# Week of April 4-April 11

1. Calculating initial velocity for Starshot sail
	1. From <https://arxiv.org/abs/1112.3016>, Appendix B, we have the following three equations for a sail being propelled by a laser:
		* 1. a = F / m = 2P/mc
			2. Ro = DsDt/2.44λ
			3. v = √(4PRo/mc)

 where

 c = speed of light

 a = acceleration of sail

 F = force on sail

 m = mass of sail

 P = power emitted by laser

 Ro = distance at which laser is turned off

 Ds = diameter of sail

 Dt = diameter of laser array

 λ = wavelength emitted by laser

 v = velocity of sail after laser is turned off

 and the sail is assumed to be 100% reflective

* 1. From <https://en.wikipedia.org/wiki/Breakthrough_Starshot>, we know the parameters for Breakthrough starshot:
		1. P = 100 GW = 1011 Watts
		2. m = 1 g = 10-3 kg
		3. Ds = √(42+42) = √32 = 5.656 m (sail is 4m x 4m)
		4. Dt = 1km = 103 m
		5. λ = 1.06 μm = 1.06 \* 10-6 m
	2. Plugging Starshot parameters into the equations, we get:
		+ 1. a = 6.66 \* 105 m/s2
				1. For some comparison, our near future sail to Jupiter has an acceleration in the range of 10-3 to 10-6 m/s2.
			2. Ro = 2.186 \* 109 m
				1. For comparison, the Moon is 3.844 \* 108 m away from Earth and Mars is 5.46 \* 1010 m away.
			3. v = 5.398 \* 107 m/s = 0.18c
				1. At this high speed, we should actually be using relativistic equations, but this paper isn’t doing that.
1. Running the simulation with Starshot parameters
	1. First, I ran the simulation with our near future probe, which has a mass of 17.7 kg and area of 4000 m2, setting its initial y velocity to what we calculated in 1c, for 12 years.



It doesn’t really curve around the Sun now, since its velocity is so high that the acceleration due to gravity from the Sun is basically negligible. It also travels 2 \* 1016 m, which is ~ 2 lightyears. It is out of the Oort cloud(1.5\*1016m) and halfway to Proxima Centauri(4.37 lightyears).

* 1. Next, I changed the sail parameters to more closely match those of Breakthrough Starshot, so now our sail has a mass of 1 gram and area of 16 m2. I ran the simulation to find where the sail is when it reaches the approximate location of Proxima Centauri, which is 4238002255618.494, 4.258587936362806e+16. I then placed Proxima Centauri in the simulation, 1AU away from the position of the sail that I just found, and with a mass of 2.446 \* 1029 kg. Running the simulation for 25 years gave this path for the sail:



Curiously, the sail does not go into the negative x now like it did before. The only thing that changed is the mass and area of the sail, so I’m not quite sure why this happened. My best guess would be that since its mass is so low now, it doesn’t get affected by the Sun’s gravity at all?

1. Finding the duration of the “interesting zone”
	1. The “interesting zone” here refers to the radius around Proxima Centauri in which our probe can study the star and the planets around it. We can use the simulation to calculate when our sail is in that radius as well as how long it will be in it.
	2. I added some if statements in the simulation to note the timestep when our sail reaches that radius from the star, as well as the timestep when it leaves that radius, which is given below:

| distance = abs(abs(pos[2]) - abs(pos[1]))science\_proximity = distance < science\_radiusif not doing\_science: # check if we can start doing science if science\_proximity[0] or science\_proximity[1]: doing\_science = True science\_start = ielse: # check if we have to stop doing science if not science\_proximity[0] and not science\_proximity[1]: doing\_science = False science\_end = i |
| --- |

 pos[2] is the position of Proxima Centauri, pos[1] is the current position of the sail, science\_start and science\_end store the timesteps at which the sail enters and leaves the radius, respectively and science\_proximity is a boolean array that stores True if the sail is in the radius in that direction and False if it is not in that radius. For example, if the sail is in the radius in the x but not in the radius in y, science\_proximity will be [True False].

* 1. Through some trial and error, and having to change the duration of the simulation, I found time periods for radii of 1, 2 and 5 AU, which are as follows:

| Radius | Duration of sim | Start day | End day | Days in the zone |
| --- | --- | --- | --- | --- |
| 1 AU | 30 years | 9132 ( September 25, 2047) | 9799 (July 23, 2049) | 667 |
| 2 AU | 30 years | 8797( October 25, 2046) | 10,132(June 21, 2050) | 1,335 (~3.66 years) |
| 5 AU | 40 years | 7792( January 24, 2044) | 11,132( March 17, 2053) | 3,340 (~9 years) |

1. Writing a program for relativistic velocity
	1. Since our sail travels at a big fraction of the speed of light, we need to consider relativity when we calculate the speed of the sail.
	2. From Dr Richmonds notes, it seems that the only difference in our calculation is when calculating acceleration.
		1. For normal calculations:

 acc = force / masssail

* + 1. For relativistic calculations:

𝛾 = 1 / (√(1 - (v/c)2))

masseff = masssail \* 𝛾

acc = force / masseff

* 1. After calculating acceleration, the rest of the calculation is same for both:

vnew = v + acc \* dt

xnew = x + ½(v + vnew)dt

t = t + dt

x = xnew

v = vnew

where dt is the timestep, v is the original velocity and x is the original distance.

We do this calculation in a loop until we reach the distance we want to.

* 1. Writing the program was simple enough, but I ran into a problem when running it. When calculating relativistic velocity for a distance of 7.786 \* 1011 m, the program reached a point where v = c, and so 𝛾 became 1 / 0. Now, the fix might be to add a softening parameter like in the simulation, but something about this feels wrong to me. Why does our velocity become equal to c? Isn’t our calculation trying to prevent that from happening?