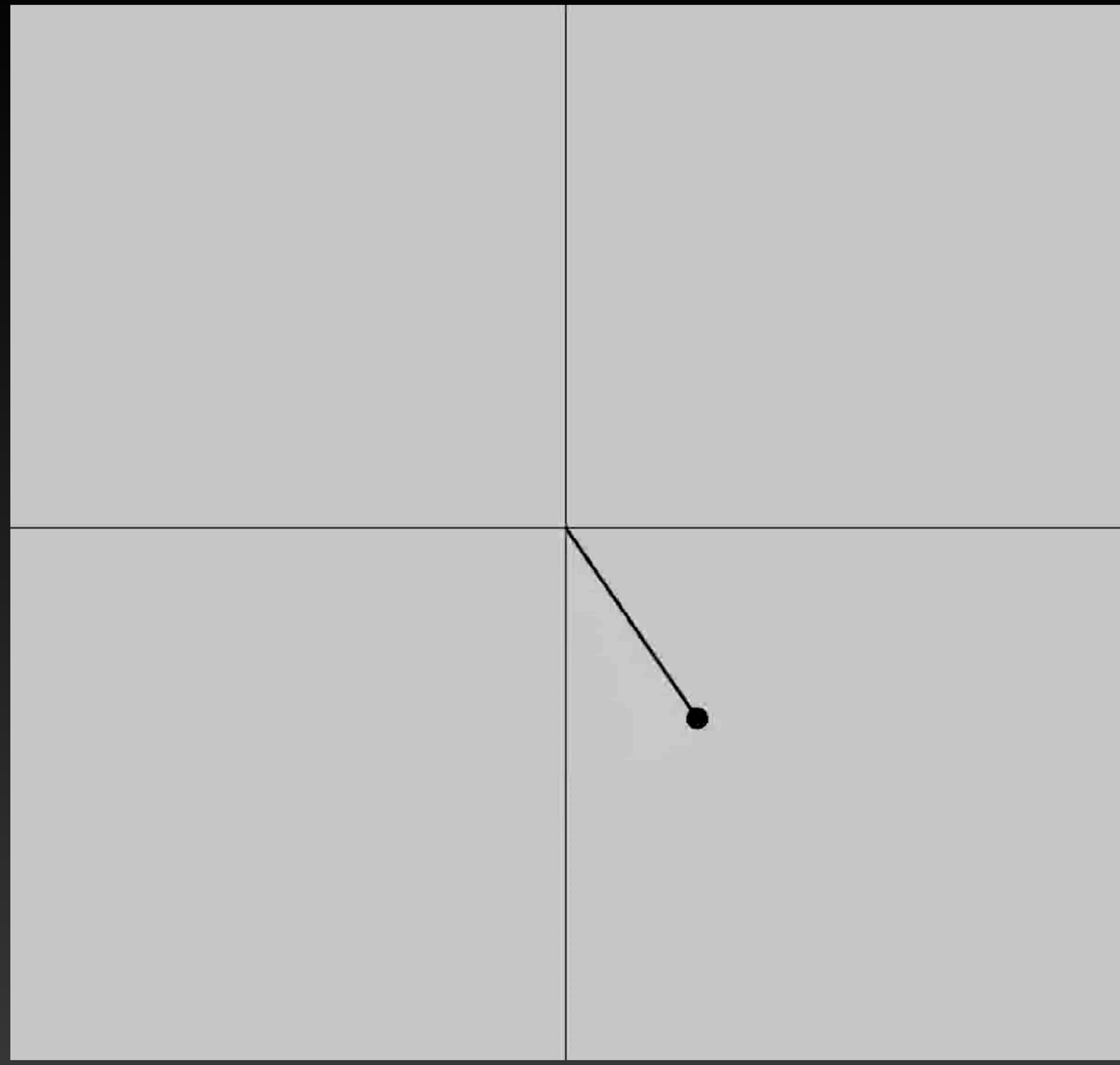
The Pendulum

- Bringing this all together gives,
  - $\tau = I\alpha$
  - $-mgL\sin\theta = mL^2\alpha$
- Now rearrange the equation to better see how the angular acceleration (α) is related to the angular position (θ).
  - $\alpha = -\frac{g}{L}\sin\theta$



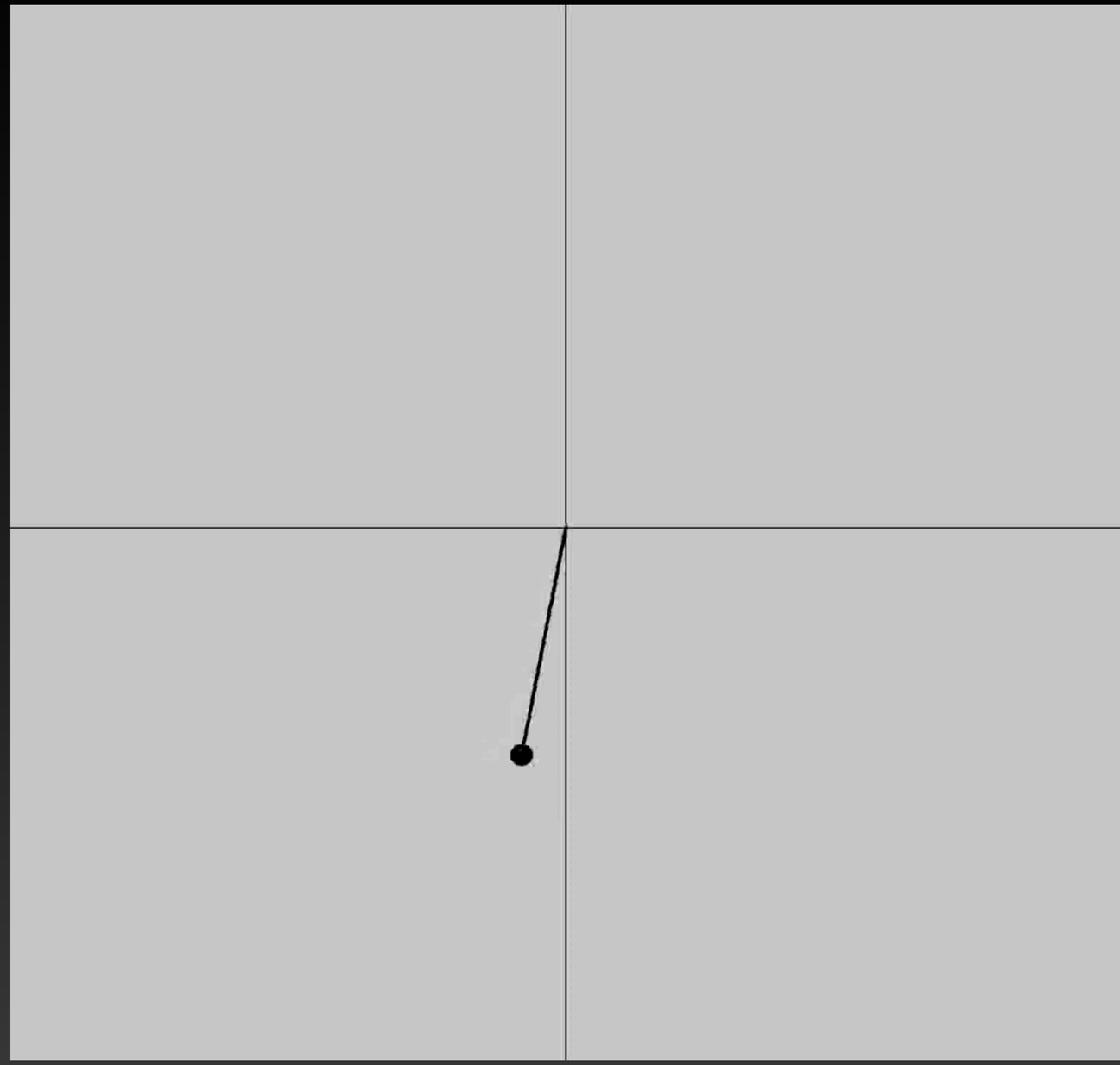
The Pendulum

- Now that we know how the angular acceleration is based on the angular position we can follow the same method we used for the simple harmonic oscillator (this time for angular variables).
- Angular velocity is $\omega = \Delta\theta/\Delta t$ and angular acceleration is $\alpha = \Delta\omega/\Delta t.$



The Pendulum

- Putting this all together we have,
  - $\alpha = -\frac{g}{L}\sin\theta$
  - $\omega_f = \omega_i + \alpha\Delta t$
  - $\theta_f = \theta_i + \omega\Delta t$
- Again, if we calculate these 3 equations over and over again we can animate the position of the mass over time.



The Pendulum
Coding Time!

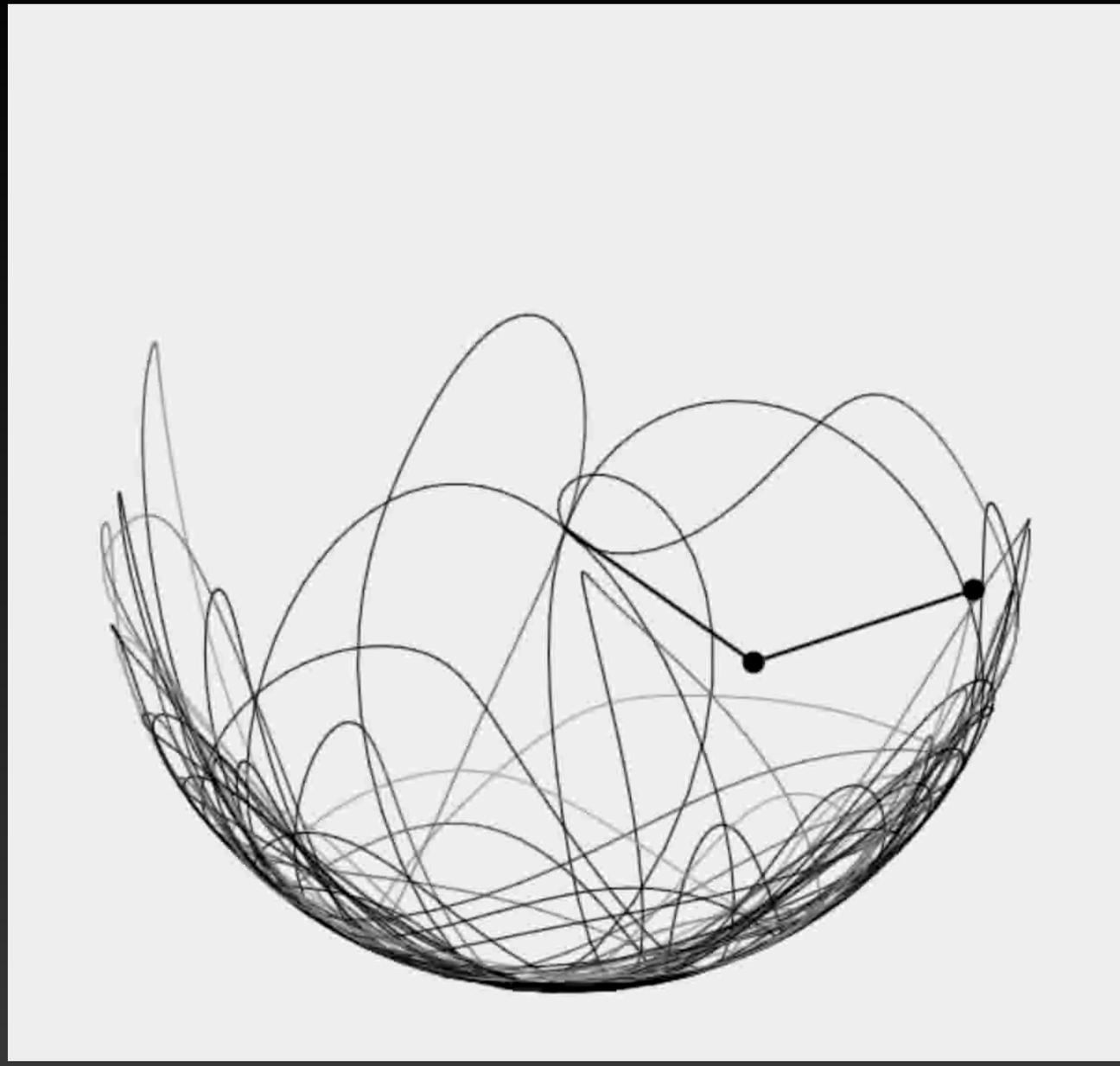
- For JavaScript go to https://editor.p5js.org/zyrxvo/sketches/VNttrXjWP
- For Python go to https://trinket.io/library/trinkets/e455871d49
- Again, all of the code used to animate the pendulum is already there. Just find the part that says,

```
**********************************
      Code Solution Here:
**********************************



**********************************
```

- and code up your solution between the lines. Again, when you think you've got it, hit the play button at the top of the page and see if it worked! And, like before, you can also play around with the initial conditions and physical constants.
- If you get stuck, the solutions can be found at
  - JavaScript: https://editor.p5js.org/zyrxvo/sketches/mvERTFb5R
  - Python: https://trinket.io/library/trinkets/77ae8a6382
- As a bonus, see if you can add damping!

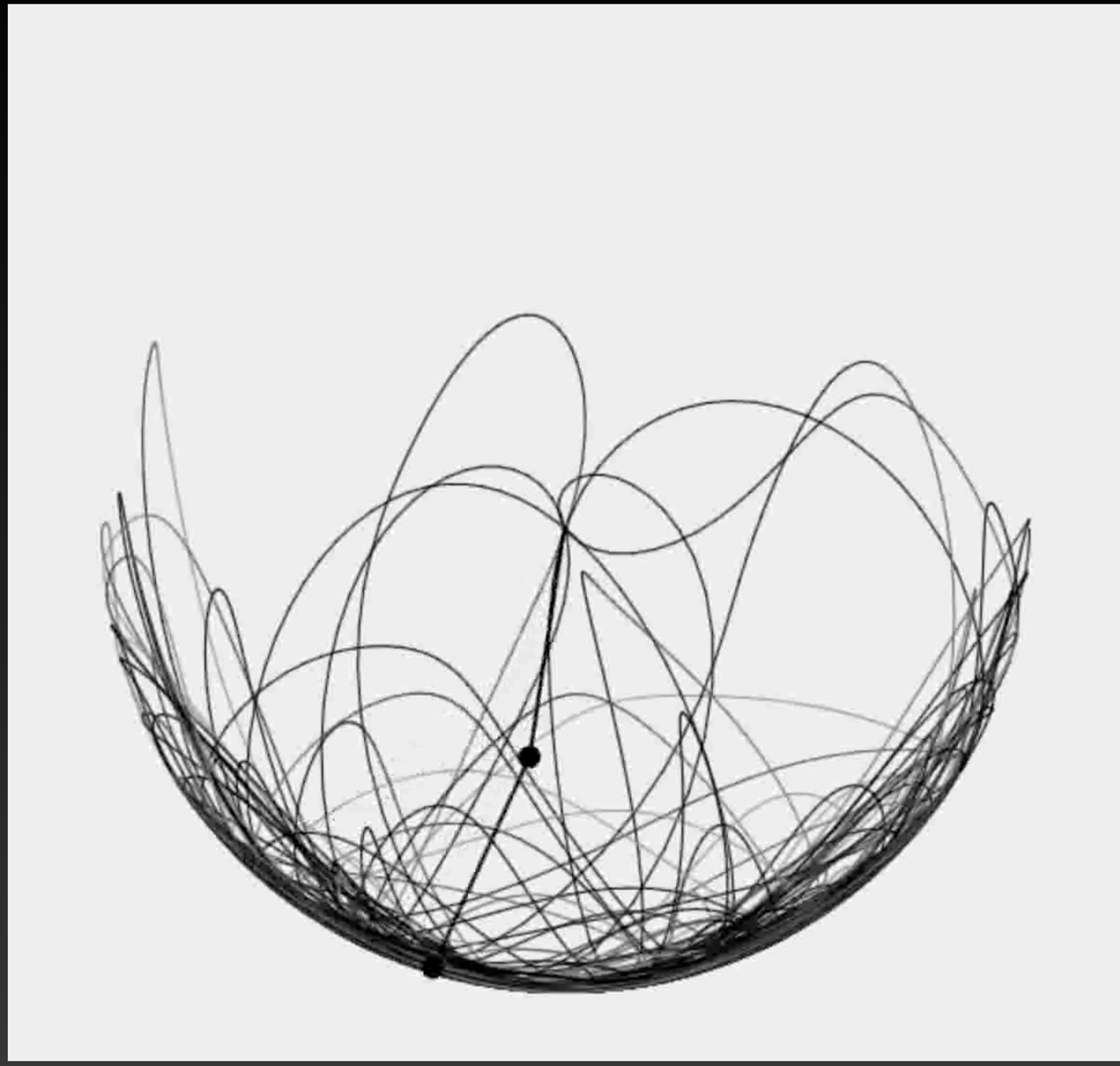
The Double Pendulum

- We are not going to derive the equations of motion for this one, instead we're going to spend some time exploring and tinkering.
- JavaScript: https://editor.p5js.org/zyrxvo/sketches/VpCiCDQ5s
- Python: https://trinket.io/library/trinkets/f0de22daa9



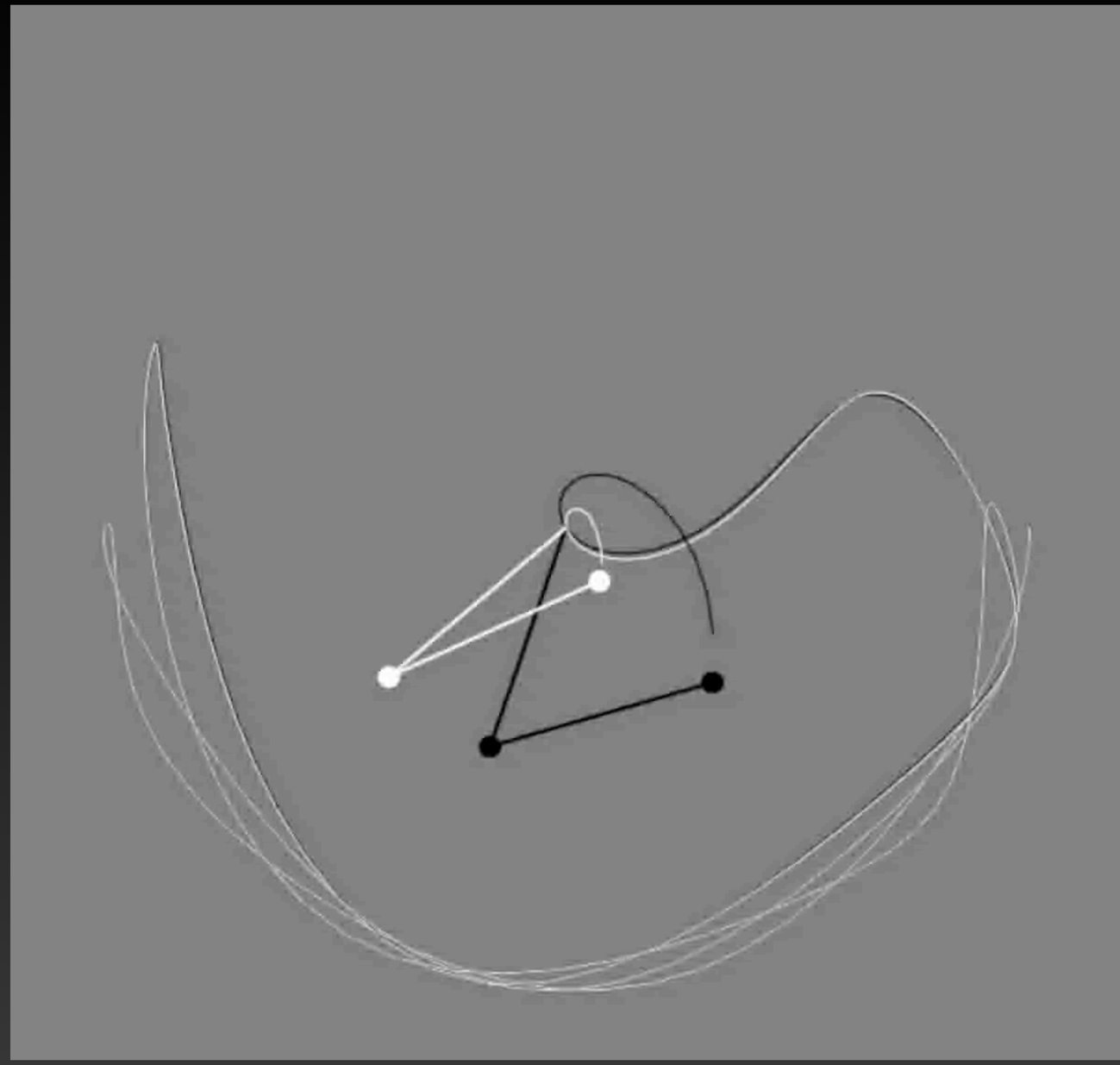
The Double Pendulum

- Play around with the initial conditions.
  - What happens if you start it so that both pendula are straight up?
  - Are you able to find any interesting behaviours?
  - Can you find a way to make the system numerically unstable? (Hint: Try increasing the velocity.)



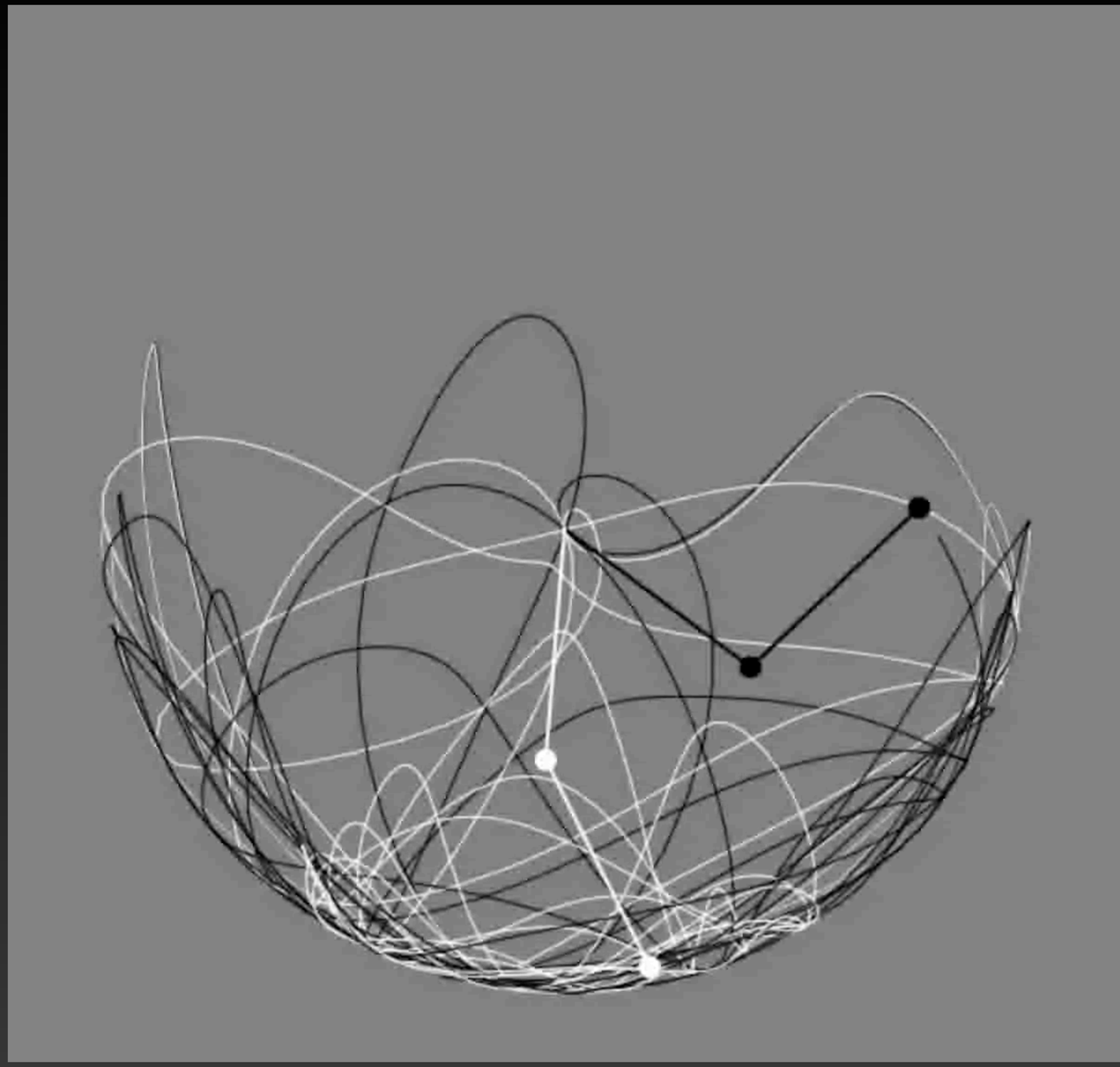
Chaos

- Small changes to the initial conditions can give very different results.
- The motion is determined, but not unpredictable.
- Small differences in initial conditions, such as errors in measurements or rounding errors in numerical computation, can yield widely diverging outcomes.



Chaos

- If we start the Double Pendulum with two almost identical initial conditions, after a short time their paths diverge. Code to play around with these can be found at
- JavaScript: https://editor.p5js.org/zyrxvo/sketches/cobz6b4Y7
- Python: https://trinket.io/library/trinkets/065cd7ab36



Thank you
Happy Coding

UNIVERSITY OF
TORONTO