JOURNAL OF
EMERGING INVESTIGATORS

Article

The journey to Proxima Centauri b

Shreyas Ramaswamy[1], Mark R. Giovinazzi[2]
[1]Leland High School, San Jose, California
[2]Department of Physics and Astronomy, University of Pennsylvania, Philadelphia, Pennsylvania

SUMMARY

In recent years, exoplanets have become one of the most exciting topics in astronomy. While sending humans to a planet light-years away from Earth remains science fiction, recent technological advancements have made the prospect of sending a small probe to an exoplanet not only fascinating but also realistic. In addition to time and ample resources, developing such a probe and guiding it to a world beyond our solar system would require a deep understanding of gravitational dynamics. Here, we considered a simulated journey to Proxima Centauri b to determine the feasibility that a rocket could travel there in the timescale of a human lifetime. We hypothesized that a rocket mission from Earth to Proxima Centauri b (including the sending of data from the rocket back to Earth) could be carried out within a human lifetime and that the gravitational influence of celestial objects would be minimal in affecting the trajectory. We found that a rocket traveling at an average speed of 3 x 10^7 m/s can reach Proxima Centauri b in as little as 50 years, including acceleration and deceleration times. We used VPython to visually simulate the entire trajectory. Our results demonstrated that interstellar travel to other exoplanets is feasible with sufficiently powerful rockets.

INTRODUCTION

People have long wondered about the possibility of travelling beyond Earth. However, this feat has been technologically impossible until very recently. With current technology, a mission to another star system could be possible, though likely economically unfeasible. For example, the unmanned Voyager I and II, launched in 1977 to explore the outer planets of the Solar System, have gold plates with attached messages for a potentially intelligent alien audience (1).

In the universe, the fastest speed that any object can travel at is 3 x 10^8 m/s or the speed of light. As of 2023, the highest speed any human-made object has traveled was 1.68 x 10^5 m/s by the Parker Solar Probe, which used gravity assists of the Sun and Venus to reach such speeds (2). In addition to conventional engines (liquid fuel, solid fuel, cold gas, and ion), rockets depend on techniques of gravity assists to catapult in space (3,4). Conventional rockets can theoretically reach very high speeds, but they would require large amounts of fuel to constantly accelerate in space (4). As the fuel requirement increases, it becomes more difficult for the rocket to launch due to the additional mass. Since we can only reach speeds which are non-relativistic (without affecting the mass of the rocket) and we have a fuel limit which restricts both our maximum speed and maximum distance traveled, the rocket can only travel so far before it runs out of fuel.

Currently, we know of more than 5,000 exoplanets, or planets beyond our solar system (5). The first exoplanet ever discovered was 51 Pegasi b in 1995 (6). At only ~4.4 light-years (ly) away, the Alpha Centauri system is the closest stellar system to the Sun. The triple star system consists of Alpha Centauri A, Alpha Centauri B, and Proxima Centauri. Alpha Centauri A and Alpha Centauri B comprise a binary star system in an elliptical orbit (e = 0.51947 ± 0.00015) with a period of P = 79.762 ± 0.019 years (7). There are no known planets within this system. Proxima Centauri is in a more distant orbit (P ~550,000 years) around the Alpha Centauri AB system. It has one known planet, Proxima Centauri b, which is the closest exoplanet to Earth and the focus of this paper (8). Important observable properties of the Alpha Centauri system include radial velocity, proper motion in right ascension and declination, and distance (**Table 1**).

Most exoplanets have been found by the transit method, in which scientists detect changes in a star's luminosity as the planet transits, or passes in front of the star (9,10). However, Proxima Centauri b does not orbit in the same plane as Earth, so its transits cannot be detected. Thus, scientists discovered it using the radial velocity method, in which a star wobbles due to the influence of its planets. Proxima Centauri b has an equilibrium temperature relatively close to Earth and is suspected to have a mass very comparable to Earth (about 1.27 Earth-masses) (11). Important estimated parameters for Proxima Centauri b determined by Jenkins et al. include mass, orbital period, eccentricity, and equilibrium temperature (**Table 2**) (9). Given the proximity and similar physical properties to Earth, Proxima Centauri b is an excellent candidate for hypothetical interstellar travel from Earth.

Parameter	Description	Unit	Alpha Centauri A	Alpha Centauri B	Proxima Centauri
α_{2000}	Right Ascension	deg	219.9020583318	219.8960962892	217.42893212396
δ_{2000}	Declination	deg	-60.8339926873	-60.8375275696	-62.67948924884
μ_α	Proper Motion in Right Ascension	mas yr^{-1}	-3679.25	-3614.39	-3781.741
μ_δ	Proper Motion in Declination	mas yr^{-1}	473.67	802.98	769.465
RV	Radial Velocity	km s_{-1}	-21.40	-18.60	-21.94
d	Distance	pc	1.325	1.256	1.30197
V	Johnson V Magnitude	mag	0.01	1.33	11.13

Table 1. Stellar parameters of Proxima Centauri. Astrometric data and photometry were drawn from other sources (13-17). Here, only point the point estimates are shown, which are the values used in the simulation.

https://doi.org/10.59720/23-081

Parameter	Unit	Value
M **(mass)**	**M⊕**	**1.27 (+0.19 / -0.17)**
P **(orbital period)**	**D**	**11.1855 (+0.0016 / -0.0014)**
e **(eccentricity)**	**---**	**0.08 (+0.07 / -0.06)**
T_{eq} **(equilibrium temperature)**	**C°**	**-39**

Table 2. Physical characteristics of Proxima Centauri b (18).

In this paper, we describe the hypothetical voyage of a small rocket from Earth to Proxima Centauri b. We aimed to find the best range of parameters for velocity and angle for our rocket launch such that the length of the journey would be within one human lifetime. We also wanted to study the impact of the Sun, Alpha Centauri ABC, and all the other celestial objects on the rocket trajectory. To test this, we simulated the trajectory of the rocket in VPython and automated gravitational influence from every orbital body. We then varied the velocity from 3 x 10^6 m/s to 5 x 10^7 m/s with a goal to understand what velocity it would take for a rocket to travel in a nonrelativistic way (significantly less than 3 x 10^8 m/s) to reach Proxima Centauri b within one human lifetime. We varied from where on Earth's surface the rocket launches to determine if the launch position significantly affects the ETA (estimated time arrival) of the rocket. Theoretically, the rocket launch we aim to simulate is possible, however, it would require huge amounts of fuel; in our simulation, we circumvent this issue by assuming infinite fuel because we want to find out what non-relativistic speed will allow the rocket to reach Proxima Centauri b within a human lifetime.

RESULT

We discretized Newton's Second Law to make it suitable for algorithmic implementation, where we used the Impulse formula to predict the next position and velocity of the rocket at each step. The Impulse formula incorporated the gravitational force felt by the rocket by all other objects. We used a function

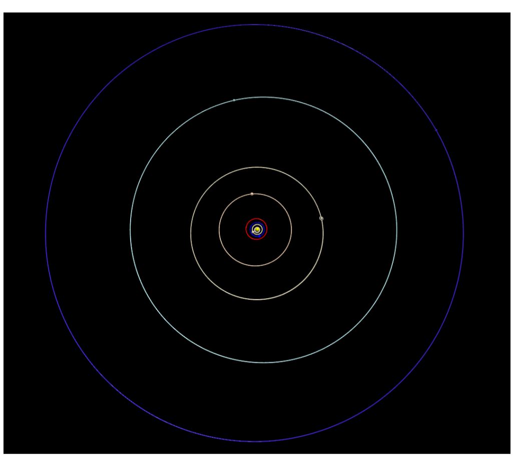

Figure 1. Solar system as shown in our simulation. Newton's law of gravitation and the Euler Cromer Algorithm, coupled with Newton's Law of Gravitational Motion were used to create the simulation. The Sun, Mercury, Venus, Earth, Mars, Jupiter, Saturn, Uranus, and Neptune in that respective order from center are shown, with balls representing the celestial bodies and circles representing their orbits around the Sun.

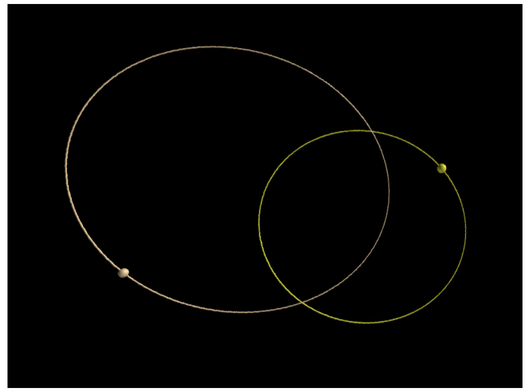

Figure 2. Alpha Centauri system as shown in our simulation. Newton's law of gravitation and the Euler Cromer Algorithm, coupled with Newton's Law of Gravitational Motion were used to create the simulation. The bright yellow ball represents Alpha Centauri A, the most massive of the Alpha Centauri AB system. The tan ball represents Alpha Centauri B, its partner binary star. The interlocking orbits indicate that this system is a binary system.

to calculate this force due to every other object.

We varied the initial launch angle and velocity and used the Rodrigues Formula to align the rocket's velocity with Proxima Centauri b. As the rocket approached the planet, it gradually decelerated until it was captured. We conclude that their simulation can accurately model the rocket's journey to Proxima Centauri b.

We simulated the entire Solar System and Alpha Centauri ABC systems using Glowscript and VPython to account for their effect on the rocket's journey (**Figures 1 - 3**). These stellar systems were completely governed by Newtonian Mechanics. Every planet had a stable orbit around its parent star, and the stellar systems were very accurate (same inclination, proper position with respect to the Sun, correct orbital patterns) models of their real-world counterparts. In our simulation, as the rocket hurtled towards Proxima Centauri b, the angle between the velocity vector and position vector (theta in the Rodrigues formula) kept constantly changing as it kept adjusting towards the planet's location (**Figure 4**).
We were able to plot out both the ETA with respect to the rocket's launch velocity, and the ETA with respect to launch angle theta. For different launch velocities, the ETA changes and can vary from 30 years, with an initial speed of 5 x 10^7 m/s, to 512 years, with a speed of 3 x 10^6 m/s (**Table 3**). The rocket's launch angle from Earth does not significantly change the ETA for a launch speed of 3 x 10^7 m/s; the ETA changes at maximum by 0.06 years. For an initial speed of 3 x 10^7 m/s, launching at angle of 0 degrees with respect to the equator provides an ETA of 50.0515 years, while the ETA

Velocity (m/s)	Time Taken (yrs)
3 x 10^6	512.028
1 x 10^7	149.953
2.7 x 10^7	55.6209
3 x 10^7	50.0515
3.3 x 10^7	45.5106
5 x 10^7	30.1094

Table 3. Different ETAs of rocket journey depending on velocity.

https://doi.org/10.59720/23-081

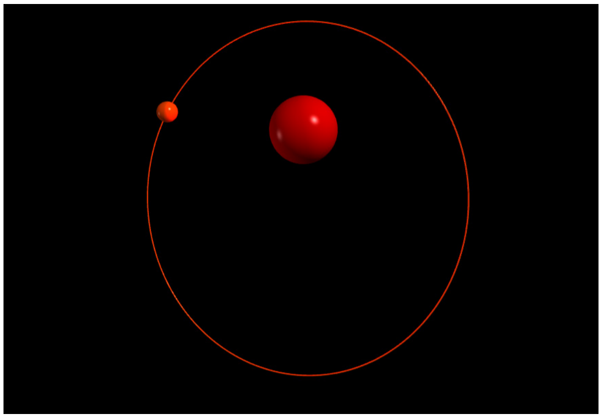

Figure 3. Proxima Centauri system as shown in our simulation. Newton's law of gravitation and the Euler Cromer Algorithm, coupled with Newton's Law of Gravitational Motion were used to create the simulation. Data from ALMA for Proxima Centauri and circular motion formula for Proxima Centauri b. The large red ball represents Proxima Centauri, and the smaller orange ball represents Proxima Centauri b around Proxima Centauri.

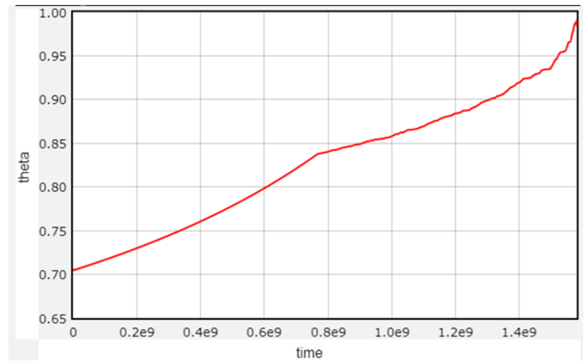

Figure 4. Angle theta vs time during the simulation. Time is in seconds and theta is in radians. This graph models the angle for the entire simulation, beginning as the simulation starts and ending as it ends.

for a launch angle of 90 degrees with respect to the equator provides an ETA of 50.1142 years (**Table 4**).

DISCUSSION

Our simulation demonstrates a complete visualization of a hypothetical rocket travelling from Earth to Proxima Centauri b. Additionally, our program somewhat acts as a sandbox for testing various configurations of planets and stars. The simulation clearly shows that if the velocity is on the order of 10^6 m/s, reaching Proxima Centauri b within 100 years is not possible. However, if the velocity is from the range of 3 x 10^7 m/s to 5 x 10^7 m/s, then it is possible for the rocket to reach Proxima Centauri b within a human lifetime. A rocket travelling 4.4 ly (the approximate straight-line distance from Earth to Proxima Centauri b) at a speed of 3 x 10^7 m/s would arrive after 44 years, which is on the magnitude of 50.0515 years. The discrepancy could be attributed to the rocket taking time to accelerate and being minimally impacted by the other objects in the system. Additionally, we verified that the planetary revolutions around the Sun are also correct, giving us confidence in the results. The variation of the angle is not a significant factor in reducing ETA, only resulting in a couple hours or days difference in the ETA (**Table 4**).

In this simulation, at 3 x 10^7 m/s the rocket would take 50.0515 years to travel to Proxima Centauri b. Due to the nature of the speed of light and other electromagnetic waves,

Launch Angle (Deg)	X (relative to center of Earth)	Y (relative to center of Earth)	Z (relative to center of Earth)	Time Taken (yrs.)
0	r	0	0	50.0515
30	$r \sin(\pi/6)$	$r \cos(\pi/6)$	0	50.0648
45	$r \sin(\pi/4)$	$r \cos(\pi/4)$	0	50.0466
60	$r \cos(\pi/3)$	$r \sin(\pi/3)$	0	50.0698
90	0	r	0	50.1142

Table 4. ETA for Launch Position. *r* refers to Earth radius. Phi assumed to be 0 deg (azimuthal angle)

it would take an additional four years for program data to reach Earth. One limitation of our simulation is that we assumed infinite fuel. In the future, this limitation may be circumvented if the rocket was able to refuel during multiple parts of its journey. But the simulation does prove that even if the velocity of the rocket is nonrelativistic, the length of a rocket journey from Earth to Proxima Centauri b is possible within a human lifetime. If our mission was launched today, many people alive today could live to see the rocket reach Proxima Centauri b. Our hypothetical rocket wasn't designed to accommodate humans because incorporating human passengers would present several additional challenges that are beyond the scope of our analysis. Future missions may even be able to carry humans and achieve quicker mission lengths by accelerating their probe to higher speeds than the ones we considered in our simulation. The rocket could transport rovers and probes to Proxima Centauri b, which could find critical information on the planet. These probes could report if liquid water, important resources, or even life itself exists on the planet. This mission could potentially discover the future home for humanity and could help scientists figure out human interstellar travel. Ultimately, our research confirmed that this rocket voyage is theoretically possible within one human lifetime if future technological improvements allow for speeds of 3-5 x 10^7 m/s.

MATERIALS AND METHODS

To simulate the launch, we used Glowscript, a version of Vpython, which allows for the creation of 3D animations and simulations using the Python programming language. Glowscript's ability to process large amounts of data made it an ideal choice for our simulation, as it allowed us to create a fully functional system composed of the Solar system and the Alpha Centauri system.

In our simulation, we implemented fourteen main objects: the Sun, Mercury, Venus, Earth, Mars, Jupiter, Saturn, Uranus, Neptune, Alpha Centauri A, Alpha Centauri B, Proxima Centauri, Proxima Centauri b, and the rocket. We used data from NASA's Horizons database as well as ALMA and TESS observational data to model the orbital mechanics of these objects. Although the observational data provided ephemerides in ecliptic galactic coordinates, we were able

https://doi.org/10.59720/23-081

to convert them to vector data using external software. With the position and velocity vector coordinates, we were able to simulate Newton's gravitational law:

$$F = \frac{GMm}{r^2} \text{ (Eqn 1)}$$

using the Euler-Cromer algorithm:

$$p_{next} = p_{current} + F_{total}\Delta t \text{ (Eqn 2a)}$$
$$r_{next} = r_{current} + \frac{p_{next}}{m}\Delta t \text{ (Eqn 2b)}$$

which updates the positions of all bodies using the impulse formula.

The journey to Proxima Centauri b in our simulation begins with the rocket positioned on the surface of Earth, at one earth radius away from the core. The rocket accelerates at 30 m/s^2 to a velocity of 30 million m/s. We chose the rocket's acceleration to be 30 m/s^2 because it is comparable to the acceleration of cubesat probes which use the Earth's magnetic field to achieve accelerations around 5*g* (12), where *g* is the gravitational acceleration on Earth of 9.8 m/s^2 (12). The rocket begins at rest and accelerates to a velocity of 3×10^7 m/s or roughly 10% of the speed of light. In our simulation, we can vary the initial launch angle, velocity, and acceleration. Once the rocket leaves the Solar System, we align its velocity direction along the position vector of Proxima Centauri b from the rocket by utilizing the Rodrigues Formula (**Equation 3**):

$$\mathbf{v}_{\mathrm{rot}} = \mathbf{v}\cos\theta + (\mathbf{k}\times\mathbf{v})\sin\theta + \mathbf{k}(\mathbf{k}\cdot\mathbf{v})(1-\cos\theta) \text{ (Eqn 3)}$$

Vector **v** refers to the rocket's instantaneous velocity, vector **r** refers to the difference in the position vectors of Proxima Centauri b and the rocket, and vector **k** is the unit vector perpendicular to vector **v** and **r**. Theta is the angle between vector **v** and the position vector **r**. The entire code used for our simulation is given in the Appendix.

As the rocket approaches approximately 100 au from Proxima Centauri b, it gradually decelerates until it reaches a velocity of 100,000 m/s, and then decelerates even further to a velocity of 1,500 m/s. When the rocket is within 0.1 au of the planet, it further decelerates to 1,000 m/s and continues to aim for the planet until it is captured and lands on the planet.

Received: February 4, 2023
Accepted: October 2, 2023
Published: April 1, 2024

REFERENCES

1. Kohlhase, C.E., and Penzo P.A. "Voyager Mission Description." *Space Science Reviews*, vol. 21, no. 2, Nov. 1977, https://doi.org/10.1007/bf00200846.
2. Garner, R. "Parker Solar Probe: Humanity's First Visit to a Star." NASA, 2019, www.nasa.gov/content/goddard/parker-solar-probe-humanity-s-first-visit-to-a-star.
3. "Rocket Propulsion." Nasa.gov, 2019, www.grc.nasa.gov/WWW/K-12/VirtualAero/BottleRocket/airplane/rocket.html.
4. "Gravity Assists - NASA Science." Science.nasa.gov, science.nasa.gov/mission/cassini/gravity-assists/.
5. Brennan, P. NASA's Exoplanet Exploration. "Cosmic Milestone: NASA Confirms 5,000 Exoplanets." Exoplanet Exploration: Planets beyond Our Solar System, 21 Mar. 2022, exoplanets.nasa.gov/news/1702/cosmic-milestone-nasa-confirms-5000-exoplanets/.
6. Michel, M. and Queloz, D. "A Jupiter-Mass Companion to a Solar-Type Star." *Nature*, vol. 378, no. 6555, Nov. 1995, https://doi.org/10.1038/378355a0.
7. Akeson, R., et al. "Precision Millimeter Astrometry of the α Centauri AB System." *The Astronomical Journal*, vol. 162, no. 1, 2021, p. 14., https://doi.org/10.3847/1538-3881/abfaff.
8. Anglada-Escudé, G., et al. "A Terrestrial Planet Candidate in a Temperate Orbit around Proxima Centauri." *Nature*, vol. 536, no. 7617, Aug. 2016, https://doi.org/10.1038/nature19106.
9. Jenkins, J.S., et al. "Proxima Centauri B Is Not a Transiting Exoplanet." *Monthly Notices of the Royal Astronomical Society*, vol. 487, no. 1, 2019, pp. 268–274., https://doi.org/10.1093/mnras/stz1268.
10. Gilbert, E.A., et al. "No Transits of Proxima Centauri Planets in High-Cadence Tess Data." *Frontiers in Astronomy and Space Sciences*, vol. 8, 2021, https://doi.org/10.3389/fspas.2021.769371.
11. "Proxima B 3D Model." Exoplanet Exploration: Planets beyond Our Solar System, exoplanets.nasa.gov/resources/2211/proxima-b-3d-model/.
12. Chen, Z., and Zosimovych, N. "Mission Capability Assessment of 3D Printing Cubesats." *IOP Conference Series: Materials Science and Engineering*, vol. 608, no. 1, 1 Aug. 2019, p. 012025, https://doi.org/10.1088/1757-899x/608/1/012025
13. Perryman, M. A. C., Lindegren, L., Kovalevsky, J., et al.1997, *Astronomy & Astrophysics*, 323, L49
14. Lindegren, L., and Kovalevsky, J. "Accuracy Predictions and Final Prospects for the Hipparcos Mission." *Highlights of Astronomy*, vol. 9, 1992, pp. 425–428., https://doi.org/10.1017/s1539299600009436.
15. Ducati, J. R. 2002, VizieR Online Data Catalog
16. Prusti, T., et al. "The Gaia Mission." *Astronomy & Astrophysics*, vol. 595, 2016, https://doi.org/10.1051/0004-6361/201629272.
17. Vallenari, A., et al. "Gaia Data Release 3. Summary of the Content and Survey Properties." *Astronomy & Astrophysics*, 2022, https://doi.org/10.1051/0004-6361/202243940.
18. Scheucher, M., et al. "Proxima Centauri b: A Strong Case for Including Cosmic-Ray-Induced Chemistry in Atmospheric Biosignature Studies." *The Astrophysical Journal*, vol. 893, no. 1, 2020, p. 12., https://doi.org/10.3847/1538-4357/ab7b74.

APPENDIX

Web VPython 3.2

```
#scene.width=5000
#scene.height=1000

Condition = True

G=6.6743e-11
ProxPos = vector(-1.464074883266791e+16,-2.445770022093117e+16,-2.8303679505000696e+16)
Proxvel = vector(-23338.528508856114,-4458.583679312704,-22260.070747468948)

ProxBPos = ProxPos + vector(2.956872716616501e+10 * 0.036,-1.486628925113508e11 * 0.05,3.626784714139253e07 * 0.07)
ProxBvel = vector(29000.8801,8540.8801,21590.8801) + Proxvel

AlphaAPos = vector( -1.54570165308796e+16, -2.619862024552269e+16, -2.799922474274963e+16)
AlphaAvel = vector( -23369.05456409598,-5737.851065498633,-20834.78962398047)

AlphaBPos = vector(-1.5456223574020744e+16, -2.619674784461988e+16,  -2.8001414324910252e+16)
AlphaBvel = vector(-22957.09101744183,-2706.225462589673,-19095.766750169452)

perES=vector(-3.510010400372842e+10,1.436627446787814e+11,1.949047737010568e+07)
perESvel=vector(-2.949271425336049e+04,-6.981134783410363e+03,1.333958105995237)
#sphere(pos=perES,radius=1e9,color=color.green)
ApHeES=vector(2.956872716616501e+10,-1.486628925113508e11,3.626784714139253e07)
#sphere(pos=ApHeES,radius=1e9,color=color.cyan)
#-5.856718246442796E+07 Y =-4.568711563588172E+06 Z = 4.999740883309308E+06


graph(xtitle="time", ytitle="theta")
pos_curve = gcurve(color=color.red)

perMS=vector(1.876736074724336e11,-8.342770429566579e10,-6.382887523725599e9)
perMSvel=vector(1.081727934599030e4,2.417573213230074e4,2.414531471264070e2)
sphere(pos=perMS,radius=1e9,color=color.orange)
ApHeMS=vector(0.956*(-2.292172344791432e11),0.956*(1.00375973370691e11),0.956*(7.729951209840350e9))
#sphere(pos=ApHeMS,radius=1e9,color=color.white)

perMec = vector(-5.85e10, -4.57e9, 4.99e9)
perMecVel = vector(-6.344720545470197e3, -4.647194187026749e4, -3.215236916087221e3)

perVen = vector(4.088411107047053E+010, -1.008596765262628E+011, -3.742228894769266E+09)
perVenvel = vector(3.222056550449575E+04, 1.303788431766580E+04,-1.680558074024945E+3)

perJup = vector(-8.122974055587324E+011, -7.265345454212829E+010,1.847783819886932E+010)
perJupvel = vector(1.009610908172987E+3, -1.240955470313854E+04, 2.894270125602105E+1)

perSat = vector(-3.366082966314669E+11, -1.463567381125879E+12, 3.883799305400991E+10)
perSatvel = vector(8.889436374098597E+3, -2.205265850944856E+3, -3.153519021935418E+2)

perUra = vector(2.760616974913828E+12,1.132727017090241E+12,-3.153861067456472E+10)
perUravel = vector(-2.631861851002263E+3,5.970743603278066E+3,5.639264556887014E+1)

perNep = vector(4.227853026694131E+12,-1.483084701171259E+12,-6.689301919715136E+10)
perNepvel = vector(1.765104067248427E+3, 5.148951233886568E+3,-1.470716588704148E+2)

sun=sphere(pos=vector(0,0,0), radius=9.1e9,color=color.yellow, make_trail = True)
sun.m=2e30
sun.v=vector(0,0,0)
sun.r = 2.1e9
sun.p = sun.m*sun.v

earth=sphere(pos=perES,radius=6.4e9,color=color.blue,make_trail=True)
earth.m=6e24
earthradius1 = 6.4e6
earth.v=perESvel
earth.p=earth.m*earth.v

#mars
```

```
mars=sphere(pos=perMS,radius=3.4e9,color=color.red, make_trail=True)
mars.m=6.4e23
#mars.m = 2*9.99e27*3
mars.v=perMSvel
mars.p=mars.m*mars.v

mercury = sphere(pos = perMec, radius = 2.44e9, color = color.white,make_trail=True)
mercury.m = 3.301e23
mercury.v = perMecVel
mercury.p = mercury.m*mercury.v

#Venus
venus = sphere(pos = perVen, radius = 6.05e9, color = vector(1.000, 0.964, 0.807),make_trail=True)
venus.m = 4.87e26
#venus.m = 2*9.99e27
venus.v = perVenvel
venus.p = venus.m*venus.v

#Jupiter
jupiter = sphere(pos = perJup, radius = 6.99e9, color = vector(1.000, 0.856, 0.724),  make_trail=True)
jupiter.m = 1.89e27
jupiter.v = perJupvel
jupiter.p = jupiter.m*jupiter.v

saturn = sphere(pos = perSat, radius = 5.82e9, color = vector(0.886, 0.856, 0.724), make_trail = True)
saturn.m = 5.86e23
saturn.v = perSatvel
saturn.p = saturn.m * saturn.v

Satring = ring(pos = perSat, radius=5.82e9*1.5, thickness=1.00e9, axis = vector(0.3,0.5,1))


uranus = sphere(pos = perUra, radius = 5.08e9, color = vector(0.765, 0.982, 1.000), make_trail = True)
uranus.m = 8.68e25
uranus.v = perUravel
uranus.p = uranus.m*uranus.v

neptune = sphere(pos = perNep, radius = 4.92e9, color = vector(0.276, 0.161, 0.887), make_trail = True)
neptune.m = 1.02e26
neptune.v = perNepvel
neptune.p = neptune.m * neptune.v




# Alpha Centauri System

Proxima = sphere(radius = 1.0728e9,  pos = ProxPos, color = vector(0.899, 0.000, 0.000), make_trail = True)
Proxima.m = 2.446e29
Proxima.v = Proxvel
Proxima.p = Proxima.m * Proxima.v

ProximaB = sphere(radius = 6.888388e8, pos = ProxBPos, color = vector(0.977, 0.261, 0.000), make_trail = True)
ProximaB.m = 7.589e24
ProximaB.v = ProxBvel
ProximaB.p = ProximaB.m * ProximaB.v

ACA = sphere(radius = 8.51117668e10, pos = AlphaAPos, color = vector(1.000, 0.997, 0.288), make_trail = True)
ACA.m = 2.188e30
ACA.v = AlphaAvel
ACA.p = ACA.m*ACA.v

ACB = sphere(radius = 6.00784209e10, pos = AlphaBPos, color = vector(1.000, 0.866, 0.660), make_trail = True)
ACB.m = 1.804e30
ACB.v = AlphaBvel
ACB.p = ACA.m * ACA.v

#ROCKET
```

```
rocket = box(pos=earth.pos + vector(0,earthradius1,0), color = color.green, size = vector(6.4e7, 6.4e7, 6.4e7), axis = vector(-1,0,0), make_trail = True)


theta = (rocket.pos.z - Proxima.pos.z)/(mag(rocket.pos - Proxima.pos))
phi = acos((rocket.pos.x - Proxima.pos.x)/(mag(rocket.pos - Proxima.pos)*sin(theta)))



thetaA = (rocket.pos.z - Proxima.pos.z)/(mag(rocket.pos - Proxima.pos))
phiA = acos((rocket.pos.x - Proxima.pos.x)/(mag(rocket.pos - Proxima.pos)*sin(theta)))



Launched = True
VF = 30000000
launchvelocity=0
AR = 30


#thetaA = 0 * pi/180
#phiA = 15 * pi/180

AZ = AR*cos(thetaA)
AX = AR *sin(thetaA)*cos(phiA)
AY = AR*sin(thetaA)*sin(phiA)

A = vector(AX, AY, AZ)

vz = launchvelocity*cos(theta)
vx = launchvelocity*sin(theta)*cos(phi)
vy = launchvelocity*sin(theta)*sin(phi)






rocket.m = 74842.741
rocket.relativevelocity = vector(vx,vy,vz)
rocket.v = earth.v - rocket.relativevelocity
rocket.p= rocket.m*rocket.v



scene.autoscale=False
scene.camera.follow(rocket)


userpan=False
dt = 5
t=0

def Ftotal (object):
    list = [rocket,sun,mercury,venus,earth, mars,jupiter,saturn,uranus,neptune, ACA, ACB, Proxima, ProximaB]
    if object in list:
        index = list.index(object)
        list.pop(index)

        a = list[0]
        b = list[1]
        c = list[2]
        d = list[3]
        e = list[4]
        f = list[5]
        g = list[6]
        h = list[7]
        i = list[8]
        j = list[9]
```

```
        k = list[10]
        l = list[11]
        m = list[12]

        ra = object.pos - a.pos
        rb = object.pos - b.pos
        rc = object.pos - c.pos
        rd = object.pos - d.pos
        re = object.pos - e.pos
        rf = object.pos - f.pos
        rg = object.pos - g.pos
        rh = object.pos - h.pos
        ri = object.pos - i.pos
        rj = object.pos - j.pos
        rk = object.pos - k.pos
        rl = object.pos - l.pos
        rm = object.pos - m.pos

        FA = -ra.hat * G * a.m * object.m/mag2(ra)
        FB = -rb.hat*G*b.m*object.m/mag2(rb)
        FC = -rc.hat*G*c.m*object.m/mag2(rc)
        FD = -rd.hat*G*d.m*object.m/mag2(rd)
        FE = -re.hat*G*e.m*object.m/mag2(re)
        FF = -rf.hat*G*f.m*object.m/mag2(rf)
        FG = -rg.hat*G*g.m*object.m/mag2(rg)
        FH = -rh.hat*G*h.m*object.m/mag2(rh)
        FI = -ri.hat*G*i.m*object.m/mag2(ri)
        FJ = -rj.hat*G*j.m*object.m/mag2(rj)
        FK = -rk.hat*G*k.m*object.m/mag2(rk)
        FL = -rl.hat*G*l.m*object.m/mag2(rl)
        FM = -rm.hat*G*m.m*object.m/mag2(rm)

        Ftotal = FA + FB+ FC + FD + FE + FF+ FG + FH + FI + FJ + FK + FL + FM
        return Ftotal


while Condition == True:
    rate(100000000)
    FS = Ftotal(sun)
    sun.p = sun.p + FS*dt
    sun.pos = sun.pos + sun.p/sun.m * dt


    if Launched == False:
        rocket.v = earth.v
        Frock = vector(0,0,0)
        rocket.pos=earth.pos + vector(earth.radius,0,0)
        rocket.v = earth.v

    else:
        if mag(rocket.relativevelocity) < VF:
            Frock = Ftotal(rocket)
            rocket.relativevelocity=rocket.relativevelocity + A*dt
            rocket.v = earth.v - rocket.relativevelocity
            rocket.p= rocket.m * rocket.v
            rocket.p=rocket.p+ Frock*dt
            rocket.pos=rocket.pos + (rocket.p/rocket.m)*dt

        else:
            v = rocket.v
            r = ProximaB.pos - rocket.pos
            k = cross(v.hat, r.hat)

            theta = acos(dot(v.hat,r.hat)/(mag(v.hat)*mag(r.hat)))
            theta = (rocket.pos.z - ProximaB.pos.z)/(mag(rocket.pos - ProximaB.pos))
            rocket.v = v*cos(theta) + cross(k.hat, v)*sin(theta) + k.hat*(dot(k.hat,v))*(1-cos(theta))
            #if t%39455:
            #    print(mag(rocket.v),mag(Proxima.pos-rocket.pos),t/(3600*24*365))
            rocket.p = rocket.m*rocket.v
            #rocket.p = rocket.p + Frock*dt
```

```
        #rocket.pos = rocket.pos + (rocket.p/rocket.m)*dt
        Frock = Ftotal(rocket)
        rocket.p=rocket.p+ Frock*dt
        rocket.pos=rocket.pos + (rocket.p/rocket.m)*dt


        dt=4.5e3

        if mag(rocket.pos-ProximaB.pos)<1.496e+11:
            rocket.v = rocket.v*0.7

    pos_curve.plot(t, theta)



    FE = Ftotal(earth)
    earth.p=earth.p+FE*dt
    earth.pos=earth.pos+ (earth.p/earth.m)*dt


    FM = Ftotal(mars)
    mars.p=mars.p+FM*dt
    mars.pos=mars.pos+ (mars.p/mars.m)*dt

    FMe = Ftotal(mercury)
    mercury.p = mercury.p + FMe*dt
    mercury.pos = mercury.pos + mercury.p/mercury.m * dt

    FV = Ftotal(venus)
    venus.p = venus.p + FV*dt
    venus.pos = venus.pos + venus.p/venus.m * dt

    FJ = Ftotal(jupiter)
    jupiter.p = jupiter.p + FJ*dt
    jupiter.pos = jupiter.pos + jupiter.p/jupiter.m * dt

    FSa = Ftotal(saturn)
    saturn.p = saturn.p + FSa*dt
    saturn.pos = saturn.pos + saturn.p/saturn.m * dt
    Satring.pos = saturn.pos

    FU = Ftotal(uranus)
    uranus.p = uranus.p + FU*dt
    uranus.pos = uranus.pos + uranus.p/uranus.m * dt

    FN = Ftotal(neptune)
    neptune.p = neptune.p + FN*dt
    neptune.pos = neptune.pos + neptune.p/neptune.m * dt

    FPROX = Ftotal(Proxima)
    Proxima.p = Proxima.p + FPROX*dt
    Proxima.pos = Proxima.pos + (Proxima.p/Proxima.m - (ACA.p+ACB.p)/(ACA.m + ACB.m))* dt
    #Proxima.pos = Proxima.pos + (Proxima.p/Proxima.m - Proxima.p/Proxima.m)* dt

    FProxB = Ftotal(ProximaB)
    ProximaB.p = ProximaB.p + FProxB*dt
    ProximaB.pos = ProximaB.pos + (ProximaB.p/ProximaB.m - (ACA.p + ACB.p)/(ACA.m + ACB.m))*dt

    FRIGIL = Ftotal(ACA)
    ACA.p = ACA.p + FRIGIL*dt
    ACA.pos = ACA.pos + ((ACA.p/ACA.m - (ACA.p+ACB.p)/(ACA.m+ACB.m) ))* dt
    #ACA.pos = ACA.pos + ((ACA.p/ACA.m - Proxima.p/Proxima.m))* dt

    FTOLIMAN = Ftotal(ACB)
    ACB.p = ACB.p + FTOLIMAN*dt
    ACB.pos = ACB.pos + (ACB.p/ACB.m - (ACA.p+ACB.p)/(ACB.m + ACA.m)) * dt
    #ACB.pos = ACB.pos + (ACB.p/ACB.m - Proxima.p/Proxima.m) * dt
```

```
if mag(rocket.pos - ProximaB.pos) < 6.88e11:

    print("The rocket has landed!")
    print("It has taken ", t/(365*24*60*60)," years to reach Proxima B")
    Condition = False

t = t + dt
```