# Week of April 11-April 18

1. Finding the duration of the “interesting zone” correctly
   1. After correcting distance calculation
      1. When I did my distance calculation at first, I just subtracted the coordinates to find the distance, which was completely wrong. I then fixed this to instead use the formula for calculating distance between 2 points.
      2. Running the program again with a timestep of 1 day and radius of 5 AU, I found the duration to be 1 day, which seems a lot more correct than what I had before.
      3. It’s possible that the duration is actually only a few hours, and the timestep is too large, so I tried with a smaller timestep. However, now the sail doesn’t even reach within 5 AU of the target, which is a problem.
      4. Turns out that changing the timestep changes the path of the sail slightly, which at larger distances becomes a major change, and so the sail misses the star entirely. To account for this, I now have to run the program twice: first to find a location in the sail’s path which is roughly the distance of the target, change the target’s location to be 1 AU away from that location, and then run it again to see how long the sail will be in the interesting zone radius.
   2. Changing location of the target for different timesteps
      1. After the first run, the program will print out the point at which the sail is roughly the target’s distance from the Sun. I just copy and paste this into the coordinates of the target, adding 1 AU to the x-coordinate. When I run it again, the program then correctly calculates the duration that the sail is within the interesting zone. Interestingly, the x-coordinate that I found changes a lot with timestep.
      2. I ran it with a few different timesteps:

| Timestep (hours) | Radius | X-Position | Duration (hours) |
| --- | --- | --- | --- |
| 24 | 5 AU | 4.13391e+12 | 24 |
| 12 | 5 AU | 1.83237e+12 | 12 |
| 6 | 5 AU | 6.85391e+11 | 6 |
| 3 | 5 AU | 1.25400e+11 | 9 |
| 1 | 5 AU | -1.64265e+11 |  |

1. Trying to calculate relativistic velocity correctly
   1. Correction
      1. I had to make a small correction, I had not included the reflectivity term in the force calculation earlier, so it has changed from:

F = P / c

to:

F = 2P / c

* + 1. Because the force is now doubled, the distance that can be calculated without crashing is also lower.
  1. Trying with smaller timesteps
     1. The program now crashes at 1 AU, so I tried to calculate the velocity for that distance using smaller timesteps.
     2. Unfortunately, the program still crashes no matter how low the timestep is set to, the only difference each time is the distance after which it crashes, as shown below:

| Timestep | Distance |
| --- | --- |
| 1 | 1.3418 \* 1011 m |
| 0.1 | 1.3486 \* 1011 m |
| 0.01 | 1.3498 \* 1011 m |
| 0.001 | 1.3499 \* 1011 m |
| 0.0001 | 1.3499 \* 1011 m |
| 0.00001 | 1.3499 \* 1011 m |

* + 1. Clearly, the program will not work for distances >= 1.35 \* 1011 m.
  1. Using new equations
     1. Since our previous equations are incorrect, Dr Richmond calculated a different equation to use for calculating relativistic velocity.
     2. Using the acceleration a0 from the laser, the velocity at a time t would be:

v = (a0\*t/c) / sqrt( 1 + (a0\*t/c)2 )

* + 1. This means we don’t need to calculate velocity iteratively, but we do need to do that for distance. Since we are calculating velocity based on distance, we still need an iterative algorithm, as below:

t = 0

x = 0

a0 = F / m

while x < x\_final:

v = (a0\*t/c) / sqrt( 1 + (a0\*t/c)2 )

x = x + v\*t

t = t + dt

* + 1. I wrote the algorithm, and tried to find the velocity for a distance of 3.844 \* 108 m (distance between Earth and Moon), and the program failed. It did not crash, but it ran for a very, very long time, and calculated the velocity to be 0.99 m/s. This is very, very wrong.
    2. Upon looking at the equation for velocity, this kind of makes sense, if we let (a0\*t/c) = p, then v = p / √(1+p2) , the denominator is always slightly larger than the numerator, so v will always be less than 1.

1. Estimating size of the laser for extremely large velocities
   1. So, apparently if we accelerate the sail for ~0.89 AU, it can reach the speed of light. Lets see how big the laser needs to be for that to happen. We know that:

Ro = DsDt/2.44λ

⇒ DtDs = 2.44λRo

where Dt is the diameter of the laser array, Ds is the diameter of the sail, Ro is the distance the sail is being accelerated and λ is the wavelength emitted by the laser. From the parameters for starshot we found before, we know that λ = 1.06 \* 10-6 m, so we have:

DsDt = (2.586 \* 10-6)Ro

From above, Ro = 1.34 \* 1011, so

DsDt = 346,577.6 = 3.465 \* 105 m2

For the starshot sail we are currently using, Ds = 5.656 m, so we have the diameter of the laser:

Dt = 61,276.09 m = 6.127 \* 104 m ~ 61.276 km

So, to go the speed of light, we need a laser array of diameter 61.276 km. Assuming the array is circular, it would have a surface area of 2.948 \* 109 m2 = 2,948 km2.

So, this laser would be 7.6 times the size of Mt Everest in diameter and would be able to fit 413,022 football fields in it.

* 1. Let's try a smaller velocity, like 0.5c. After some trial and error, I found that Ro = 1.85 \* 1010 m gives a velocity of 1.516 \* 108 m/s, which is pretty close. So, now we get:

DsDt = 47,841 m2

Ds = 5.656, so Dt = 8,458 m = 8.458 km, almost the height of Mt Everest!

Area of laser array = 56.191 km2, can fit 7,869 football fields.

* 1. Clearly, this is not feasible in our lifetime. Let’s just stick to 0.2c