Contents

Chapter 1

Sound

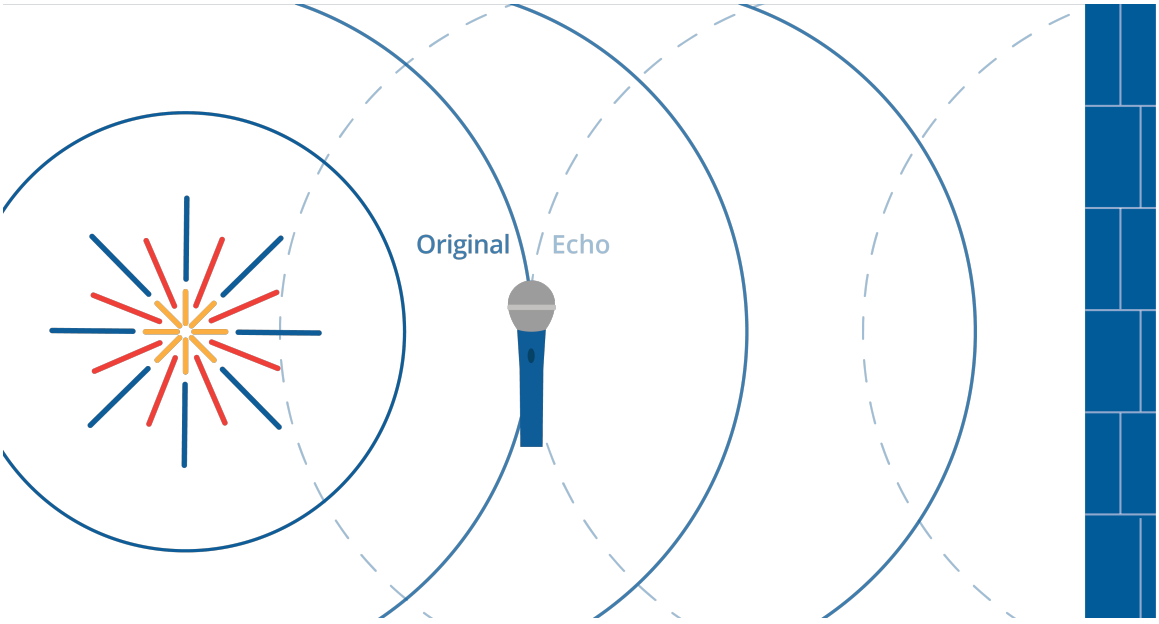
When you set off a firecracker, it makes a sound.

Let's break that down a little more. Inside the cardboard wrapper of the firecracker, there is potassium nitrate (KNO_3), sulfur (S), and carbon(C). These are all solids. When you trigger the chemical reactions with a little heat, these atoms rearrange themselves to be potassium carbonate (K_2CO_3), potassium sulfate (K_2SO_4), carbon dioxide (CO_2), and nitrogen (N_2). Note that the last two are gasses.

The molecules of a solid are much more tightly packed than the molecules of a gas. So after the chemical reaction, the molecules expand to fill a much bigger volume. The air molecules nearby get pushed away from the firecracker. They compress the molecules beyond them, and those compress the molecules beyond them.

This compression wave radiates out as a sphere; its radius growing at about 343 meters per second ("The speed of sound").

The energy of the explosion is distributed around the surface of this sphere. As the radius increases, the energy is spread more and more thinly around. This is why the firecracker seems louder when you are closer to it. (If you set off a firecracker in a sewer pipe, the sound will travel much, much farther.)



This compression wave will bounce off of hard surfaces. If you set off a firecracker 50 meters from a big wall, you will hear the explosion twice. We call the second one an "echo".

The compression wave will be absorbed by soft surfaces. If you covered that wall with pillows, there would be almost no echo.

The study of how these compression waves move and bounce is called *acoustics*. Before you build a concert hall, you hire an acoustician to look at your plans and tell you how to make it sound better.

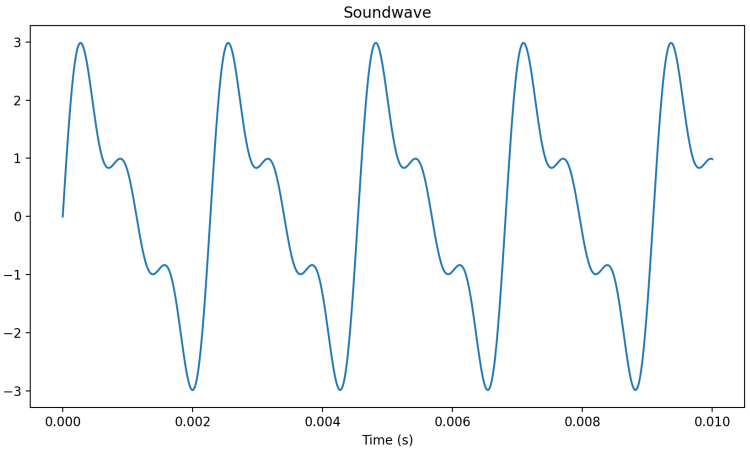
1.1 Pitch and frequency

The string on a guitar is very similar to the weighted spring example. The farther the string is displaced, the more force it feels pushing it back to equilibrium. Thus, it moves back and forth in a sine wave. (OK, it isn't a pure sine wave, but we will get to that later.)

The string is connected to the center of the boxy part of the guitar, which is pushed and pulled by the string. That creates compression waves in the air around it.

If you are in the room with the guitar, those compression waves enter your ear and push and pull your eardrum, which is attached to bones that move a fluid that tickles tiny hairs, called *cilia*, in your inner ear. This is how you hear.

We sometimes see plots of sound waveforms. The x-axis represents time. The y-axis represents the amount the air is compressed at the microphone that converted the air pressure into an electrical signal.



If the guitar string is made tighter (by the tuning pegs) or shorter (by the guitarist's fingers on the strings), the string vibrates more times per second. We measure the number of

waves per second and we call it the *frequency* of the tone. The unit for frequency is *Hertz*: cycles per second.

Musicians have given the different frequencies names. If the guitarist plucks the lowest note on his guitar, it will vibrate at 82.4 Hertz. The guitarist will say "That pitch is low E." If the string is made half as long (by a finger on the 12th fret), the frequency will be twice as fast (164.8 Hertz), and the guitarist will say "That is E an octave up."

For any note, the note that has twice the frequency is one octave up. The note that has half the frequency is one octave down.

The octave is a very big jump in pitch, so musicians break it up into 12 smaller steps. If the guitarist shortens the E string by one fret, the frequency will be $82.4 \times 1.059463 \approx 87.3$ Hertz.

Shortening the string one fret always increases the frequency by a factor of 1.059463. Why?

Because $1.059463^{12} = 2$. That is, if you take 12 of these hops, you end up an octave higher.

This, the smallest hop in western music, is referred to as a *half step*.

Exercise 1 Notes and frequencies

Working Space

The note A near the middle of the piano, is 440Hz. The note E is 7 half steps above A. What is its frequency?

Answer on Page 85

1.2 Chords and harmonics

Of course, a guitarist seldom plays only one string at a time. Instead, they use the frets to pick a pitch for each string and strums all six strings.

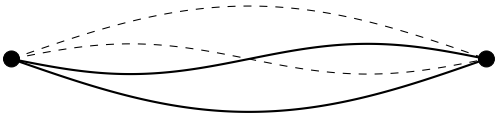
Some combinations of frequencies sound better than others. We have already talked about the octave: If one string vibrates twice for each vibration of another, they sound sweet together.

Musicians speak of "the fifth". If one string vibrates three times and the other vibrates twice in the same amount of time, they sound sweet together.

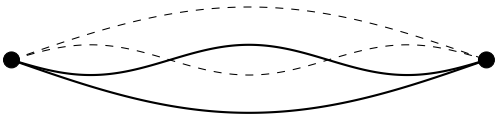
Likewise, if one string vibrates 4 times while the other vibrates 3 times, they sound sweet together. Musicians call this "the third."

Each of these different frequencies tickle different cilia in the inner ear, so you can hear all six notes at the same time when the guitarist strums their guitar.

When a string vibrates, it doesn't create a single sine wave. Yes, the string vibrates from end-to-end, and this generates a sine wave at what we call *the fundamental frequency*. However, there are also "standing waves" on the string. One of these standing waves is still at the centerpoint of the string, but everything to the left of the centerpoint is going up, while everything to the right is going down. This creates *an overtone* that is twice the frequency of the fundamental.



The next overtone has two still points — it divides the string into three parts. The outer parts are up, while the inner part is down. Its frequency is three times the fundamental frequency.



And so on. 4 times the fundamental, 5 times the fundamental, etc.

In general, tones with many overtones tend to sound bright. Tones with just the fundamental sound thin.

Humans can generally hear frequencies from 20Hz to 20,000Hz (or 20kHz). Young people tend to be able to hear very high sounds better than older people.

Dogs can generally hear sounds in the 65Hz to 45kHz range.

1.3 Making waves in Python

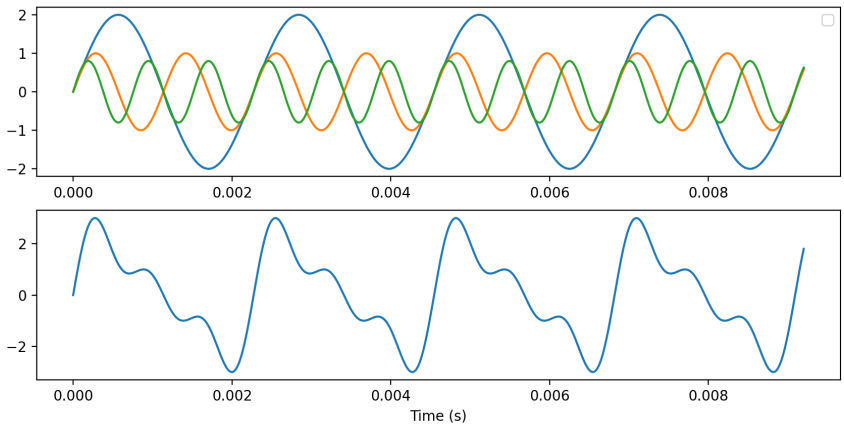
Let's make a sine wave and add some overtones to it. Create a file named `harmonics.py`.

```
import matplotlib.pyplot as plt
import math
```

```
# Constants: frequency and amplitude
fundamental_freq = 440.0 # A = 440 Hz
fundamental_amp = 2.0

# Up an octave
first_freq = fundamental_freq * 2.0 # Hz
first_amp = fundamental_amp * 0.5

# Up a fifth more
second_freq = fundamental_freq * 3.0 # Hz
second_amp = fundamental_amp * 0.4

# How much time to show
max_time = 0.0092 # seconds

# Calculate the values 10,000 times per second
time_step = 0.00001 # seconds

# Initialize
time = 0.0
times = []
totals = []
fundamentals = []
firsts = []
seconds = []

while time <= max_time:
    # Store the time
    times.append(time)

    # Compute value each harmonic
    fundamental = fundamental_amp * math.sin(2.0 * math.pi * fundamental_freq * time)
    first = first_amp * math.sin(2.0 * math.pi * first_freq * time)
    second = second_amp * math.sin(2.0 * math.pi * second_freq * time)

    # Sum them up
    total = fundamental + first + second

    # Store the values
    fundamentals.append(fundamental)
    firsts.append(first)
    seconds.append(second)
    totals.append(total)

    # Increment time
```

```
    time += time_step

# Plot the data
fig, ax = plt.subplots(2, 1)

# Show each component
ax[0].plot(times, fundamentals)
ax[0].plot(times, firsts)
ax[0].plot(times, seconds)
ax[0].legend()

# Show the totals
ax[1].plot(times, totals)
ax[1].set_xlabel("Time (s)")

plt.show()
```

When you run it, you should see a plot of all three sine waves and another plot of their sum:



1.3.1 Making a sound file

The graph is pretty to look at, but make let's a file that we can listen to.

The WAV audio file format is supported on pretty much any device, and a library for writing WAV files comes with Python. Let's write some sine waves and some noise into a WAV file.

Create a file called `soundmaker.py`

```
import wave
import math
import random

# Constants
frame_rate = 16000 # samples per second
duration_per = 0.3 # seconds per sound
frequencies = [220, 440, 880, 392] # Hz
amplitudes = [20, 125]
baseline = 127 # Values will be between 0 and 255, so 127 is the baseline
samples_per = int(frame_rate * duration_per) # number of samples per sound

# Open a file
wave_writer = wave.open('sound.wav', 'wb')

# Not stereo, just one channel
wave_writer.setnchannels(1)

# 1 byte audio means everything is in the range 0 to 255
wave_writer.setsampwidth(1)

# Set the frame rate
wave_writer.setframerate(frame_rate)

# Loop over the amplitudes and frequencies
for amplitude in amplitudes:
    for frequency in frequencies:
        time = 0.0
        # Write a sine wave
        for sample in range(samples_per):
            s = baseline + int(amplitude * math.sin(2.0 * math.pi * frequency * time))
            wave_writer.writeframes(bytes([s]))
            time += 1.0 / frame_rate

        # Write some noise after each sine wave
        for sample in range(samples_per):
            s = baseline + random.randint(0, 15)
            wave_writer.writeframes(bytes([s]))

# Close the file
wave_writer.close()
```

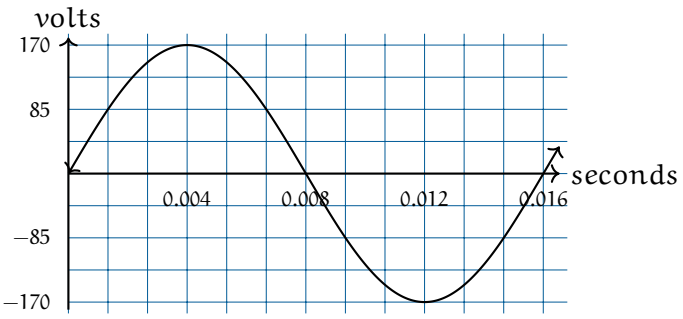
When you run it, it should create a sound file with several tones of different frequencies and volumes. Each tone should be followed by some noise.

CHAPTER 2

Alternating Current

We have discussed the voltage and current created by a battery. A battery pushes the electrons in one direction at a constant voltage; this is known as *Direct Current* or DC. A battery typically provides between 1.5 and 9 volts.

The electrical power that comes into your home on wires is different. If you plotted the voltage over time, it would look like this:



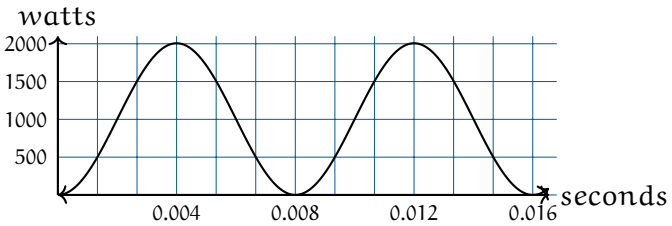
The x axis here represents ground. When you insert a two-prong plug into an outlet, one is "hot" and the other is "ground". Ground represents 0 volts and should be the same voltage as the dirt under the building.

The voltage is a sine wave at 60Hz. Your voltage fluctuates between -170v and 170v. Think for a second what that means: The power company pushes electrons at 170v, then pulls electrons at 170v. It alternates back and forth this way 60 times per second.

2.1 Power of AC

Let's say you turn on your toaster, which has a resistance of 14.4 ohms. How much energy (in watts) does it change from electrical energy to heat? We know that $I = V/R$ and that watts of power are IV. So, given a voltage of V, the toaster is consuming V^2/R watts.

However, V is fluctuating. Let's plot the power the toaster is consuming:



Another sine wave! Here is a lesser-known trig identity: $(\sin(x))^2 = \frac{1}{2} - \frac{1}{2}\cos(2x)$

This is actually a cosine wave flipped upside down, scaled down by half the peak power, and translated up so that it is never negative. Note that it is also twice the frequency of the voltage sine wave.

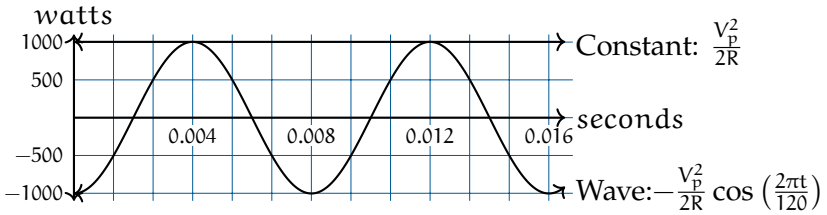
If we say the peak voltage is V_p and the resistance of the toaster is R, the power is given by

$$\frac{V_p^2}{2R} - \frac{V_p^2}{2R}\cos\left(\frac{2\pi t}{120}\right)$$

As a toaster user and as someone who pays a power bill, you are mostly interested in the average power. To get the average power, you take the area under the power graph and divide it by the amount of time.

We can think of the area under the curve as two easy-to-integrate quantities summed:

- A constant function of $y = frac V_p^2 2R$
- A wave $y = -\frac{V_p^2}{2R}\cos\left(\frac{2\pi t}{120}\right)$



When we integrate that constant function, we get $\frac{tV_p^2}{2R}$.

When we integrate that wave for a complete cycle we get...zero! The positive side of the wave is canceled out by the negative side.

So, the average power is $\frac{V_p^2}{2R}$ watts.

Someone at some point said, "I'm used to power being V^2/R. Can we define a voltage measure for AC power such that this is always true?"

So we started using V_{rms} which is just $\frac{V_p}{\sqrt{2}}$. If you look on the back of anything that plugs into a standard US power outlet, it will say something like "For 120v". What they mean is, "For 120v RMS, we expect the voltage to fluctuate back and forth from 170v to -170v."

Notice that this is the same Root-Mean-Squared that we defined earlier, but now we know that if $y = \sin(x)$, the RMS of y is $1/\sqrt{2} \approx 0.707$.

For current, we do the same thing. If the current is AC, the power consumed by a resistor is $I_{RMS}^2 R$, where I_{RMS} is the peak current divided by $sqrt2$.

2.2 Power Line Losses

A wire has some resistance. Thinner wires tend to have more resistance than thicker ones. Aluminum wires tend to have more resistance than copper wires.

Let's say that the power that comes to your house has to travel 20 km from the generator in a cable that has about 1Ω of resistance per km. Let's also say that your home is consuming 12 kilowatts of power.

If that power is 120v RMS from the generator to your home, what percentage of the power is lost heating the power line? 10 amps RMS flow through your home. When that current goes through the wire, $I^2R = (100)(20) = 2000watts$ is lost to heat.

This means the power company would need to supply 14 kilowatts of power, knowing that 2 kilowatts would be lost on the wires.

What if the power company moved the power at 120,000 volts RMS? Now only 0.01 amps RMS flow through your home. When that current goes through the wire $I^2R = (0.0001)(20) = .002$ watts of power are lost on the power lines.

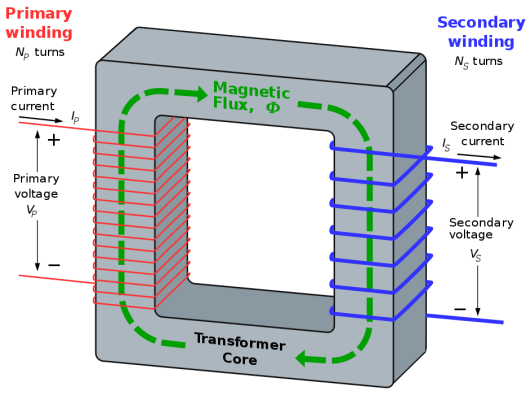
This is much, much more efficient. The only problem is that 120,000 volts would be incredibly dangerous. So the power company moves power long distances at very high voltages, like 765 kV. Before the power is brought into your home, it is converted into a lower voltage using a *transformer*.

2.3 Transformers

A transformer is a device that converts electrical power from one voltage to another. A good transformer is more than 95% efficient. The details of magnetic fields, flux, and inductance are beyond the scope of this chapter, so we are going to give a relatively

simple (and admittedly incomplete) explanation for now.

A transformer is a ring with two sets of coils wrapped around it.



(Diagram from Wikipedia)

When alternating current is run through the primary winding, it creates magnetic flux in the ring. The magnetic flux induces current in the secondary winding.

If V_P is the voltage across the primary winding and V_S is the voltage across the secondary winding, they are related by the following equation:

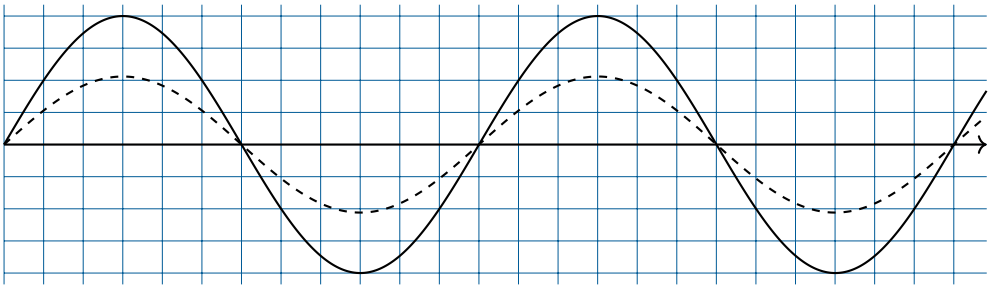
$$\frac{V_P}{V_S} = \frac{N_P}{N_S}$$

where N_P and N_S are the number of turns in the primary and secondary windings.

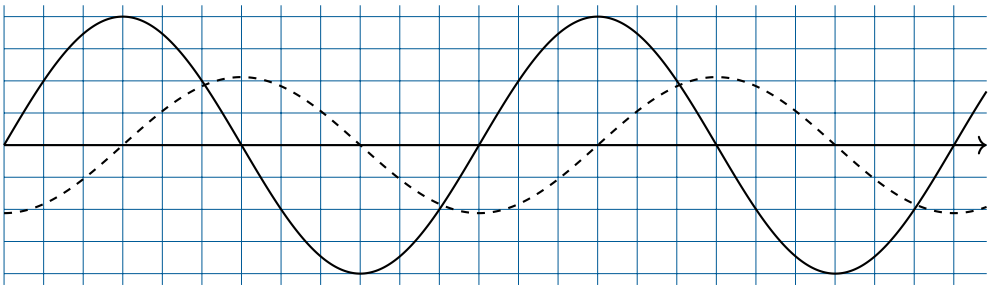
There are usually at least two transformers between you and the very high voltage lines. There are transformers at the substation that make the voltage low enough to travel on regular utility poles. On the utility poles, you will see cans that contain smaller transformers. Those step the voltage down to make the power safe to enter your home.

2.4 Phase and 3-phase power

If two waves are “in sync”, we say they have the same *phase*.

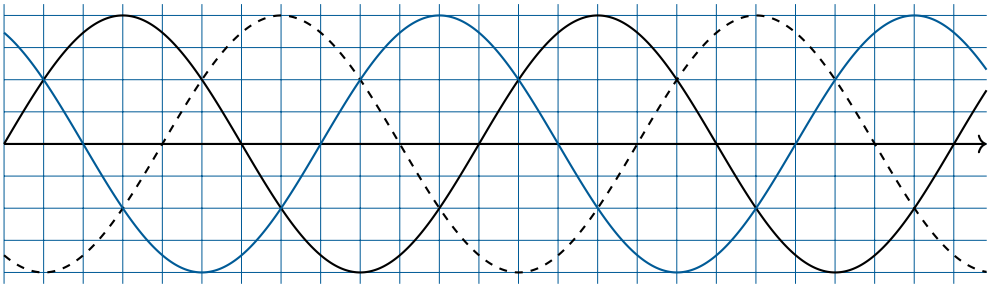


If they are the same frequency, but are not in sync, we can talk about the difference in their phase.



Here, we see that the smaller wave is lagging by $\pi/2$ or $90°$.

In most power grids, there are usually three wires carrying the power. The voltage on each is $2\pi/3$ out of phase with the other two:



This is nice in two ways:

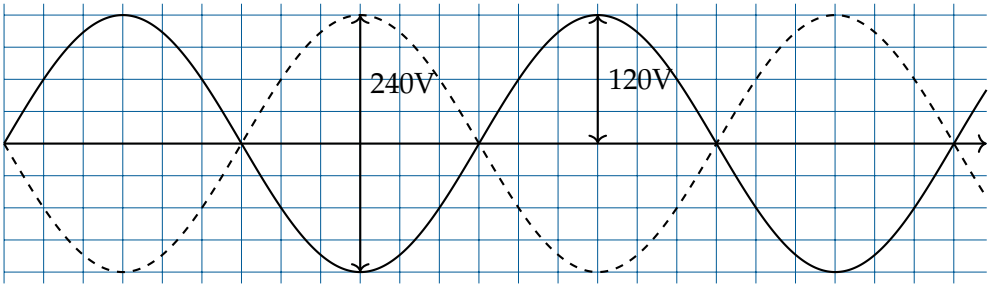
- While the power in each wire is fluctuating, the total power is not fluctuating at all.
- While the power plant is pushing and pulling electrons on each wire, the total number number of electrons leaving the load is zero.

(Both these assume that there each wire is attached to a load with the same constant resistance.)

In big industrial factories, you will see all three wires enter the building. Large amounts

of smooth power delivery means a great deal to an industrial user.

In residential settings, each home gets its power from one of the three wires. However, two wires typically carry power into the home. Each one carries 120V RMS, but they are out of phase by 180 degrees. Lights and small appliances are connected to one of the wires and ground, so they get 120V RMS. Large appliances, like air conditioners and washing machines, are connected across the two wires, so they get 240V RMS.

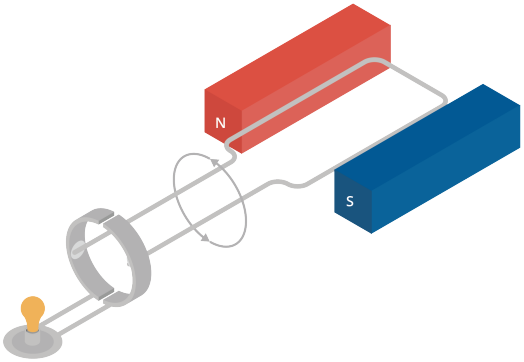


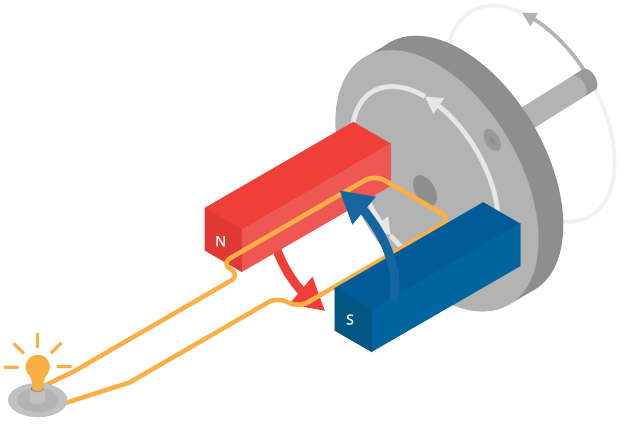
How do you get two circuits, 180 degrees out of phase, from one circuit? Using a center-tap transformer.

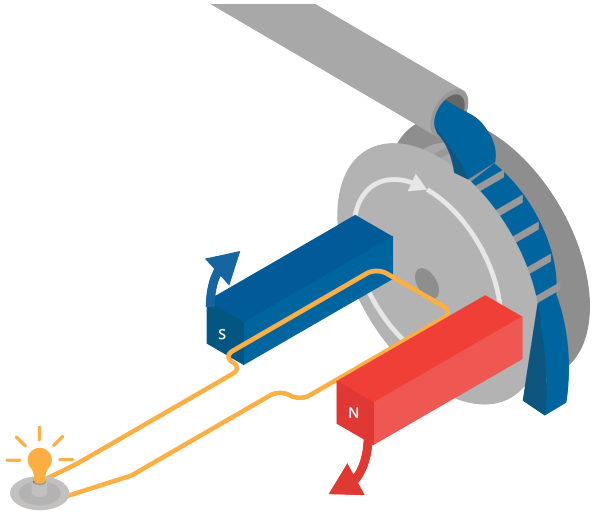
FIXME: Diagram here

Chapter 3

Electric Motor





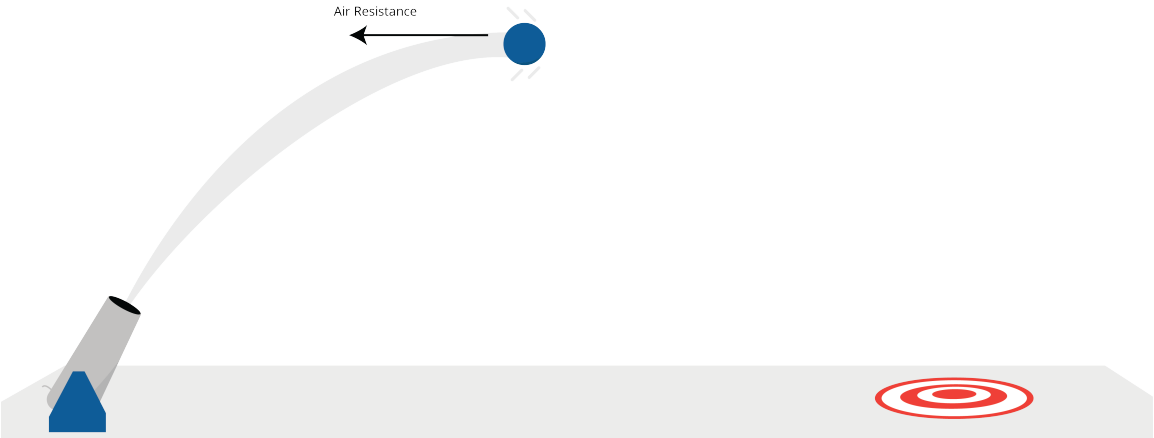

S
N

Chapter 4

Drag

The very first computers were created to do calculations of how artillery would fly when shot at different angles. The calculations were similar to the ones you just did for the flying hammer, with two important differences:

- They were interested in two dimensions: the height and the distance across the ground.
- However, artillery flies a lot faster than a hammer, so they also had to worry about drag from the air.



4.1 Wind resistance

The first thing they did was put one of the shells in a wind tunnel. They measured how much force was created when they pushed 1 m/s of wind over the shell. Let's say it was 0.1 newtons.

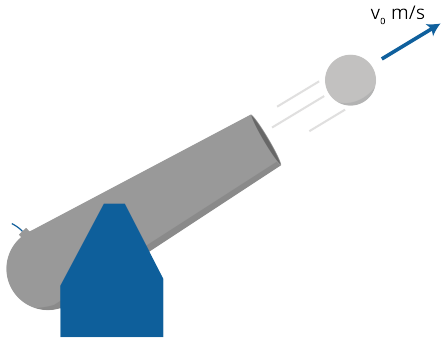
One of the interesting things about the drag from the air (often called *wind resistance*) is that it increases with the *square* of the speed. Thus, if the wind pushing on the shell is 3 m/s, instead of 1 m/s, the resistance is $3^2 \times 0.1 = 0.9$ newtons.

(Why? Intuitively, three times as many air molecules are hitting the shell and each molecule is hitting it three times harder.)

So, if a shell is moving with the velocity vector v, the force vector of the drag points in the exact opposite direction. If μ is the force of wind resistance of the shell at 1 m/s, then the magnitude of the drag vector is $\mu|v|^2$.

4.2 Initial velocity and acceleration due to gravity

Let's say a shell is shot out of a tube at s m/s, and the tube is tilted θ radians above level.



The initial velocity will be given by the vector $[s\cos(\theta), s\sin(\theta)]$

(The velocity of the shell is actually a 3-dimensional vector, but we are only going to worry about height and horizontal distance; we are assuming that the operator pointed it in the right direction.)

To figure out the path of the shell, we need to compute its acceleration. We remember that

$$F = ma$$

(Note that F and a are vectors.) Dividing both sides by m, we get:

$$a = \frac{F}{m}$$

Let's figure out the net force on the shell, so that we can calculate the acceleration vector.

If the shell has a mass of b, the force due to gravity will be in the downward direction, with a magnitude of $9.8b$ newtons.

To get the net force, we will need to add the force due to gravity with the force due to wind resistance.

4.3 Simulating artillery in Python

Create a file called `artillery.py`.

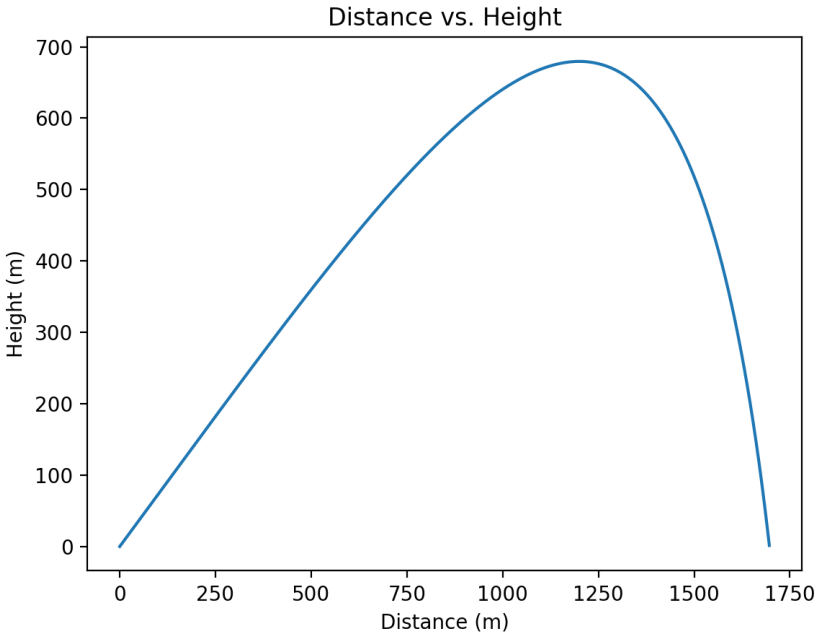
```
import numpy as np
import matplotlib.pyplot as plt

# Constants
mass = 45 # kg
start_speed = 300.0 # m/s
theta = np.pi/5 # radians (36 degrees above level)
time_step = 0.01 # s
wind_resistance = 0.05 # newtons in 1 m/s wind
force_of_gravity = np.array([0.0, -9.8 * mass]) # newtons

# Initial state
position = np.array([0.0, 0.0]) # [distance, height] in meters
velocity = np.array([start_speed * np.cos(theta), start_speed * np.sin(theta)])
time = 0.0 # seconds

# Lists to gather data
distances = []
heights = []
times = []

# While shell is aloft
while position[1] >= 0:
    # Record data
    distances.append(position[0])
    heights.append(position[1])
    times.append(time)

    # Calculate the next state
    time += time_step
    position += time_step * velocity

    # Calculate the net force vector
    force = force_of_gravity - wind_resistance * velocity**2

    # Calculate the current acceleration vector
    acceleration = force / mass
```

```
        # Update the velocity vector
        velocity += time_step * acceleration

    print(f"Hit the ground {position[0]:.2f} meters away at {time:.2f} seconds.")

    # Plot the data
    fig, ax = plt.subplots()
    ax.plot(distances, heights)
    ax.set_title("Distance vs. Height")
    ax.set_xlabel("Distance (m)")
    ax.set_ylabel("Height (m)")
    plt.show()
```

When you run it, you should get a message like:

```
Hit the ground 1696.70 meters away at 20.73 seconds.
```

You should also see a plot of the shell's path:



4.4 Terminal velocity

If you shot the shell very, very high in the sky, it would keep accelerating toward the ground until the force of gravity and the force of the wind resistance were equal. The speed at which this happens is called the *terminal velocity*. The terminal velocity of a falling human is about 53 m/s.

Exercise 2 Terminal velocity

What is the terminal velocity of the shell described in our example?

Working Space

Answer on Page 85

CHAPTER 5

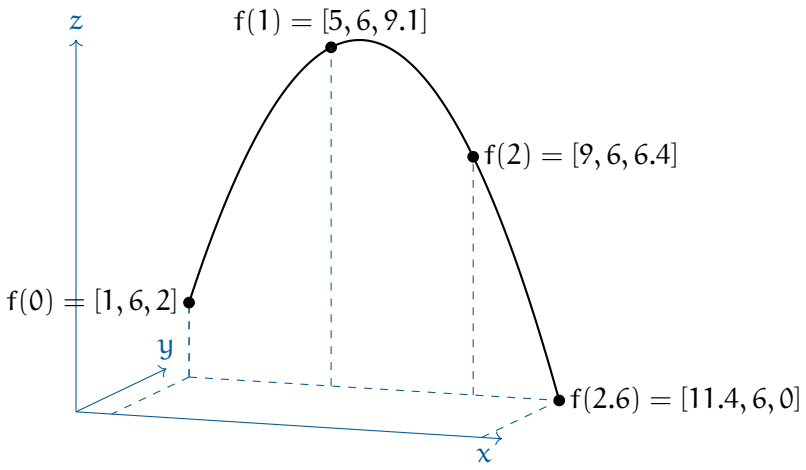
Vector-valued Functions

In the last chapter, you calculated the flight of the shell. For any time t, you could find a vector $[\texttt{distance}, \texttt{height}]$. This can be thought of as a function f that takes a number and returns a 2-dimensional vector. We call this a *vector-valued* function from $\mathbb{R} \to \mathbb{R}^2$.

We often make a vector-valued function by defining several real-valued functions. For example, if you threw a hammer with an initial upward speed of 12 m/2 and a horizontal speed of 4 m/s along the x axis from the point $(1, 6, 2)$, its position at time t (during its flight) would be given by:

$$f(t) = [4t + 1, 6, -4.8t^2 + 12t + 2]$$

In other words, x is increasing with t, y is constant, and z is a parabola.



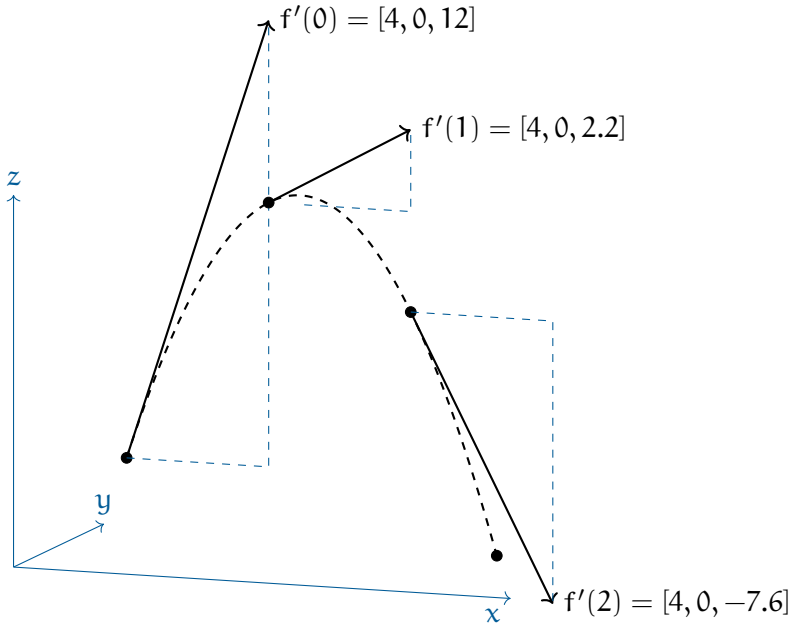
5.1 Finding the velocity vector

Now that we have its position vector, we can differentiate each component separately to get its velocity as a vector-valued function:

$$f'(t) = [4, 0, -9.8t + 12]$$

In other words, the velocity is constant along the x-axis, zero along the y-axis, and de-

creasing with time along the z axis.

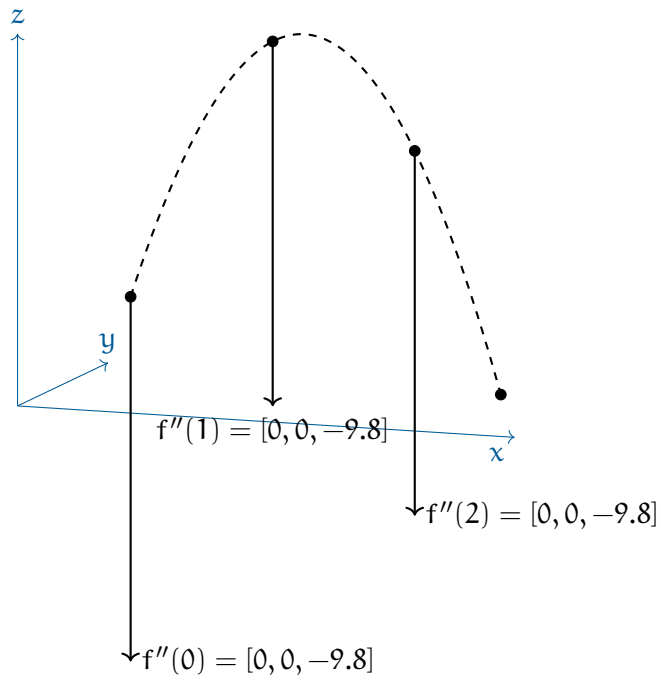


5.2 Finding the acceleration vector

Now that we have its velocity, we can get its acceleration as a vector-valued function:

$$f''(t) = [0, 0, -9.8]$$

There is no acceleration along the x or y axes. It is accelerating down at a constant $9.8 m/s^2$.


z
y
x
f″(1) = [0, 0, −9.8]
f″(2) = [0, 0, −9.8]
f″(0) = [0, 0, −9.8]

CHAPTER 6

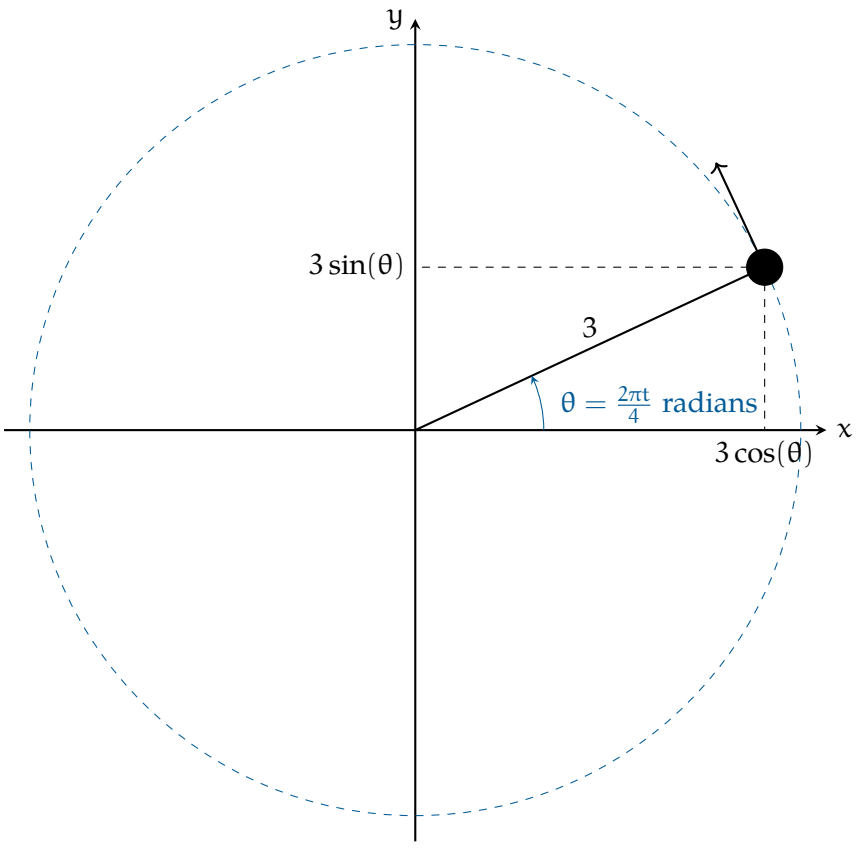
Circular Motion

Let's say you tie a 0.16 kg billard ball to a long string and begin to swing it around in a circle above your head. The string is 3 meters long, and the ball returns to where it started every 4 seconds. If you start your stopwatch as the ball crosses the x-axis, the position of the ball at any time t given by:

$$p(t) = [3\cos\left(\frac{2\pi}{4}t\right), 3\sin\left(\frac{2\pi}{4}t\right), 2]$$

(This assumes that the ball would be going counter-clockwise if viewed from above. The spot you are standing on is considered the origin $[0, 0, 0]$.)

Notice that the height is a constant — 2 meters in this case. That isn't very interesting, so we will talk just about the first two components. Here is what it would look like from above:



In this case, the radius, r, is 3 meters. The period, T, is 4 seconds. In general, we say that circular motion is given by:

$$p(t) = \left[r\cos\frac{2\pi t}{T}, r\cos\frac{2\pi t}{T}\right]$$

A common question is "How fast is it turning right now?" If you divide the 2π radians of a circle by the 4 seconds it takes, you get the answer "About 1.57 radians per second." This is known as *angular velocity* and we typically represent it with the lowercase Omega: ω. (Yes, it looks a lot like a "w".) To be precise, in our example, the angular velocity is $\omega = \frac{\pi}{2}$.

Notice that this is different from the question "How fast is it going?" This ball is traveling the circumference of $6\pi \approx 18.85$ meters every 4 seconds. This means the speed of the ball is about 4.71 meters per second.

6.1 Velocity

The velocity of the ball is a vector, and we can find that vector by differentiating each component of the position vector.

For any constants a and b:

Expression	Derivative
$a\sin bt$	$ab\cos bt$
$a\cos bt$	$-ab\sin bt$

Thus, in our example, the velocity of the ball at any time t is given by:

$$v(t) = \left[-\frac{3(2\pi)}{4}\sin\frac{2\pi t}{4}, \frac{3(2\pi)}{4}\cos\frac{2\pi t}{4}, 0\right]$$

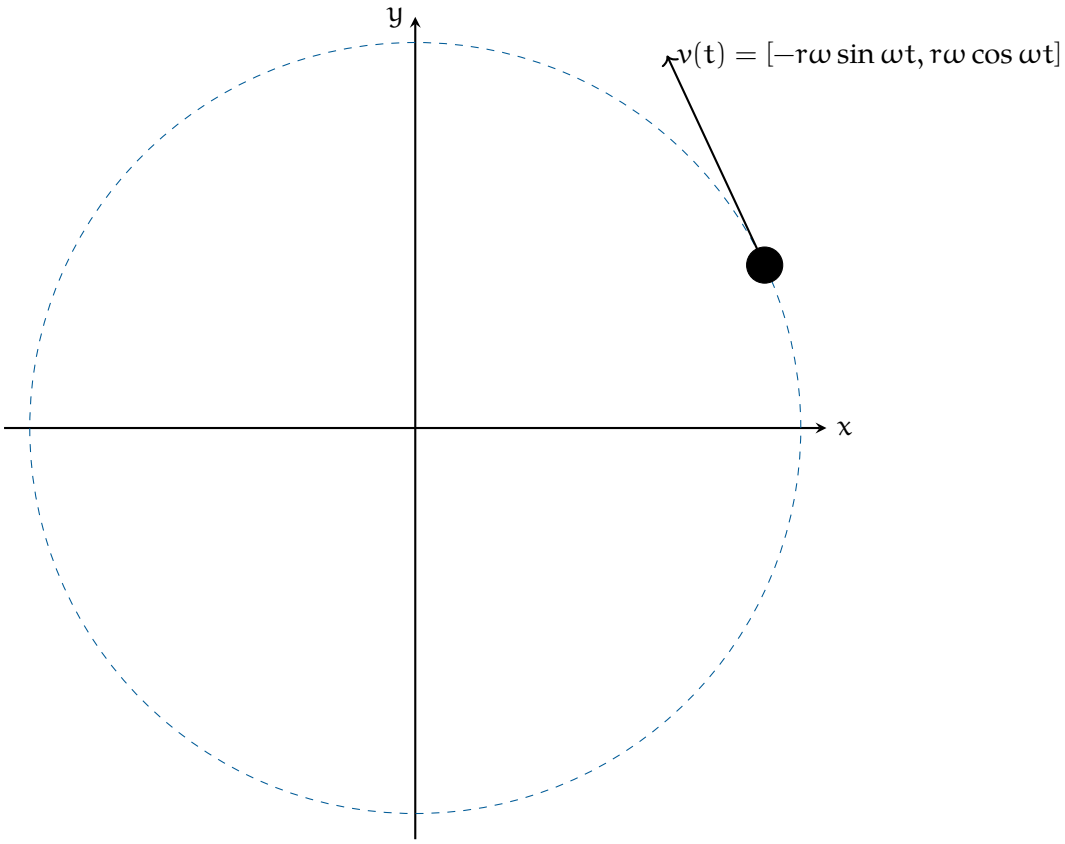
Notice that the velocity vector is perpendicular to the position vector. It has a constant magnitude.

In general, an object traveling in a circle at a constant speed has the velocity vector:

$$v(t) = [-r\omega\sin\omega t, r\omega\cos\omega t]$$

where $t = 0$ is the time that it crosses the x axis. If $\omega$is negative, that means the motion would be clockwise when viewed from above.

The magnitude of the velocity vector is $r\omega$.

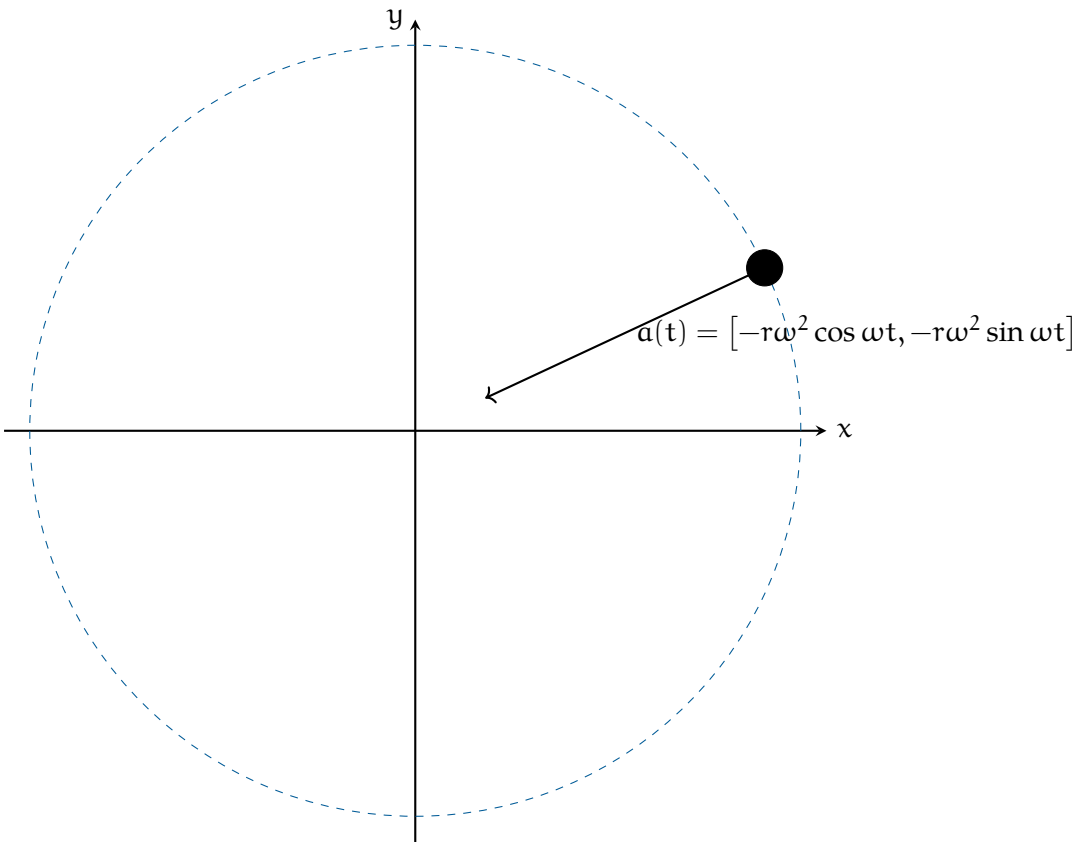


6.2 Acceleration

We can get the acceleration by differentiating the components of the velocity vector.

$$a(t) = \left[-r\omega^2 \cos \omega t, -r\omega^2 \sin \omega t\right]$$

Notice that the acceleration vector points toward the center of the circle it is traveling on. That is, when an object is traveling on a circle at a constant speed, its only acceleration is toward the center of the circle.



The magnitude of the acceleration vector is $r\omega^2$.

6.3 Centripetal force

How hard is the ball pulling against your hand? That is, if you let go, the ball would fly in a straight line. The force you are exerting on the string is what causes it to accelerate toward the center of the circle. We call this the *centripetal force*.

Recall that $F = ma$. The magnitude of the acceleration is $r\omega^2 = 3\left(\frac{2pi}{4}\right)^2 \approx 7.4$ m/s. The mass of the ball is 0.16 kg. So, the force pulling against your hand is about 1.18 newtons.

The general rule is that when something is traveling in a circle at a constant speed, the centripetal force needed to keep it traveling in a circle is:

$$F = mr\omega^2$$

If you know the radius r and the speed v of the object, here is the rule:

$$F = \frac{mv^2}{r}$$

Exercise 3 Circular Motion

Just as your car rolls onto a circular track with a radius of 200 m, you realize your 0.4 kg cup of coffee is on the slippery dashboard of your car. While driving 120 km/hour, you hold the cup to keep it from sliding.

What is the maximum amount of force you would need to use? (The friction of the dashboard helps you, but the max is when the friction is zero.)

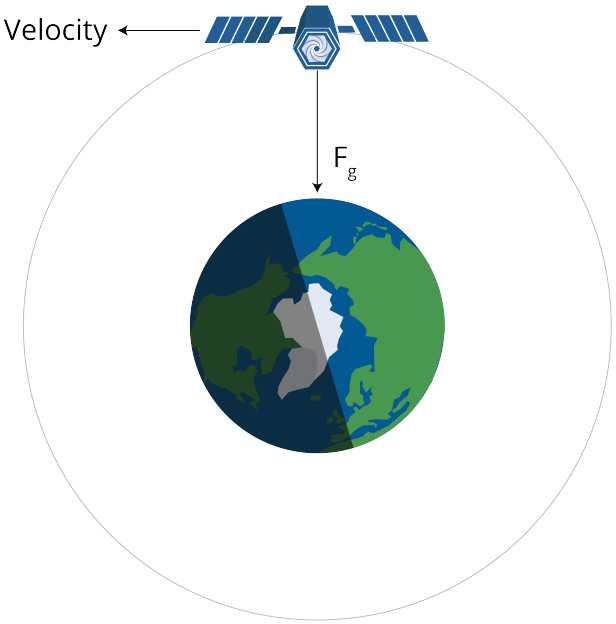
Working Space

Answer on Page 85

CHAPTER 7

Orbits

A satellite stays in orbit around the planet because the pull of the planet's gravity causes it to accelerate toward the center of the planet.



The satellite must be moving at a very particular speed to keep a constant distance from the planet — to travel in a circular orbit. If it is moving too slowly, it will get closer to the planet. If it is going too fast, it will get farther from the planet.

The radius of the earth is about 6.37 million meters. A satellite that is in a low orbit is typically about 2 million meters above the ground. At that distance, the acceleration due

to gravity is more like $6.8 m/s^2$, instead of the $9.8 m/s^2$ that we experience on the surface of the planet.

How fast does the satellite need to be moving in a circle with a radius of 8.37 million meters to have an acceleration of $6.8 m/s^2$? Real fast.

Recall that the acceleration vector is

$$a = \frac{v^2}{r}$$

Thus the velocity v needs to be:

$$v = \sqrt{ar} = \sqrt{6.8(8.37x10^6} = 7,544 \text{ m/s}$$

(That's 16,875 miles per hour.)

When a satellite falls out of orbit, it enters the atmosphere at that 7,544 m/s. The air rushing by generates so much friction that the satellite gets very, very hot, and usually disintegrates.

7.1 Astronauts are not weightless

Some people see astronauts floating inside an orbiting spacecraft and think there is no gravity: that the astronauts are so far away that the gravity of the planet doesn't affect them. This is incorrect. The gravity might be slightly less (Maybe 6 newtons per kg instead of 9.8 newtons per kg), but the weightless they experience is because they and the spacecraft is in free fall. They are just moving so fast (in a direction perpendicular to gravity) that they don't collide with the planet.

Exercise 4 Mars Orbit

Working Space

The radius of Mars is 3.39 million meters. The atmosphere goes up another 11 km. Let's say you want to put a satellite in a circular orbit around Mars with a radius of 3.4 million meters.

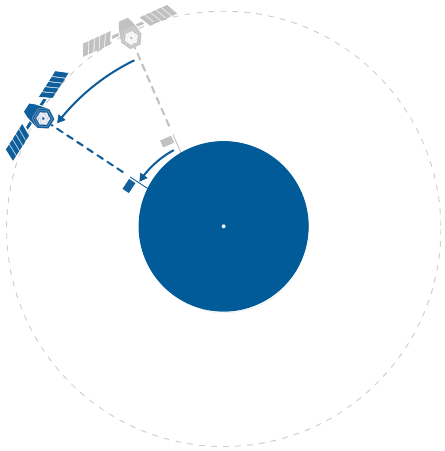
The acceleration due to gravity on the surface of Mars is $3.721 \mathrm{m/s^2}$. We can safely assume that it is approximately the same 11 km above the surface.

How fast does the satellite need to be traveling in its orbit? How long will each orbit take?

Answer on Page 85

7.2 Geosynchronous Orbits

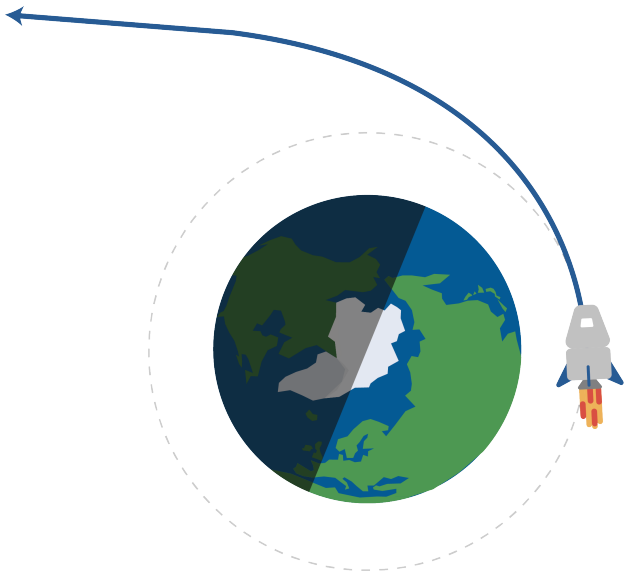
The planet earth rotates once a day. Satellites in low orbits circle the earth many times a day. Satellites in very high orbits circle less than once per day. There is a radius at which a satellite orbits exactly once per day. Satellites at this radius are known as "geosynchronous" or "geostationary", because they are always directly over a place on the planet.



The radius of a circular geosynchronous orbit is 42.164 million meters. (About 36 km above the surface of the earth.)

A geosynchronous satellite travels at a speed of 3,070 m/s.

Geosynchronous satellites are used for the Global Positioning Satellite system, weather monitoring system, and communications system.



FIXME: Add text for escape velocity

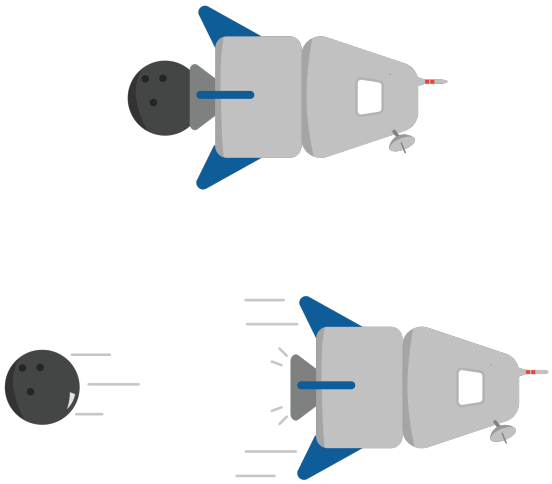
CHAPTER 8

Rocketry

Rockets propel hot gases, which recreates an equal and opposite reaction that pushes it forwards, even in a vacuum.

Even without anything to push against, the rocket can still move forward thanks to Newton's Third Law.

Imagine a spacecraft with a bowling ball attached to the back. If that spacecraft exerts a force to throw the bowling ball backwards, the ball will exert a force on the ship, moving it forwards.

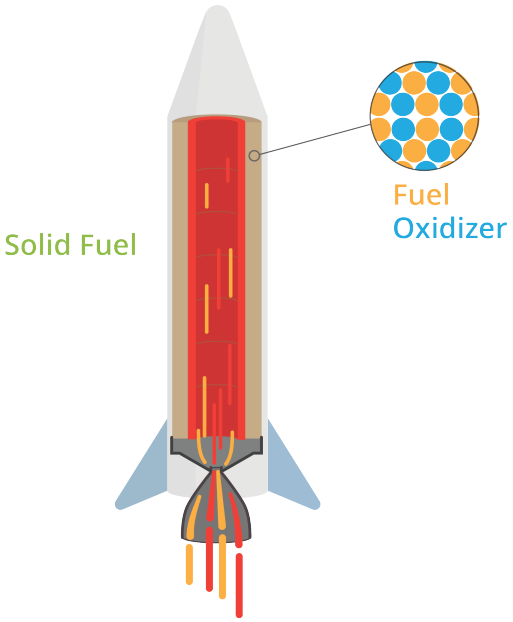


Instead of a bowling ball, real-life rockets usually "throw" particles of hot gas at very high speeds. Rockets carry their own oxidizer to provide oxygen to allow fuel to burn.

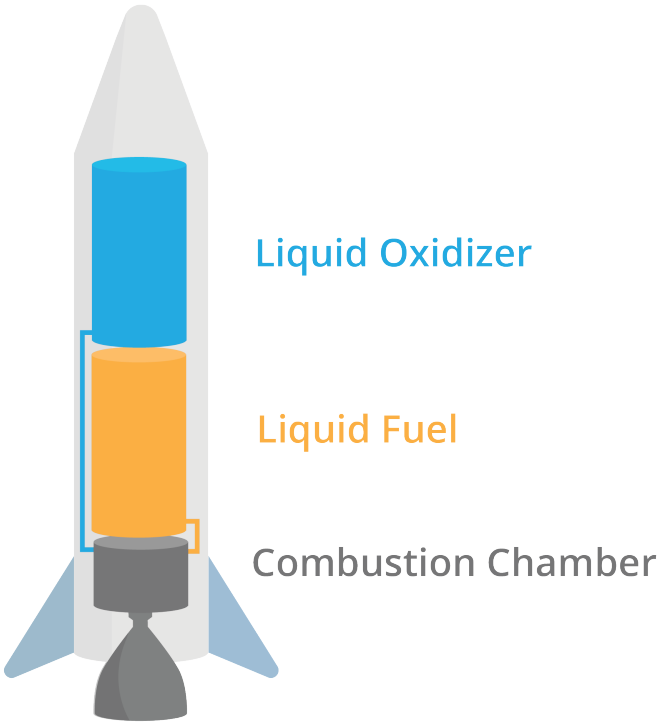
8.1 Types of rocket motors

There are two main types of chemical rockets.

One type is a *Solid Fuel Rocket*, which ignites a solid fuel-oxidizer mix. Once the solid fuel is ignited, it can't be stopped until all of the fuel is exhausted.



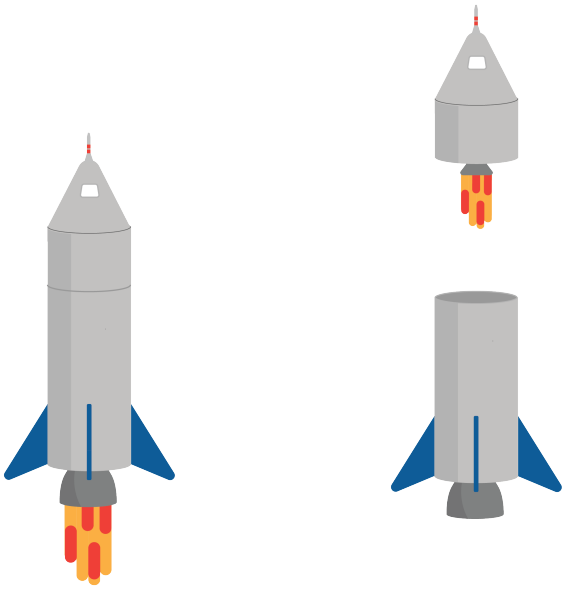
The other main type of chemical rocket is called a *Liquid Fuel Rocket*. Liquid fuel rockets contain separate tanks for liquid fuel and liquid oxygen. Fuel pumps bring them both to a combustion chamber where they ignite and exit the rocket. Most liquid fuel engines can control their thrust.



8.2 Tyranny of the rocket equation

Chemical rockets can only burn the fuel that they bring with them. However, the more fuel you carry, the heavier the vehicle will be.

One way to help reduce this weight is by using *staging*.



Staging allows rockets to drop unnecessary structural mass once they've used up a certain amount of fuel.

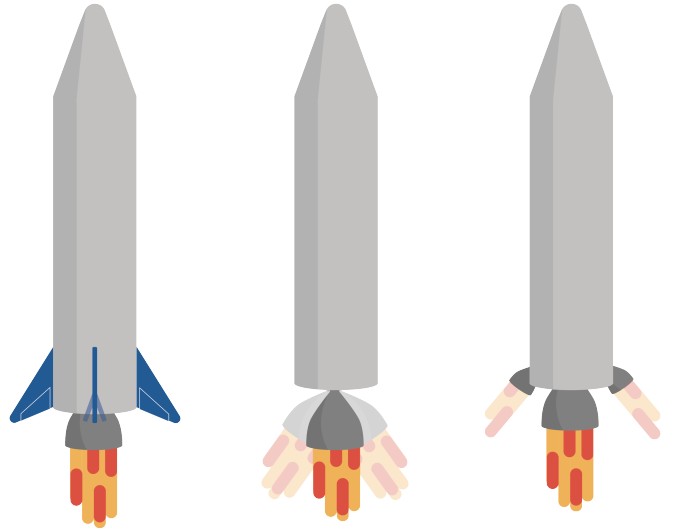
8.3 Control in atmosphere

There are several common ways that engineers have managed to control rockets' direction in the atmosphere. Usually, on-board sensors detect the orientation of the rocket, and can automatically adjust these controls to keep the rocket going the correct direction.

One method is using *movable fins*. The fins work similarly to control surfaces that we covered in the airplanes chapter.

Another method of control uses a *gimbaled engine*. [pros and cons]

A more outdated method is using *vernier engines*, which are two smaller engines that control attitude. However, this adds a large amount of weight to the rocket, so they are less frequently used today. [pros and cons]



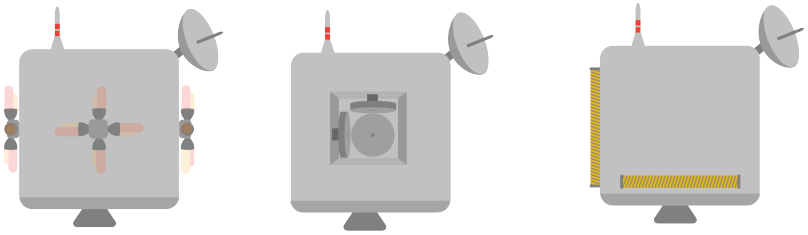
8.4 Control in space

The previous section describes ways that engineers control rockets in the atmosphere, but most rockets will end up in the vacuum of space. There are several common ways to adjust the orientation in space.

One method is using *RCS thrusters*. An RCS, or reaction control system, is a series of small thrusters that are used to change the direction and position of a spacecraft.

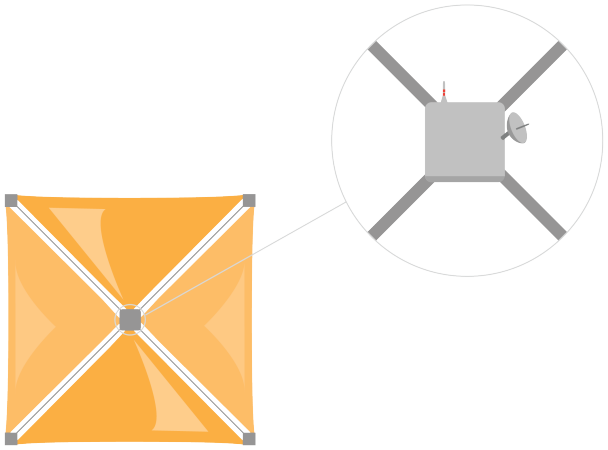
Another method called *reaction wheels* uses angular momentum to rotate the spacecraft. By accelerating and decelerating wheels on three axes, the spacecraft can rotate in any direction.

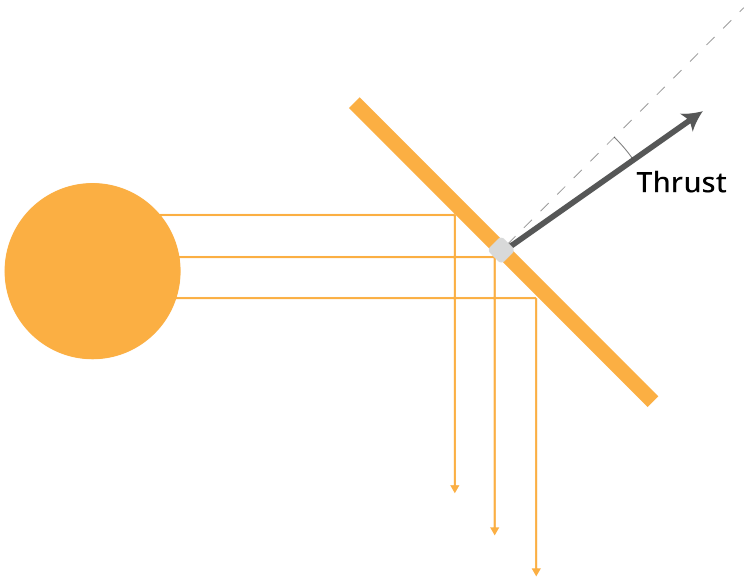
A third common attitude control technology is *magnetorquer*. Magnetorquers use electromagnets and the earth's magnetic field to adjust the orientation of the spacecraft.



8.5 Alternative propulsion

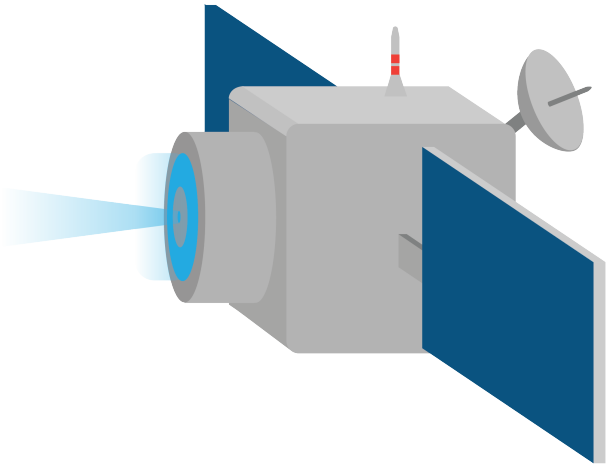
One type of alternate propulsion is called a *solar sail*. Solar sails use lightweight reflective surfaces to use photons in space to propel the spacecraft without on-board fuel.





Photons reflect off of the surface of the sail. However, since the surface is not perfectly reflective, some of those photons are absorbed, and they produce a horizontal equal and opposite reaction. That small force causes the net thrust to be slightly skewed away from a right angle to the sail.

ion propulsion



Chapter 9

Simulation with Vectors

In an earlier chapter, you wrote a python program that simulated the flight of a hammer to predict its altitude. Your simulation dealt only with scalars. Now, you are ready to create simulations of positions, velocities, accelerations, and forces as vectors.

In this chapter, you are going to simulate two moons that, as they wandered through the vast universe, get caught in each other's gravity well. We will assume there are no other forces acting upon the moons.

9.1 Force, Acceleration, Velocity, and Position

We talked about the magnitude of a gravitational attraction between two masses:

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

where F is the magnitude of the force in newtons, m_1 and m_2 are the masses in kg, r is the distance between them in meters, and g is the universal gravitational constant: 6.67430×10^{-11}.

What is the direction? For the two moons, the force on moon 1 will pull toward moon 2. Likewise, the force on moon 2 will pull toward moon 1.

Of course, if something is big (like the sun), you need to be more specific: The force points directly at the center of mass of the object that is generating the force.

Each of the moons will start off with a velocity vector. That velocity vector will change over time as the moon is accelerated by the force of gravity. If you have a mass m with an initial velocity vector of $\vec{v_0}$ that is being accelerated with a constant force vector $\vec{F}$, at time t, the new velocity vector will be:

$$\vec{v_t} = \vec{v_0} + \frac{t}{m}\vec{F}$$

If an object is at an initial position vector of $\vec{p_0}$ and moves with a constant velocity vector $\vec{v}$ for time t, the new position will be given by

$$\vec{p}_t = \vec{p}_0 + t\vec{v}$$

9.2 Simulations and Step Size

As two moons orbit each other, the force, acceleration, velocity, and position are changing smoothly and continuously. It is difficult to simulate truly continuous things on a digital computer.

However, think about how a movie shows you many frames each second. Each frame is a still picture of the state of the system. The more frames per second, the smoother it looks.

We do a similar trick in simulations. We say "We are going run our simulation in 2 hour steps. We will assume that the acceleration and velocity were constant for those two hours. We will update our position vectors accordingly, then we will recalculate our acceleration and velocity vectors."

Generally, as you make the step size smaller, your simulation will get more accurate and take longer to execute.

9.3 Make a Text-based Simulation

To start, you are going to write a Python program that simulates the moons and prints out their position for every time step. Later, we will add graphs and even animation.

We are going to assume the two moons are traveling the same plane so we can do all the math and graphing in 2 dimensions.

Each moon will be represented by a dictionary containing the state of the moon:

- Its mass in kilograms
- Its position — A 2-dimensional vector represent x and y coordinates of the center of the moon.
- Its velocity — A 2-dimensional vector
- Its radius — Each moon has a radius so we know when the centers of the two moons are so close to each other that they must have collided.
- Its color — We will use that when do the plots and animations. One moon will be red, the other blue.

There will then be a loop where we will update the positions of the moons and then

recalculate the acceleration and velocities.

How much time will be simulated? 100 days or until the moons collide, whichever comes first.

We will use numpy arrays to represent our vectors.

Create a file called `moons.py`, and type in this code:

```
import numpy as np

# Constants
G = 6.67430e-11              # Gravitational constant (Nm^2/kg^2)
SEC_PER_DAY = 24 * 60 * 60   # How many seconds in a day?
MAX_TIME = 100 * SEC_PER_DAY # 100 days
TIME_STEP = 2 * 60 * 60      # Update every two hours

# Create the inital state of Moon 1
m1 = {
    "mass": 6.0e22,  # kg
    "position": np.array([0.0, 200_000_000]),  # m
    "velocity": np.array([100.0, 25.0]),  # m/s
    "radius": 1_500_000.0,  # m
    "color": "red" # For plotting
}

# Create the inital state of Moon 2
m2 = {
    "mass": 11.0e22,  # kg
    "position": np.array([0.0, -150_000_000]),  # m
    "velocity": np.array([-45.0, 2.0]),  # m/s
    "radius": 2_000_000.0,  # m
    "color": "blue" # For plotting
}

# Lists to hold positions and time
position1_log = []
position2_log = []
time_log = []

# Start at time zero seconds
current_time = 0.0

# Loop until current time exceed Max Time
while current_time <= MAX_TIME:

    # Add time and positions to log
    time_log.append(current_time)
    position1_log.append(m1["position"])
    position2_log.append(m2["position"])
```

```
    # Print the current time and positions
    print(f"Day {current_time/SEC_PER_DAY:.2f}:")
    print(f"\tMoon 1:({m1['position'][0]:,.1f},{m1['position'][1]:,.1f})")
    print(f"\tMoon 2:({m2['position'][0]:,.1f},{m2['position'][1]:,.1f})")

    # Update the positions based on the current velocities
    m1["position"] = m1["position"] + m1["velocity"] * TIME_STEP
    m2["position"] = m2["position"] + m2["velocity"] * TIME_STEP

    # Find the vector from moon1 to moon2
    delta = m2["position"] - m1["position"]

    # What is the distance between the moons?
    distance = np.linalg.norm(delta)

    # Have the moons collided?
    if distance < m1["radius"] + m2["radius"]:
        print(f"*** Collided {current_time:.1f} seconds in!")
        break

    # What is a unit vector that points from moon1 toward moon2?
    direction = delta / distance

    # Calculate the magnitude of the gravitational attraction
    magnitude = G * m1["mass"] * m2["mass"] / (distance**2)

    # Acceleration vector of moon1 (a = f/m)
    acceleration1 = direction * magnitude / m1["mass"]

    # Acceleration vector of moon2
    acceleration2 = (-1 * direction) * magnitude / m2["mass"]

    # Update the velocity vectors
    m1["velocity"] = m1["velocity"] + acceleration1 * TIME_STEP
    m2["velocity"] = m2["velocity"] + acceleration2 * TIME_STEP

    # Update the clock
    current_time += TIME_STEP

print(f"Generated {len(position1_log)} data points.")
```

When your run the simulation, you will see the positions of the moons for 100 days:

```
> python3 moons.py
Day 0.00:
 Moon 1:(0.0,200,000,000.0)
 Moon 2:(0.0,-150,000,000.0)
Day 0.08:
 Moon 1:(720,000.0,200,180,000.0)
 Moon 2:(-324,000.0,-149,985,600.0)
```

```
Day 0.17:
 Moon 1:(1,439,990.7,200,356,896.1)
 Moon 2:(-647,995.0,-149,969,507.0)
...
Day 100.00:
 Moon 1:(119,312,305.5,283,265,313.5)
 Moon 2:(17,393,287.9,-60,319,261.9)
Generated 1201 data points.
```

Look over the code. Make sure you understand what every line does.

9.4 Graph the Paths of the Moons

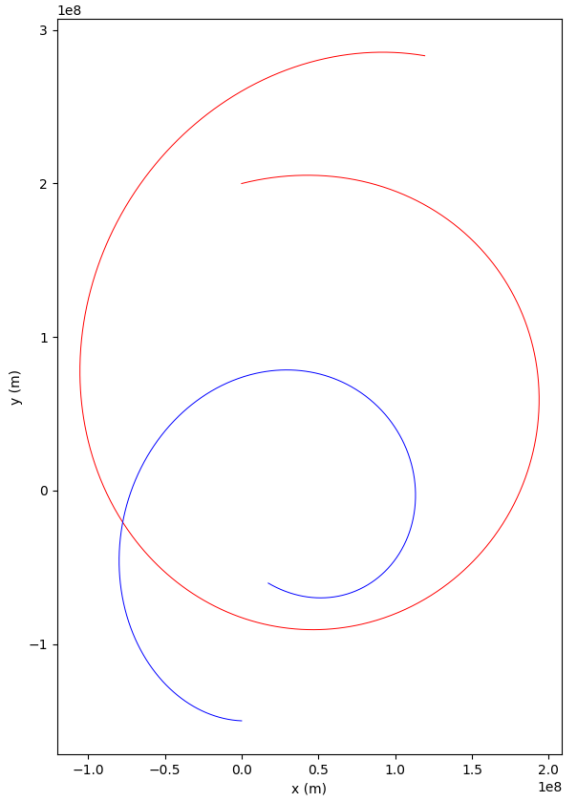
Now, you will use the matplotlib to graph the paths of the moons. Add this line to the beginning of `moons.py`.

```
import matplotlib.pyplot as plt
```

Add this code to the end of your `moons.py`:

```
# Convert lists to np.arrays
positions1 = np.array(position1_log)
positions2 = np.array(position2_log)

# Create a figure with a set of axes
fig, ax = plt.subplots(1, figsize=(7.2, 10))

# Label the axes
ax.set_xlabel("x (m)")
ax.set_ylabel("y (m)")
ax.set_aspect("equal", adjustable='box')

# Draw the path of the two moons
ax.plot(positions1[:, 0], positions1[:, 1], m1["color"], lw=0.7)
ax.plot(positions2[:, 0], positions2[:, 1], m2["color"], lw=0.7)

# Save out the figure
fig.savefig("plotmoons.png")
```

When you run it, your `plotmoons.png` should look like this:



It is nifty to see the paths, but we don't know where each moon was at a particular time. In fact, it is difficult to figure out which end of each curve was the beginning and which was the ending.

What if we added some lines and labels every 300 steps to put a sense of time into the plot? Add one more constant after the import statements:

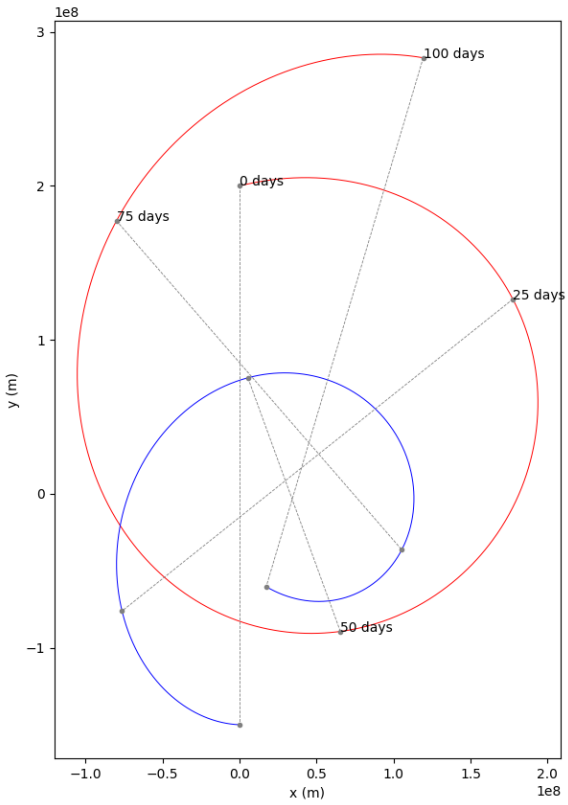
```
PAIR_LINE_STEP = 300  # How time steps between pair lines
```

Immediately before you save the figure to the file, add the following code:

```
# Draw some pair lines that help the
# viewer understand time in the graph
i = 0
while i < len(positions1):

    # Where are the moons at the ith entry?
    a = positions1[i, :]
    b = positions2[i, :]
    ax.plot([a[0], b[0]], [a[1], b[1]], "--", c="gray", lw=0.6, marker=".")

    # What is the time at the ith entry?
    t = time_log[i]
```

```
    # Label the location of moon 1 with the day
    ax.text(a[0], a[1], f"{t/SEC_PER_DAY:.0f} days")
    i += PAIR_LINE_STEP
```

When you run it, your plot should look like this:

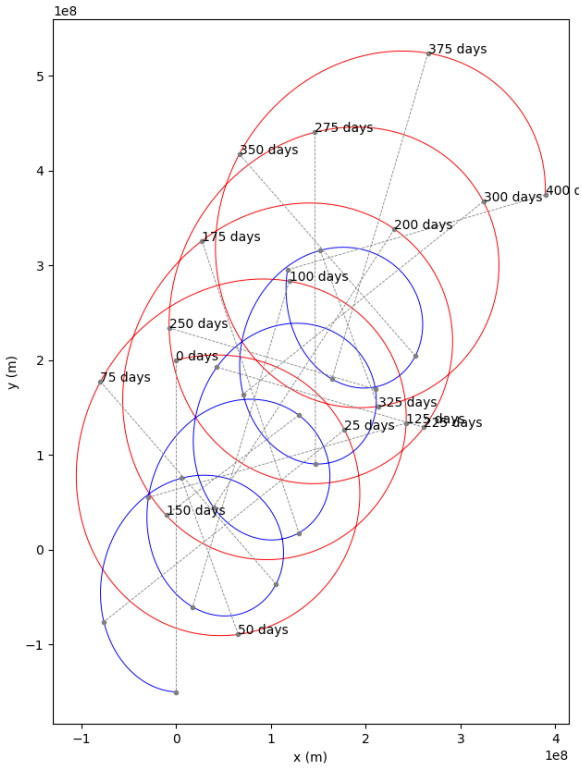


Now you can get a feel for what happened. The moons were attracted to each other by gravity and started to circle each other. The heavier moon accelerates less quickly, so it makes a smaller loop.

Maybe we will get a better feel for what is happening if we look at more time. Let's increase it to 400 days. Change the relevant constant:

```
MAX_TIME = 400 * SEC_PER_DAY # 100 days
```

Now it should look like this:



Now you can see the pattern. They are rotating around each other and the pair is gradually migrating up and to the right.

9.5 Conservation of Momentum

You are observing an extra important idea: the momentum of a system will be conserved. That is, absent forces from outside the system, the velocity of the center of mass will not change.

We can compute the initial center of mass and its velocity. In both cases, we just do a weighted average using the mass of the moon as the weight.

Immediately after you initialize the state of two moons, calculate the initial center of mass and its velocity:

```
# Calculate the initial position and velocity of the center of mass
tm = m1["mass"] + m2["mass"]  # Total mass
cm_position = (m1["mass"] * m1["position"] + m2["mass"] * m2["position"]) / tm
cm_velocity = (m1["mass"] * m1["velocity"] + m2["mass"] * m2["velocity"]) / tm
```

Let's record the center of mass for each time. Before the loop starts, create a list to hold them:

```
cm_log = []
```

Inside the loop (before any calculations), append the current center of mass position to the log:

```
    cm_log.append(cm_position)
```

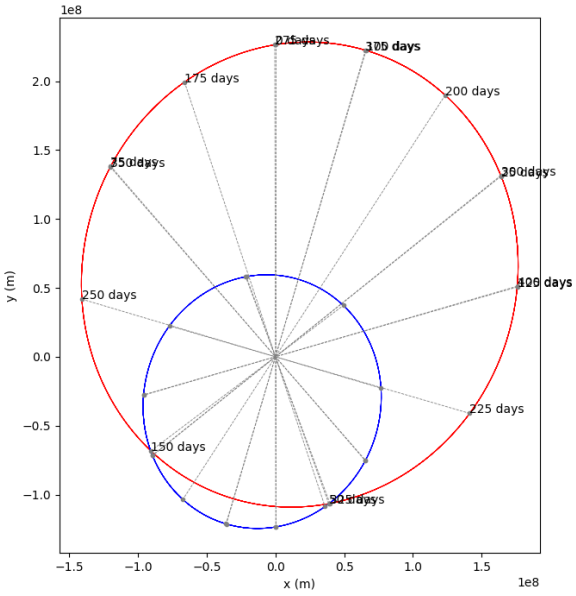
Anywhere later in the loop (after you update the positions of the moon), update `cm_position`:

```
    # Update the center of mass
    cm_position = cm_position + cm_velocity * TIME_STEP
```

Now, let's look at the positions of the moons relative to the center of mass. Before you do any plotting, convert the list to a numpy array and subtract it from the positions:

```
cms = np.array(cm_log)

# Make positions relative to the center of mass
positions1 = positions1 - cms
positions2 = positions2 - cms
```

When you run it, you can really see what is happening:



The moons are tracing elliptical paths. The center of mass is the focus point for both of them.

9.6 Animation

One of the features of matplotlib that not a lot of people understand is how to make animations with it. This seems like a really great opportunity to make an animation showing the position, velocity, acceleration of the moons. We will also show the center of mass.

The trick to animations is that you create a bunch "artist" objects. You create a function that updates the artists. matplotlib will call your functions, tell the artists to draw themselves, and make a movie out of that.

Make a copy of `moons.py` called `animate_moons.py`.

Edit it to look like this:

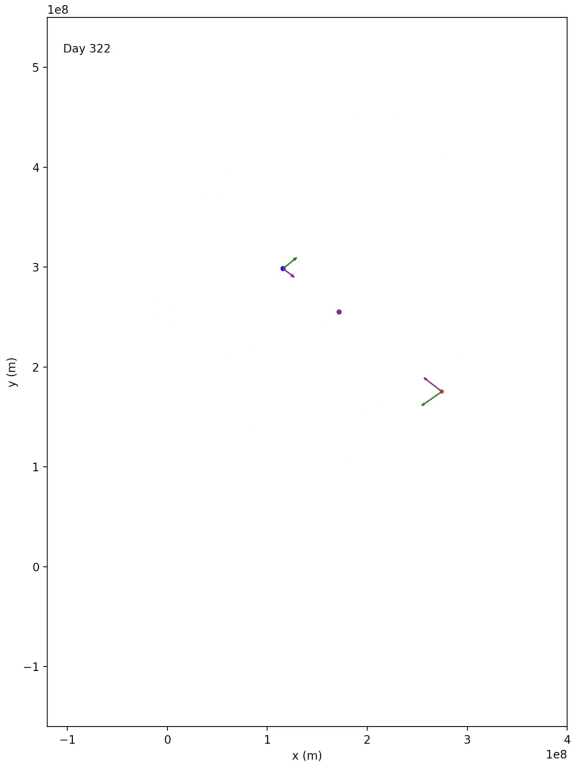
```
import numpy as np
import matplotlib.pyplot as plt

# Import animation support and artists
from matplotlib.animation import FuncAnimation
from matplotlib.patches import Circle, FancyArrow
from matplotlib.text import Text

# Constants
G = 6.67430e-11  # Gravitational constant (Nm^2/kg^2)
SEC_PER_DAY = 24 * 60 * 60  # How many seconds in a day?
MAX_TIME = 400 * SEC_PER_DAY  # 100 days
TIME_STEP = 12 * 60 * 60  # Update every 12 hours
FRAMECOUNT = MAX_TIME / TIME_STEP  # How many frames in animation
ANI_INTERVAL = 1000 / 50  # ms for each frame in animation

# The velocity and acceleration vectors are invisible
# unless we scale them up.  A lot.
VSCALE = 140000.0
ASCALE = VSCALE * 800000.0

# Create the inital state of Moon 1
m1 = {
    "mass": 6.0e22,  # kg
    "position": np.array([0.0, 200_000_000]),  # m
    "velocity": np.array([100.0, 25.0]),  # m/s
    "radius": 1_500_000.0,  # m
    "color": "red",  # For plotting
}

# Create the inital state of Moon 2
m2 = {
    "mass": 11.0e22,  # kg
    "position": np.array([0.0, -150_000_000]),  # m
    "velocity": np.array([-45.0, 2.0]),  # m/s
```

```
    "radius": 2_000_000.0,  # m
    "color": "blue",  # For plotting
}

# Calculate the initial position and velocity of the center of mass
tm = m1["mass"] + m2["mass"]  # Total mass
cm_position = (m1["mass"] * m1["position"] + m2["mass"] * m2["position"]) / tm
cm_velocity = (m1["mass"] * m1["velocity"] + m2["mass"] * m2["velocity"]) / tm

# Start at time zero seconds
current_time = 0.0

# Create the figure and axis
fig, ax = plt.subplots(1, figsize=(7.2, 10))

# Set up the axes
ax.set_xlabel("x (m)")
ax.set_xlim((-1.2e8, 4e8))
ax.set_ylabel("y (m)")
ax.set_ylim((-1.6e8, 5.5e8))
ax.set_aspect("equal", adjustable="box")
fig.tight_layout()

# Create artists that will be edited in animation
time_text = ax.add_artist(Text(0.03, 0.95, "", transform=ax.transAxes))
circle1 = ax.add_artist(Circle((0, 0), radius=m1["radius"], color=m1["color"]))
circle2 = ax.add_artist(Circle((0, 0), radius=m2["radius"], color=m2["color"]))
circle_cm = ax.add_artist(Circle((0, 0), radius=m2["radius"], color="purple"))
varrow1 = ax.add_artist(FancyArrow(0, 0, 0, 0, color="green", head_width=m1["radius"]))
varrow2 = ax.add_artist(FancyArrow(0, 0, 0, 0, color="green", head_width=m2["radius"]))
acc_arrow1 = ax.add_artist(
    FancyArrow(0, 0, 0, 0, color="purple", head_width=m1["radius"])
)
acc_arrow2 = ax.add_artist(
    FancyArrow(0, 0, 0, 0, color="purple", head_width=m2["radius"])
)


# This function will get called for every frame
def animate(frame):

    # Global variables needed in scope from the model
    global cm_position, cm_velocity, current_time, m1, m2

    # Global variables needed in scope from the artists
    global time_text, varrow1, varrow2, acc_arrow1, acc_arrow2, circle1, circle2, circle_cm

    print(f"Updating artists for day {current_time/SEC_PER_DAY:.1f}.")

    # Update the positions based on the current velocities
    m1["position"] = m1["position"] + m1["velocity"] * TIME_STEP
    m2["position"] = m2["position"] + m2["velocity"] * TIME_STEP
```

```
# Update day label
time_text.set_text(f"Day {current_time/SEC_PER_DAY:.0f}")

# Update positions of circles
circle1.set_center(m1["position"])
circle2.set_center(m2["position"])

# Update velocity arrows
varrow1.set_data(
    x=m1["position"][0],
    y=m1["position"][1],
    dx=VSCALE * m1["velocity"][0],
    dy=VSCALE * m1["velocity"][1],
)
varrow2.set_data(
    x=m2["position"][0],
    y=m2["position"][1],
    dx=VSCALE * m2["velocity"][0],
    dy=VSCALE * m2["velocity"][1],
)

# Update the center of mass
cm_position = cm_position + cm_velocity * TIME_STEP
circle_cm.set_center(cm_position)

# Find the vector from moon1 to moon2
delta = m2["position"] - m1["position"]

# What is the distance between the moons?
distance = np.linalg.norm(delta)

# Have the moons collided?
if distance < m1["radius"] + m2["radius"]:
    print(f"*** Collided {current_time:.1f} seconds in!")

# What is a unit vector that points from moon1 toward moon2?
direction = delta / distance

# Calculate the magnitude of the gravitational attraction
magnitude = G * m1["mass"] * m2["mass"] / (distance**2)

# Acceleration vector of moons (a = f/m)
acceleration1 = direction * magnitude / m1["mass"]
acceleration2 = (-1 * direction) * magnitude / m2["mass"]

# Update the acceleration arrows
acc_arrow1.set_data(
    x=m1["position"][0],
    y=m1["position"][1],
    dx=ASCALE * acceleration1[0],
    dy=ASCALE * acceleration1[1],
```

```
    )
    acc_arrow2.set_data(
        x=m2["position"][0],
        y=m2["position"][1],
        dx=ASCALE * acceleration2[0],
        dy=ASCALE * acceleration2[1],
    )

    # Update the velocity vectors
    m1["velocity"] = m1["velocity"] + acceleration1 * TIME_STEP
    m2["velocity"] = m2["velocity"] + acceleration2 * TIME_STEP

    # Update the clock
    current_time += TIME_STEP

    # Return the artists that need to be redrawn
    return (
        time_text,
        varrow1,
        varrow2,
        acc_arrow1,
        acc_arrow2,
        circle1,
        circle2,
        circle_cm,
    )


# Make the rendering happen
animation = FuncAnimation(
    fig,
    animate,
    np.arange(FRAMECOUNT),
    interval=ANI_INTERVAL
)

# Save the rendering to a video file
animation.save("moonmovie.mp4")
```

When you run this, it will take longer than the previous versions. You should have a video file that shows a simulation of the moons tracing their elliptical paths around their center of mass:



9.7 Challenge: The Three-Body Problem

It is time to stretch a little as a physicist and programmer: You are going to make a new version of `moons.py` that handles three moons instead of just two.

This is known as "The Three-Body Problem", and people have tried for centuries to come up with a way to figure out (from the initial conditions) where the three moons would be at time t without doing a simulation. And no one has.

For a lot of problems, the outcome is not very sensitive to the initial conditions. For example, consider the flight of a cannonball: If it leaves the muzzle of the cannon a little faster, it will go a little farther.

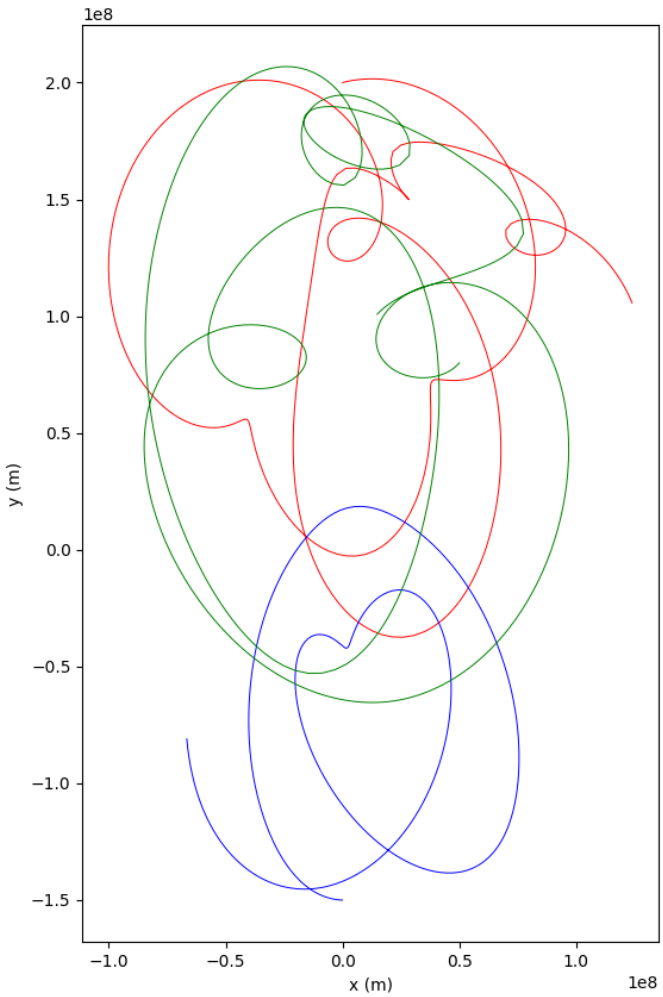
For the three-body problem, the outcome can be radically different even if the initial conditions are very similar.

(There is a whole field of mathematics studying systems that are very sensitive to initial conditions. It is known as *dynamical systems* or *chaos theory*.

Copy `moons.py` to `3moons.py`. Here is a reasonable initial state for your third moon:

```
m3 = {
    "mass": 4.0e22,  # kg
    "position": np.array([50_000_000, 80_000_000]),  # m
    "velocity": np.array([-30.0, -35.0]),  # m/s
    "radius": 1_700_000.0,  # m
    "color": "green"
}
```

If we run that simulation for 100 days, we get a plot like this:



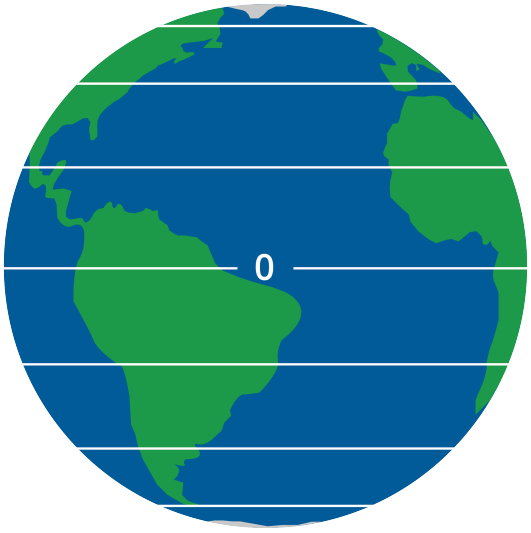
Visibly, you can see this is very different from the two-body problem that just traced ellipses around the center of mass.

Chapter 10

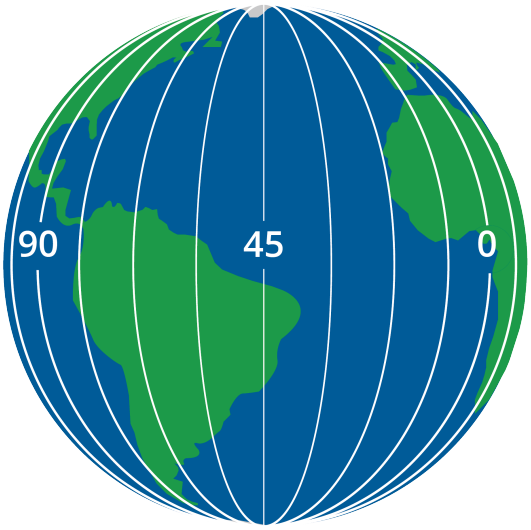
Longitude and Latitude

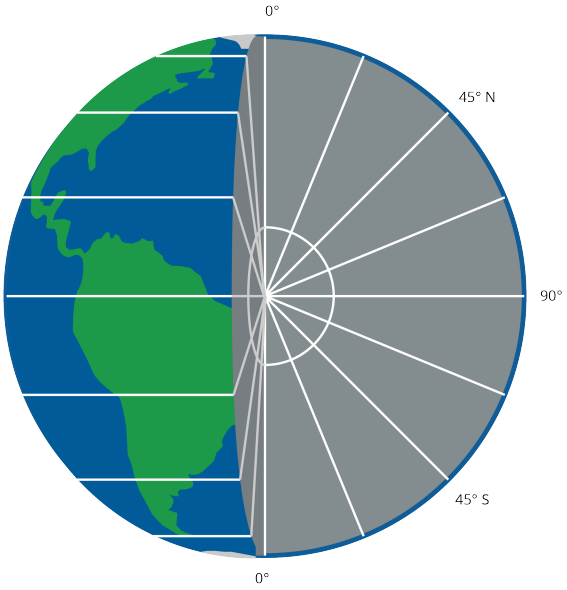
The Earth can be represented as a sphere, and the position of a point on its surface can be described using two coordinates: latitude and longitude.

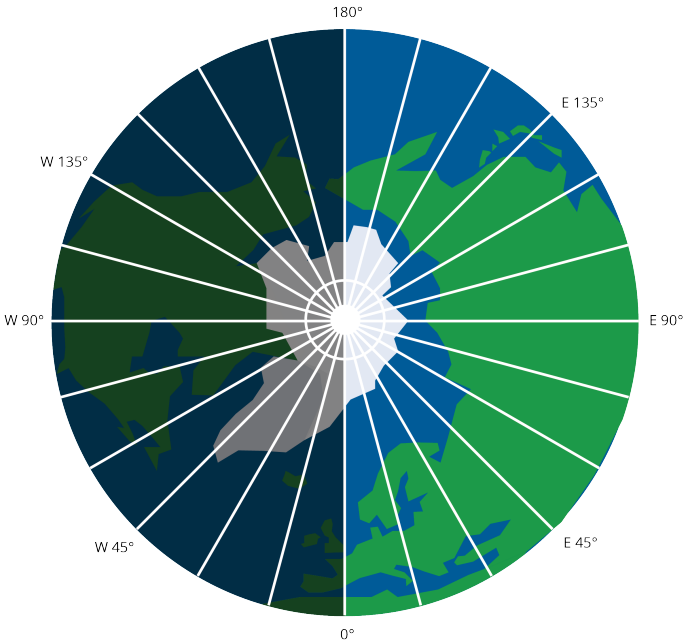
Latitude is a measure of a point's distance north or south of the equator, expressed in degrees. It ranges from $-90°$ at the South Pole to $+90°$ at the North Pole, with $0°$ representing the Equator.



Longitude, on the other hand, measures a point's distance east or west of the Prime Meridian (which passes through Greenwich, England). It ranges from $-180°$ to $+180°$, with the Prime Meridian represented as $0°$.







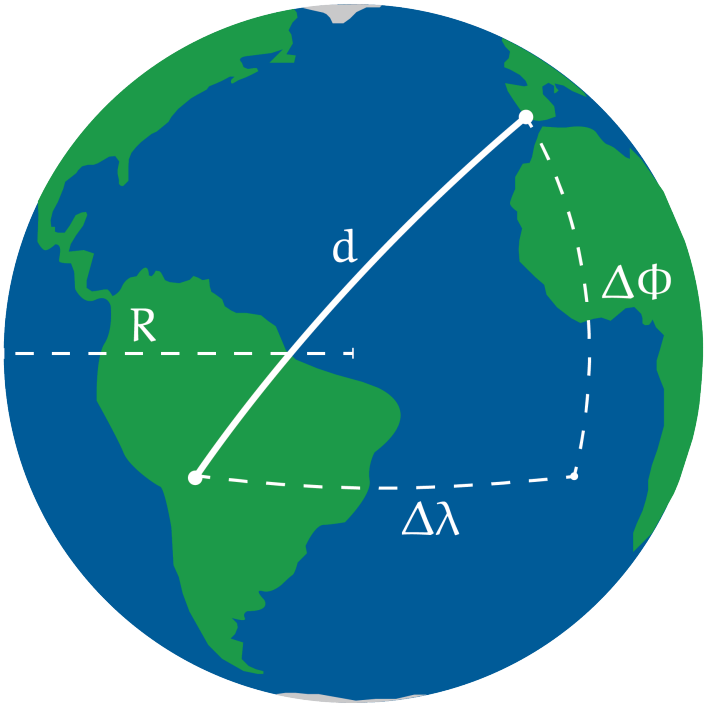
10.1 Nautical Mile

A nautical mile is a unit of measurement used primarily in aviation and maritime contexts. It is based on the circumference of the Earth, and is defined as one minute (1/60°) of

latitude. This makes it directly related to the Earth's geometry, unlike a kilometer or a mile, which are arbitrary in nature. The exact value of a nautical mile can vary slightly depending on which type of latitude you use (e.g., geodetic, geocentric, etc.), but for practical purposes, it iss often approximated as 1.852 kilometers or 1.15078 statute miles.

10.2 Haversine Formula

The haversine formula is an important equation in navigation for giving great-circle distances between two points on a sphere from their longitudes and latitudes. It is especially useful when it comes to calculating distances between points on the surface of the Earth, which we represent as a sphere for simplicity.



In its basic form, the haversine formula is as follows:

$$a = \sin^2\left(\frac{\Delta\phi}{2}\right) + \cos(\phi_1)\cos(\phi_2)\sin^2\left(\frac{\Delta\lambda}{2}\right)$$

$$c = 2 \cdot \operatorname{atan2}\left(\sqrt{a}, \sqrt{1-a}\right)$$

$$d = R \cdot c$$

Here, ϕ represents the latitudes of the two points (in radians), $\Delta\phi$ and $\Delta\lambda$ represent the differences in latitude and longitude (also in radians), and R is the radius of the Earth. The result, d, is the distance between the two points along the surface of the sphere.

Chapter 11

Tides and Eclipses

Your life on earth involves many orbital paths:

- The earth is spinning. If you are standing at the equator, you are traveling at 1,674 km per hour around the center of the planet. We are all spinning east, which is why the sun comes up in the east and sets in the west.
- The earth is orbiting the sun. It takes 365.242 days for the earth to go once around the sun. This is why different constellations appear at different times during the year — we only see the stars at night and the direction of night shifts as the earth moves around the sun.
- The moon is orbiting the earth. The moon travels once around the earth once every 27.3 days.

You can see the effects of these orbits on our planet. Let's go over a few.

11.1 Leap Years

Note that it takes 365.242 days for the earth to go around the sun. If we declared "The calendar will *always* be 365 days per year!" then the seasons would gradually shift by 0.242 days every year. After a century, they would have migrated 24 days.

So, we made a rule: "Every fourth year, we will add an extra day to the calendar!" The years 2021, 2022,and 2023 get no February 29th, but 2024 does.

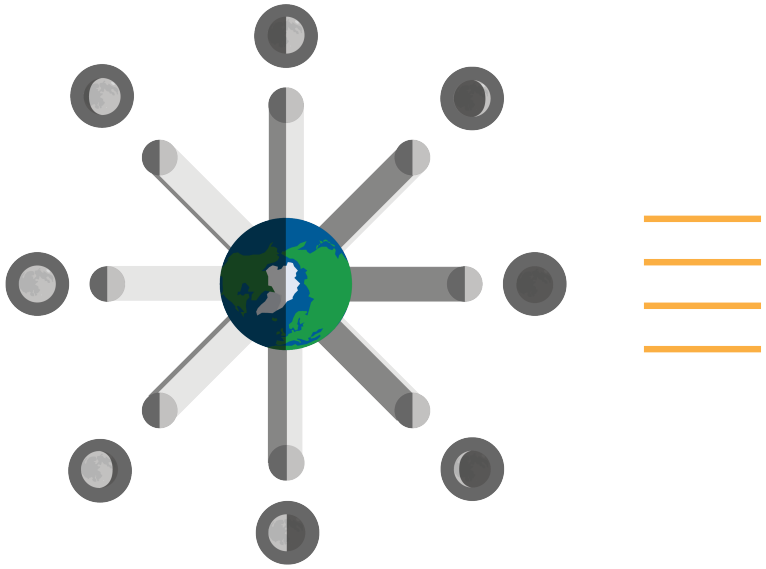
That got us a calendar with an average 365.25 days per year, so the seasons would not have migrated as quickly, but they still would have migrated about three days every four hundred years.

So, we made another rule: "There will be no February 29th in the three century years (multiples of 100) that are not multiples of 400." So the year 1900 had no Feb 29, but the year 2000 had one. Now, the average number of days per year is 365.2425.

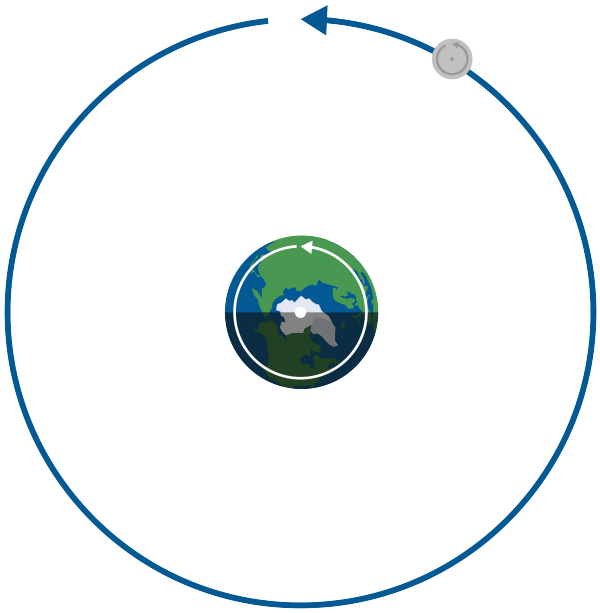
11.2 Phases of the Moon

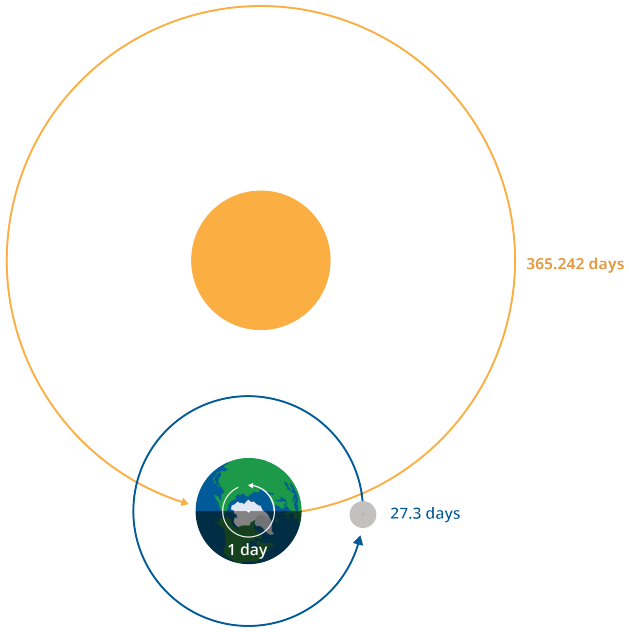
The earth, the moon, and the sun form a triangle. If you were standing on the moon, you could measure the angle between the light coming from the sun and the the light going to the earth. That angle would fluctuate between 0 degrees and 180 degrees.

- When the angle was close to 0, the people on earth would see a full moon.
- When the angle was close to 90 degrees, the people on earth would see a half moon.
- When the angle was nearing 180 degrees, the people on earth would see a slim crescent.
- When the angle was very close to 180 degrees, the moon would be dark. This is called a "new moon."



Even though it takes 27.3 days for the moon to travel around the earth once, it takes 29.5 days to get from one full moon to the next. Why? In the 27.3 days that it took the moon to travel around the earth, the earth has moved about 17 degrees around the sun. To get back into the same triangle configuration takes another 2.2 days.



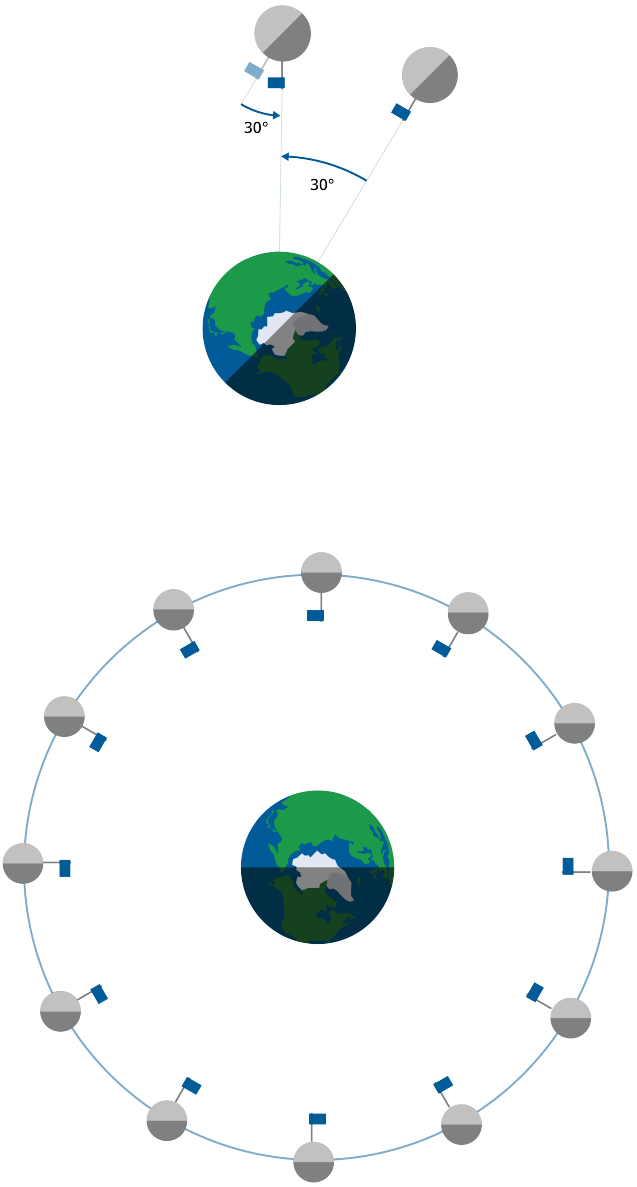


To explain why we often see a curve in the shadow of the moon, we can look at a ball that has one side painted yellow and the other red.

As we rotate the ball, we can see that the straight color boundary between each hemisphere begins to look curved. The curve we see in the moon is due to this same basic principle of how the shading of spheres works.

FIXME: Add text about scale In all of these graphics, we have been using incorrect scale. Here is the true scale of the distance of the earth and the moon with accurate radii:

FIXME: Add text about tidal lock

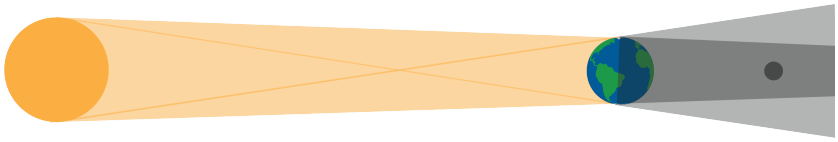


11.3 Eclipses

While the earth orbits the sun and the moon orbits the earth, the two orbits are *not in the same plane*. We call the plane that the earth orbits the sun in the *ecliptic plane*. The plane of the moon's orbit is about 5 degrees tilted from the ecliptic plane.

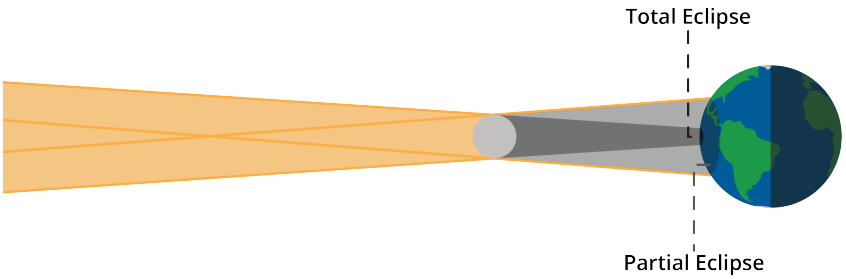
Note that the moon passes through the ecliptic plane only twice every 27.3 days. Imagine

that the instant it passed through the ecliptic plane was also the precise instant of a full moon. The sun, the earth, and the moon would be in a straight line! The earth would cast a shadow upon the moon — it would go from a bright full moon to a dark moon until the moon moved back out of the shadow of the earth. This is known as a *lunar eclipse*.



The diameter of the moon is a little more than a quarter the diameter of the earth, so they don't have to be in perfect alignment for the moon to be darkened. Lunar eclipses actually happen once or twice per year.

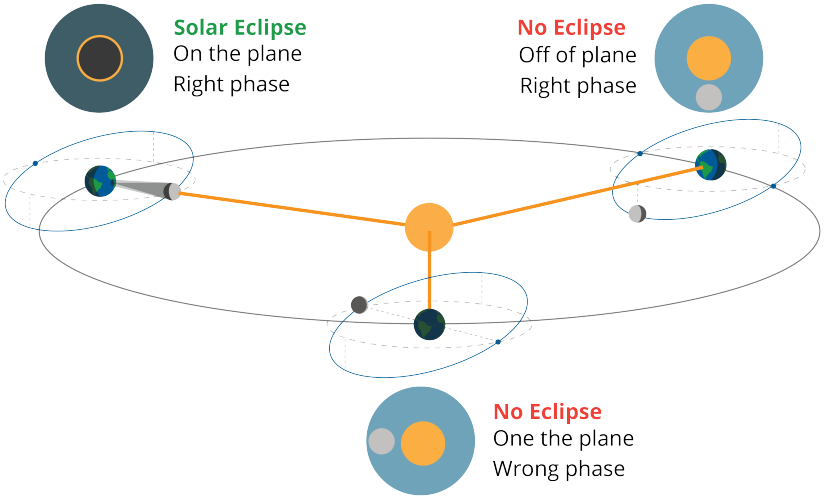
Now, imagine that the instant the moon passed through the ecliptic plane was also the precise instant of a new moon. The sun, the moon, and the earth would be in a straight line! The moon would cast a shadow upon some part of the earth. To a person in that shadow, the sun disappear behind the moon. This is known as a *solar eclipse*.



The sun is pretty big, so if the moon blots out just part of it, we call it a *partial solar eclipse*. There are a few partial solar eclipses every year. Note that because the moon's shadow

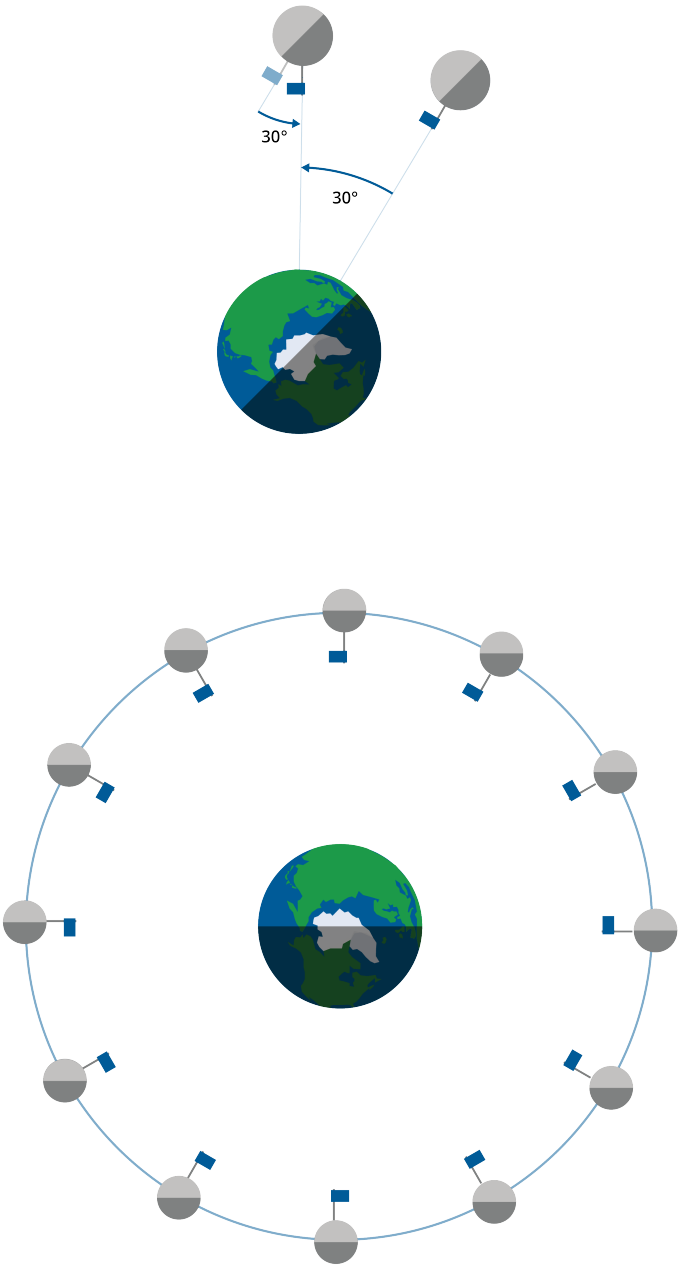
is too small to shade the whole earth, only certain parts of the world will experience any solar eclipses.

Every 18 months or so, there is a total eclipse of the sun. Once again, only certain parts of the world experience it. You can expect to experience a total eclipse of the sun at your home about once every 375 years.



11.4 The Far Side of the Moon

Like the earth, the moon spins on its axis. Due to earth's gravity, the rotation of the moon slowed down until its spin matched the rate it orbits earth. That is: we are always looking at the same side of the moon. Until we orbited the moon, we had no idea what the far side looked like.



Some people call it "The Dark Side of the Moon," but it gets just as much sunshine as the side that faces earth. The name comes from the fact that we lose communication with spacecraft (like the Apollo missions to the moon) when they are on the far side of the moon. When we lose communications with a craft, we often say "It went dark."

11.5 Tides

When we say "The moon orbits the earth," that is a bit of an oversimplification. The force of gravity that pulls the moon toward the earth, also pulls the earth toward the moon. The earth is about 81 times heavier than the moon, so the moon moves more, but the moon definitely moves the earth.

The center of the moon and the center of the rotate around each other. The point they rotate around is inside the the earth, but it is closer to the surface of the earth than it is to the center of the earth.

Orbits happen, remember, when the centripetal force is equal to gravitational force. So the centripetal force created by the earth being swung by the moon is equal to the gravitational force that the moon exerts on all the mass on the moon.

However.

The parts of the earth that are closer to the moon experience less centripetal force (away from the moon) and more gravitational force (toward the moon).

The parts of the earth that are farther from the moon experience more centripetal force (away from the moon) and less gravitational force (away from the moon).

The effects are not big. For example, you won't notice that you can jump higher when the moon is overhead. You will lose only about 1/200,000 of your weight.

But the ocean is huge.: 1/200,000 of its weight is a lot of force.

The water in the oceans bulges a little both toward the moon and away from it.

The earth is still rotating. If you are at the beach as your longitude slides into one of these bulges, you say "Hey, the tide is rising!" The peak of these bulges is known as "high tide". Because there is a bulge on each side of the planet, high tide comes twice a day.

This is a lunar tide – because it is caused by the moon. There is a similar effect from the sun, but the sun is very, very far away: solar tidal forces are about half as powerful lunar tidal forces. When the sun and the moon work together, the tides are stronger. This is called a *spring tide*. Spring tides don't happen in the spring time; they happen close to full moons and new moons.

When the moon and the sun are working against each other, the tides are weaker. This is called a *neap tide*. Neap tides happen when you see a half moon in the sky.

11.5.1 Computing the Forces

We are enumerating several forces that shape the water on the planet. All these forces are pulling on your body too. In these exercises, you are going to calculate how each force would effect a 1 kg mass on the surface of the earth.

Here are some numbers you will need:

- The mass of the earth: 5.97219×10^{24} kg
- The mass of the sun: 1.9891×10^{30} kg
- The mass of the moon: 7.347673×10^{22} kg
- Radius of the earth at the equator: $6,371$ km
- Average distance from the center of the earth to the center of the sun: 149.6×10^{6} km
- Average distance from the center of the moon to the center of the earth: $384,467$ km.

Exercise 5 Life Among the Orbits 1: Earth Gravity

Working Space

If the earth were still and alone in the universe, there would still be the force of gravity. We have said that that a kilogram on the surface of the earth is pulled toward the center of the earth with a force of 9.8 N.

Confirm that the gravity of the earth pulls a 1kg mass on the surface of the planet with a force of about 9.8 N.

You will need the formula for gravitation:

$$F_g = \frac{g m_1 m_2}{r^2}$$

If we measure distance in km and mass in kg, the gravitation constant g is 6.67430×10^{-17}.

Answer on Page 86

Exercise 6 Life Among the Orbits 2: Earth Centripetal Force

Working Space

What if we add the spinning of the earth? The spinning would try to throw the kg into space. The formula for centripetal force is

$$F_c = \frac{mv^2}{r}$$

Calculate the centripetal force on a 1 kg mass on the surface of the earth. It doesn't fly off into space, so the force due to gravity must be bigger. How many times bigger?

Assume that the mass is on the equator, thus rotating around the earth at 465 m/s.

Does the centripetal force increase, decrease, or stay the same as you get closer to the north pole?

Answer on Page 86

Exercise 7 Life Among the Orbits 3: The Moon's Gravity

Working Space

Now we add the moon's gravitational force to our model.

When the moon is directly overhead, how strongly will it pull at the 1 kg mass on the equator?

When the moon is directly underfoot, how strongly will it pull at the 1 kg mass on the equator?

Is that a big difference?

Answer on Page 86

Exercise 8 Life Among the Orbits 4: The Swing of the Moon

Working Space

Now we add the moon's motion. The moon and the earth swing each other around. This creates a centripetal force. They both travel in nearly a circle centered at their center of mass.

How far is the center of mass of the moon and the earth from the center of the earth? (You can imagine a see-saw with the center of the earth on one end and the center of the moon on the other. Where would the balance point be?)

What point on the surface of the earth is closest to the center of mass? How far is it?

What point on the surface of the earth is farthest from the center of mass? How far is it?

Answer on Page 87

Exercise 9 Life Among the Orbits 5: Lunar Centripetal Force

Working Space

The moon swings us around that center of mass once every 27.3 days. (Forget about the spinning of the earth for this part.) What is the largest and smallest centripetal forces on the surface of the earth created by this swinging

What is the largest centripetal force on a 1 kg mass with the moon directly underfoot? (You need an answer from the previous question: There is a point on the surface of the earth that is 11,044,000 m from the center of gravity.)

What is the resulting centripetal force on a 1 kg mass with the moon directly overhead? (You will need the other answer from the previous exercise: That point is 1,698,000 m from the center of mass of the moon and the earth.)

For this problem is probably easier to use this formula for centripetal force:

$$F_c = mr\omega^2$$

Where m is mass in kg, r is radius in m, and ω is the angular velocity in radians per second.

Answer on Page 87

Exercise 10 Life Among the Orbits 6: Net Force

Working Space

Now add together the two forces at both the nearest point to the moon and the farthest.

Answer on Page 88

11.5.2 Solar Tidal Forces

The sun has a much larger gravitational effect on the earth than the moon does:

- When the sun is overhead, it will pull on a 1 kg mass with a force of about 0.00593 N.
- When the moon is overhead, it will pull on a 1 kg mass with a force of about 0.0000343 N.

Why are lunar tides about twice as powerful solar tides?

Tides occur because the pull of gravity and the pull of the centripetal force are out of balance somewhere on the planet. The sun is so far away that the effects of gravitational and centripetal forces are very close to equal everywhere on earth.

Appendix A

Answers to Exercises

Answer to Exercise 1 (on page 5)

A is 440 Hz. Each half-step is a multiplication by $\sqrt[12]{2} = 1.059463094359295$ So the frequency of E is $(440)(2^{7/12}) = 659.255113825739859$

Answer to Exercise 2 (on page 23)

The force of gravity is $9.8 \times 45 = 441$ newtons.

At any speed s, the force of wind resistance is $0.05 \times s^2 = 0.05s^2$ newtons.

At terminal velocity, $0.05s^2 = 441$.

Solving for s, we get $s = \sqrt{\frac{441}{0.05}}$

Thus, terminal velocity should be about 94 m/s.

Answer to Exercise 3 (on page 33)

$$\frac{120 \text{ km}}{1\text{hour}} = \frac{1000 \text{ m}}{1 \text{ km}} \frac{120 \text{ km}}{1\text{hour}} \frac{1 \text{ hour}}{3600 \text{ seconds}} = 33.3 \text{ m/s}$$

$$F = \frac{mv^2}{r} = \frac{0.4(33.3)^2}{200} = 2.2 \text{ newtons}$$

Answer to Exercise 3 (on page 37)

$$v = \sqrt{3.721(3.4 \times 10^6)} = 3,557 \text{ m/s}$$

The circular orbit is $2\pi(3.4 \times 10^6) = 21.4 \times 10^6$ meters in circumference.

The period of the orbit is $(21.4 \times 10^6)/3,557 \approx 6,000$ seconds.

Answer to Exercise 5 (on page 79)

The earth and 1 kg on the surface would attract each other with a force of:

$$F_g = \frac{\left(6.67430 \times 10^{-17}\right)\left(5.97219 \times 10^{24}\right)(1)}{6,371^2} = \frac{3.98583 \times 10^8}{4.0590 \times 10^7} = 9.7987 \text{ N}$$

Thus, if the earth were still and alone in the universe, the oceans would form a perfect sphere.

Answer to Exercise 6 (on page 80)

$$F_c = \frac{(1)(465)^2}{6,371,000} = 0.03373 \text{ N}$$

So the spinning of the earth is trying to throw you into space, but the force of gravity is about 289 times more powerful.

This centripetal force decreases as you move from the equator to the north pole. In fact, at the north pole, there is no centripetal force. Thus, the spinning of the earth makes the oceans an oblate ellipsoid instead of a perfect sphere: the diameter going from pole-to-pole is shorter than a diameter measured at the equator.

You should feel a teensy-tiny bit lighter on your feet at the equator than you do at the north pole: 0.34% lighter.

Answer to Exercise 7 (on page 81)

Overhead, the moon is $384,467 - 6,371 = 378,096$ km from your 1 kg mass.

$$F_g = \frac{gm_1m_2}{r^2} = \frac{\left(6.67430 \times 10^{-17}\right)\left(7.347673 \times 10^{22}\right)(1)}{378,096^2} = \frac{4.9040574 \times 10^6}{1.42956585216 \times 10^{11}} = 3.43058 \times 10^{-5} \text{ N}$$

This is a very small force: The force due to earth's gravity is nearly three hundred thousand times stronger.

Underfoot, the moon is $384,467 + 6,371 = 390,838$

$$F_g = \frac{gm_1m_2}{r^2} = \frac{\left(6.67430 \times 10^{-17}\right)\left(7.347673 \times 10^{22}\right)(1)}{390,838^2} = \frac{4.9040574 \times 10^6}{1.52754 \times 10^{11}} = 3.2103{\times}10^{-5}\ \text{N}$$

The force due to the moon's gravity is about 6% stronger when the the moon is overhead than when it is underfoot.

Answer to Exercise 8 (on page 82)

If we let r be the distance (in km) from the center of the earth to the center of mass, the distance from the center of the mass to the center of the moon is $384,467 - r$.

To find the balance point, multiply each mass by how far it is from the center of mass:

$$\left(5.97219 \times 10^{24}\right) r = \left(7.347673 \times 10^{22}\right)(384,467 - r)$$

Solving for r:

$$r = \frac{4,730.15}{1 + 0.0123} = 4,673\ \text{km}$$

The point on the earth closest to this? It is where the moon is directly overhead. The it is $6,371 - 4,673 = 1,698$ km from the center of mass.

The point on the earth farthest from this? It is where the moon is directly underfoot. The it is $6,371 + 4,673 = 11,044$ km from the center of mass.

Answer to Exercise 9 (on page 83)

First, lets figure out ω. It travels through 2π radians in 27.3 days. 27.3 days = 2,358,720 seconds. $\omega = \frac{2\pi}{2,358,720} = 2.663811435 \times 10^{-6}$

$$F_c = (1)(11,044,000)(2.663811435 \times 10^{-6})^2 = 7.8365 \times 10^{-5}$$

Now the weakest:

$$F_c = (1)(1,698,000)(2.663811435 \times 10^{-6})^2 = 1.20512 \times 10^{-5}$$

Answer to Exercise 10 (on page 84)

Closest to the moon, the gravitational force of the moon and the centripetal forces are in the same direction: toward the moon.

$$F_{total} = 1.20488 \times 10^{-5} + 3.43045 \times 10^{-5} = 4.6356 \times 10^{-5}\ \text{N}$$

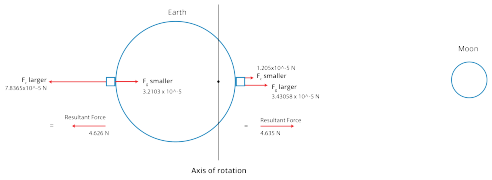
Farthest from the moon, the gravitational force of the moon and the centripetal forces are in opposite directions:

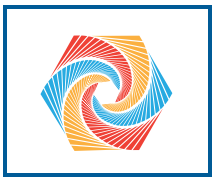
$$F_{total} = 7.8367 \times 10^{-5} - 3.2104 \times 10^{-5} = 4.62604^{-5}\text{N}$$

This is great conclusion: The two forces are basically equal: one pulls the water closest to the moon toward the moon, the other pulls water farthest from the moon away from the moon.

Both forces are pretty small: The force due to earth's gravity is about $211,000$ times more than either.

And that is why there are two basically equally large high tides every day.





Index