The Yarkovsky Effect
Modeling the Effect of Solar Radiation on the Orbit of Asteroid (308635) 2005 YU55

Andrew Block
Department of Physics and Astronomy, Minnesota State University Moorhead

March 9, 2021

Abstract

This project examines the effect of solar radiation on the orbit of the asteroid (308635) 2005 YU55 by using the python package REBOUND. This effect is is known as the Yarkovsky Effect, and is modeled as a constant force acting on the asteroid. 2005 YU55 was simulated backward in time for a period of ten years, with and without the Yarkovsky Effect. In both instances, the date and time of the closest approach the asteroid made to Earth was 0.0021 ± 0.0001 AU on 2011-11-09 at 00:00 UTC ± 2.4 hours. There was no observed impact on the orbit of the asteroid.

1 Introduction

The first asteroid discovered was the minor planet Ceres by Italian astronomer Giuseppe Piazzi in January 1801[1]. Since then the number of asteroids discovered has grown considerably, with the Minor Planet Center having data on over one million objects[2]. Approximately 1700 of them are classified as "potentially hazardous objects", meaning that its orbit brings it with a close proximity to Earth. To that end, it's been decided that determining the orbits of these asteroids is beneficial. Determining these orbits is an N-body problem, where N is the number of gravitationally bound bodies in the system of interest. For N $>$ 2, an exact solution is not possible. Obtaining orbits for the bodies requires a computational estimate for the orbit.

The Yarkovsky Effect[3] describes how an asteroid's motion through space can change due to solar radiation striking the asteroid, warming the surface. The warmed asteroid emits photons, which creates a thrust on the side of the asteroid that was exposed to the sun, moving the asteroid slightly. The goal of this project is to apply the Yarkovsky Effect to translational motion and examine how the orbit of an asteroid changes as a result of being struck by solar radiation. Specifically, this projects examines what effect, if any, the Yarkovsky Effect has on asteroid (308635) 2005 YU55, referred to hereafter as 2005 YU55.

The Python package REBOUND[4] is used to model the first five planets of the solar system, as well as the asteroid. The thrust caused by the emitted photons can be added into the code, creating a model of the asteroid's orbit accounting for solar radiation forces.

This asteroid has made several close passes to Earth in recent years, most recently on 2011-11-08 at 23:28 UTC, when it passed within 0.85 lunar distances from Earth[5], meaning small changes in its orbit could bring it into a potential collision course with Earth. This project attempts to determine if solar radiation is something that should be considered when determining the orbits of asteroids and other potentially hazardous objects.

2 Methods

This project had two parts. The first part was determining an expression for the thrust created when the asteroid warms due to the solar radiation. The second part was taking that expression and using it in REBOUND to model the orbit with this force applied.

2.1 Force Expression

For the sake of keeping things simple, this process assumes that the Yarkovsky Effect is a constant force acting on the asteroid.

To find the total force acting on the asteroid, the number of photons emitted by the asteroid must be found. The number density of blackbody photons with wavelength between λ and $\lambda + d\lambda$ is given by dividing the blackbody energy $u_\lambda d\lambda$ in the wavelength range by the photon energy $h\nu = hc/\lambda$

$$n_\lambda d_\lambda = \frac{\lambda u_\lambda}{hc} d\lambda \ , \tag{1}$$

Where $u_\lambda d\lambda$ is given by[6]:

$$u_\lambda d\lambda = \frac{8\pi hc/\lambda^5}{exp(hc/kT) - 1}$$

Using this expression for u_λ, number density of photons is[6]:

$$n_\lambda d\lambda = \frac{8\pi/\lambda^4}{exp(hc/\lambda kT) - 1} d\lambda \tag{2}$$

The total number density of photons is given as:

$$n = \int_0^\infty n_\lambda d\lambda = \int_0^\infty \frac{8\pi/\lambda^4}{exp(hc/\lambda kT) - 1} d\lambda \tag{3}$$

Which can be evaluated to be:

$$n = 16\pi\zeta(3)\left(\frac{kT}{hc}\right)^3$$

ζ is the Riemann zeta function, which has the value $\zeta(3) \simeq 1.202$

The average temperature of the side of an asteroid exposed to sunlight has been shown to be $\approx$ 200 K[7]. The number density of photons released by an asteroid with this temperature is

$$n = 1.63 \times 10^{14} m^{-3}$$

Photons emitted by the asteroid follow the black body distribution, and their peak wavelength can be estimated with Wien's Displacement Law[6]

$$\lambda_{max} = \frac{b}{T}, \tag{4}$$

where T is the temperature of the blackbody and b is $2.89 \times 10^{-3} m \cdot K$. Substituting in the temperature gives a peak wavelength of 1.4×10^{-5} m, which is in the infrared range.
From this, the photon energy can be found[6]

$$E_{photon} = \frac{hc}{\lambda},$$

h is planck's constant, 6.626×10^{-34} Js, and c is the speed of light, 3.0×10^{8} m/s. In this case $E_{photon} = 1.4 \times 10^{-20}$ J per photon.

The radiation pressure produced by photons is given as[6]

$$P_{rad} = \frac{1}{3} u \tag{5}$$

where u is the energy density. The energy density can be found by multiplying the energy per photon by the number density of photons. The energy density is then

$$u = E_{photon} \times n = 2.3 \times 10^{-6} J/m^3,$$

and the radiation pressure is

$$P_{rad} = \frac{1}{3} u = \frac{1}{3} \times 2.3 \times 10^{-6} J/m^3 = 7.6 \times 10^{-7} N/m^2$$

Since pressure is defined as a force per area, multiplying this result by the surface area of the asteroid gives a value of the force exerted by the radiation emitted by the asteroid. YU55 has a diameter of 360 $\pm$ 40 m[8]. For the purposes of this model, the asteroid is assumed to be spherical. Taking the largest possible value of the radius to get the largest possible force, the surface area of the asteroid is

$$A = 4\pi \times (200m)^2 = 5.02 \times 10^5 m^2$$

This gives a constant force acting on the asteroid of:

$$F = P \times A = 7.6 \times 10^{-7} N/m^2 \times 5.02 \times 10^5 m^2 = 0.38N$$

This expression for force can then be coded into REBOUND to see how the force changes the orbit.

For a comparison, the magnitude of the gravitational force between the asteroid and the sun can be calculated. The asteroid has an approximate mass of 5.0×10^{10}

kg, based on masses of asteroids of similar size and composition[9]. The magnitude of gravitational force is

$$F_{grav} = \frac{G \times m_{sun} \times m_{asteroid}}{r^2}$$

Using $m_{sun} = 1.99 \times 10^{30}$ kg and $r = 1.15$ AU $= 1.72 \times 10^{11}$ m.

$$F_{grav} = 2.14 \times 10^8 \text{ N}$$

The gravitational force is larger than the thrust created by the warming by a factor of $\approx 10^9$, suggesting that the gravitational force will dominate, potentially to the point of completely nullifying the effect of the solar radiation.

2.2 Computation

Sample code used for modeling and position subtraction can be found in the appendix. All code, as well as the calculations done in determining error used for this project can be found at `https://git.io/vFP4b`

REBOUND is an astrodynamics package written in C and Python. The force found in the previous section was added to the code to see how the solar radiation changes the orbit. The orbits for the bodies that have been coded in are calculated by REBOUND using ias15, which is a 15th order integrator that can calculate orbits accurately for a large number ($\approx 10^9$) of orbits[10].

First, the simulation is created and the Sun, the first five planets of the solar system, and the asteroid were loaded into the simulation. From here, the integration of the bodies can be done. A time step, an array of times, and an array of positions in the x and y directions must be initialized. From here, the integration was done inside of a for loop. For each iteration through the loop, the x and y positions of the planets and the asteroid were added to the position lists that were originally defined. From here, plots of the positions of the asteroids can be made, as well as a plot of the minimum approach distance to Earth, which is known to be 0.85 lunar distances (.00217 AU) on 2011-11-08 [8].

The simulation was run twenty times total, ten times without accounting for the Yarkovsky Effect and ten times with the force included. The first run started on 2017-11-01 at 15:30 UTC and ran ten years into the past. Each subsequent run then started twenty four hours after the last, with the tenth run starting on 2017-11-10 at 15:30 UTC.

Two tests were done to examine the total effect of solar radiation on the asteroid. The first was to check the asteroid's minimum distance from earth during the time period of interest. If there was any effect, then the minimum distance from the Earth, as well as the time at which this minimum occurs should change, even if just a small amount.

The second check was to record the positions of the asteroid at all times and compare the positions during the simulation with no radiation force to the simulation with the radiation force run at the same starting time. Again, if the Yarkovsky Effect had any impact on the orbit, the positions in the simulations should differ.

3 Results

The x and y positions of the asteroids in both cases, with and without the Yarkovsky force accounted for, were stored in arrays and were plotted against time to show the time of closest approach. Figure 1 shows the orbits of the objects in the simulation. The thickest of the orbits is 2005 YU55.

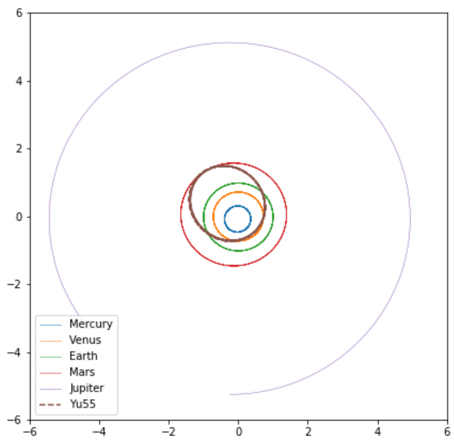


Figure 1: Orbit with only gravitational effects. Axes are in AU.

The simulation was run ten times for each situation, starting at a different day each time. All plots generated by the simulations can be seen at the link provided above. For the sake of brevity, only the plots for the simulation starting on 2017-11-10 are reproduced here.

Figure 2 shows the plot of distance to Earth against time without the Yarkovsky force.

The closest approach according to this plot was at 0.001966 AU at a time of -6.005601 years, or on 2011-11-09 at 02:24:19 UTC.

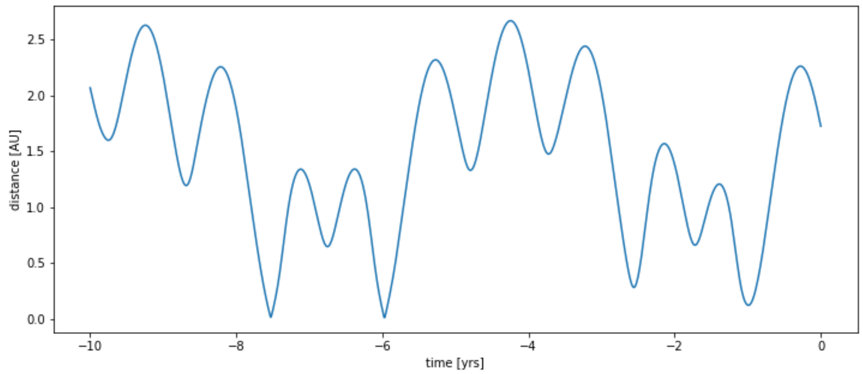


Figure 2: Asteroid distance from Earth against time, not accounting for solar radiation.

Figure 3 shows the plot of closest approach against time with the Yarkovsky force accounted for.

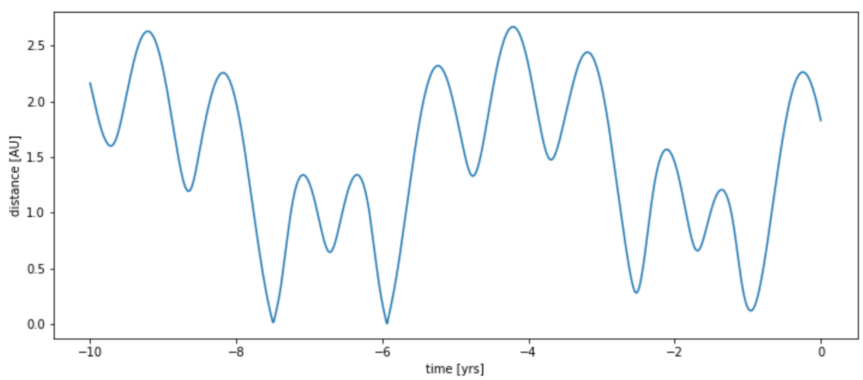


Figure 3: Asteroid distance from Earth against time, accounting for solar radiation.

Adding the Yarkovsky Effect, the closest approach occured at a distance of 0.001966 AU at a time of -6.005601 years. This is the same as when not accounting for the Yarkovsky Effect.

Using each of the ten simulations values for the distance at closest approach, an average value and standard deviation for the closest approach can be found. For the case when not accounting for the Yarkovsky Effect the average value of the distance at closest approach is 0.0021 $\pm$0.0001 AU. When accounting for the Yarkovsky Effect, the average value for the distance at closest approach is 0.0021 $\pm$0.0001 AU. The values are consistent with each other.

The time of the closest approach can also be checked. 2005 YU55 made its closest approach at 23:28 UTC on 2011-11-08. The time of the intercept can be found from the closest approach plots, and the difference between the accepted value for the time at closest approach and the time found in the simulation can be calculated. Again, the times were found to be the same, regardless of the presence of the Yarkovsky Effect, so this calculation only had to be done once.

The Time Calculation code shows the calculation of the average time at closest approach as well as the uncertainty in that measurement. The time at closest approach with uncertainty is then 2455874.5 $\pm$0.1 JD, which corresponds to 2011-11-09 at 00:00 UTC $\pm$ 2.4 hours.

As a second test, the asteroid's position array for both cases can be subtracted from one another to get a sense of how much the force actually moves the asteroid. The greatest change in the asteroid's x position is 0 AU. The greatest change in the asteroid's y direction is 0 AU. This can be seen in the Position Subtraction Code in the link provided above.

4 Conclusion

The discrepancy between the accepted distance at closest approach and the distance found in the simulations is

$$|0.00217 \text{ AU} - 0.0021 \text{ AU}| = 0.00007 \text{ AU}$$

This discrepancy is much smaller than the uncertainty in the measure of distance at closest approach and as such is not significant,

The discrepancy between the accepted time of closest approach and the time found in code can also be examined. The accepted time of closest approach when presented as a Julian Date is 2455874.4 JD. The discrepancy is then

$$|2455874.4 \text{ JD} - 2455874.5 \text{ JD}| = 0.1 \text{ JD}$$

The discrepancy between the accepted time of closest approach and the time found in the simulation is not significant.

The purpose of this project was to determine whether solar radiation has a significant effect in the orbit of the asteroid 2005 YU55. Based on the information presented above, the effect that solar radiation has on the orbit of the asteroid is nonexistent for the time scale examined in this project. Figures 2 and 3 plot the distance the asteroid is from Earth for one of the sets of initial conditions. The average values and uncertainty for the distance and time at closest approach do not change when the Yarkovsky Effect is accounted for.

This project only examined a single asteroid's orbit, but the process could be easily extended to any asteroid, though for asteroids that have properties similar to to YU55, it is reasonable to think that the result would be similar. Additionally, this project also only focused on the magnitude of the force involved. Overall, over the time scale that this project was focused on, the Yarkovsky Effect had no effect on the orbit of the asteroid 2005 YU55.

References

[1] Cunningham, J.C., et al. 2011. Giuseppe Piazzi: The Controversial Discovery and Loss of Ceres in 1801

[2] Galache, J.L, et al. 2015. The Need for Speed in Near-Earth Asteroid Characterization

[3] Vokrouhlicky, D., Broz, M., Bottke, W.F., et al. 2005 . Dynamics of Populations of Planetary Systems, 197, 14

[4] Rein, H., Liu, Shangfei., REBOUND, 2012

[5] Vereshchagina, I. A., Sokov, E. N., Gorshanov, D. L., et al. 2013. Astronomicheskii Vestnik,48,5

[6] Carol, B., Ostlie, D., 2007. An Introduction to Modern Astrophysics, 69, 234-237

[7] Henry, D.B, 1991, The temperature of an asteroid

[8] Busch, M.W., et al, 2012, Shape and Spin of Near-Earth Asteroid 308635 (2005 YU55) From Radar Images and Speckle Tracking.

[9] Carry, B, 2012, Density of Asteroids

[10] Rein, H., Spiegel, D., 2015, IAS15: A fast, adaptive, high-order integrator for gravitational dynamics, accurate to machine precision over a billion orbits

5 Appendix

5.1 Code

Samples of code used to generate results and figures. All notebooks and outputs can be found at `https://git.io/vFP4b`.

REBOUND Model (1)

November 14, 2017

1 Begin work on 10/6/17

2 Imports

```
In [2]: import rebound
        import numpy as np
        import matplotlib.pyplot as plt
        import datetime
        from astropy.table import Table
        %matplotlib inline
```

3 Initialize Simulation

```
In [4]: sim = rebound.Simulation() #Create the simulation

        sim.add("Sun", date = "2017-11-10 15:30") #Add the various bodies into the solar system
        sim.add("Mercury", date = "2017-11-10 15:30")
        sim.add("Venus", date = "2017-11-10 15:30")
        sim.add("Earth", date = "2017-11-10 15:30")
        sim.add("Mars", date = "2017-11-10 15:30")
        sim.add("Jupiter", date = "2017-11-10 15:30")
        sim.add("308635", date = "2017-11-10 15:30")


        #fig = rebound.OrbitPlot(sim, slices=True, color=True,lim=2., limz=0.36, unitlabel="[AU]
```

```
Searching NASA Horizons for 'Sun'... Found: Sun (10).
Searching NASA Horizons for 'Mercury'... Found: Mercury Barycenter (199).
Searching NASA Horizons for 'Venus'... Found: Venus Barycenter (299).
Searching NASA Horizons for 'Earth'... Found: Earth-Moon Barycenter (3).
Searching NASA Horizons for 'Mars'... Found: Mars Barycenter (4).
Searching NASA Horizons for 'Jupiter'... Found: Jupiter Barycenter (5).
Searching NASA Horizons for '308635'... Found: 308635 (2005 YU55).
```

```
/Users/ke9885jd/anaconda3/lib/python3.6/site-packages/rebound/horizons.py:128: RuntimeWarning: W
  warnings.warn("Warning: Mass cannot be retrieved from NASA HORIZONS. Set to 0.", RuntimeWarnin
```

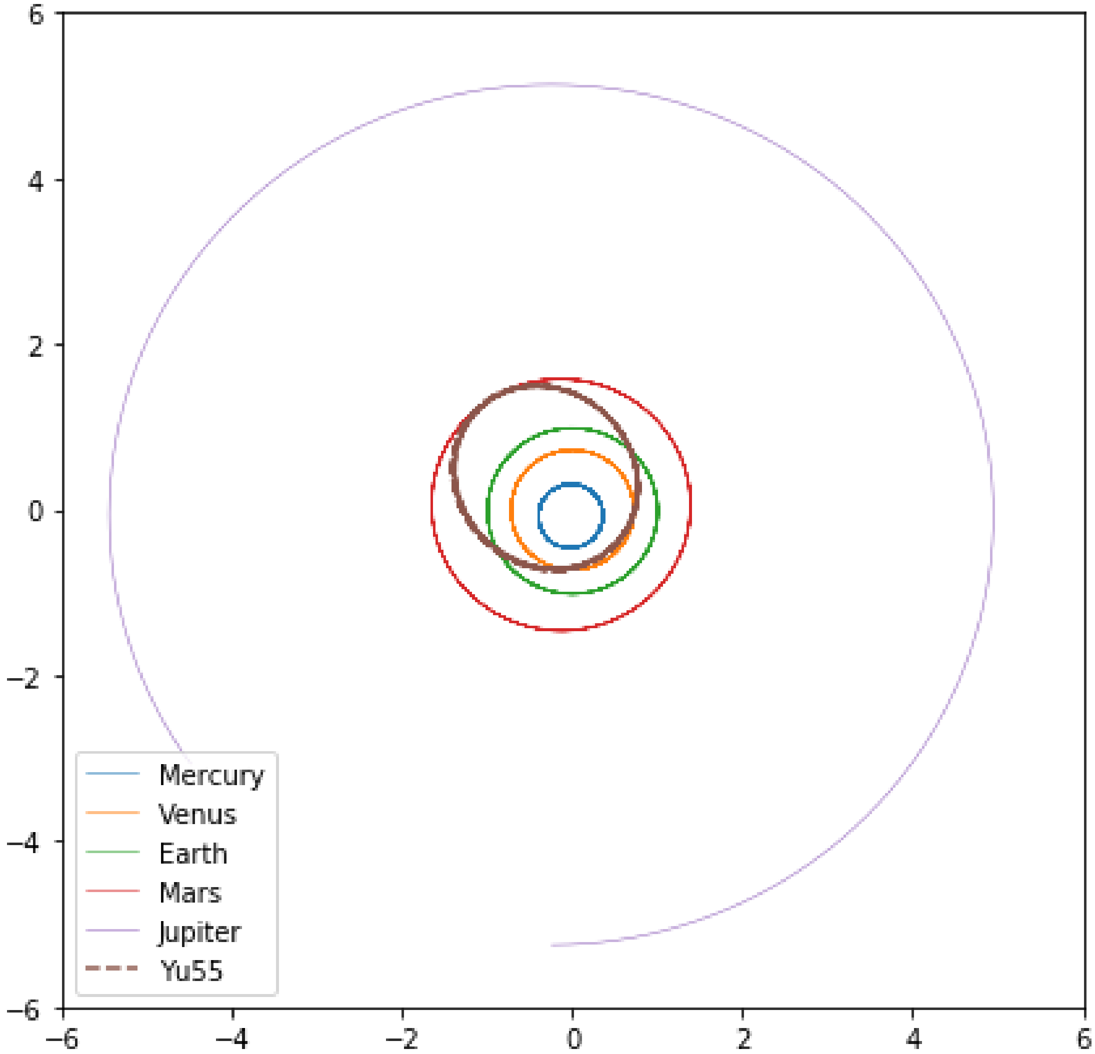
4 Integrate Solar System Backward in Time

```
In [5]: sim.dt = -0.00001 #Time step for integration
        Noutputs = 10000 # Store postions of planets and asteroids at 1000 times during the inte
        year = 2 * np.pi #If we choose G = 1 (which REBOUND does), then one year = 2Pi
        times = np.linspace(0, -10 * year, Noutputs) # An array of linearly spaced times
        x = np.zeros((6, Noutputs)) #Empty array to store x positions
        y = np.zeros((6, Noutputs)) #Empty array to store y positions
```

```
In [6]: sim.integrator = "ias15" #One of many fine integrators to choose from
        sim.move_to_com() #Move to COM frame
        ps = sim.particles #Array of pointer. Changes as simulations runs


        for i, time in enumerate(times): #Integrate and store positions in the originally create
            sim.integrate(time)
            x[0][i] = ps[1].x #Mercury x
            y[0][i] = ps[1].y #Mercury y
            x[1][i] = ps[2].x #Venus x
            y[1][i] = ps[2].y #Venus y
            x[2][i] = ps[3].x #Earth x
            y[2][i] = ps[3].y #Earth y
            x[3][i] = ps[4].x #Mars x
            y[3][i] = ps[4].y #Mars y
            x[4][i] = ps[5].x #Jupiter x
            y[4][i] = ps[5].y #Jupiter y
            x[5][i] = ps[6].x #Asteroid x
            y[5][i] = ps[6].y #Asteroidy y
```

```
In [10]: fig = plt.figure(figsize = (7, 7))  #Create a plot to show object orbits
         ax = plt.subplot(111)
         ax.set_xlim([-6,6])
         ax.set_ylim([-6,6])
         plt.plot(x[0], y[0], label='Mercury', linewidth = 0.5)
         plt.plot(x[1], y[1], label='Venus', linewidth = 0.5)
         plt.plot(x[2], y[2], label='Earth', linewidth = 0.5)
         plt.plot(x[3], y[3], label='Mars', linewidth = 0.5)
         plt.plot(x[4], y[4], label='Jupiter', linewidth = 0.5)
         plt.plot(x[5], y[5], '--', label='Yu55', linewidth = 1.5)
         plt.legend()
         plt.savefig('OrbitWithoutForcesThin.png')
         # The asteroid (brown orbit) interects the orbit of several planets. Plot closest appro
         #close it gets
```

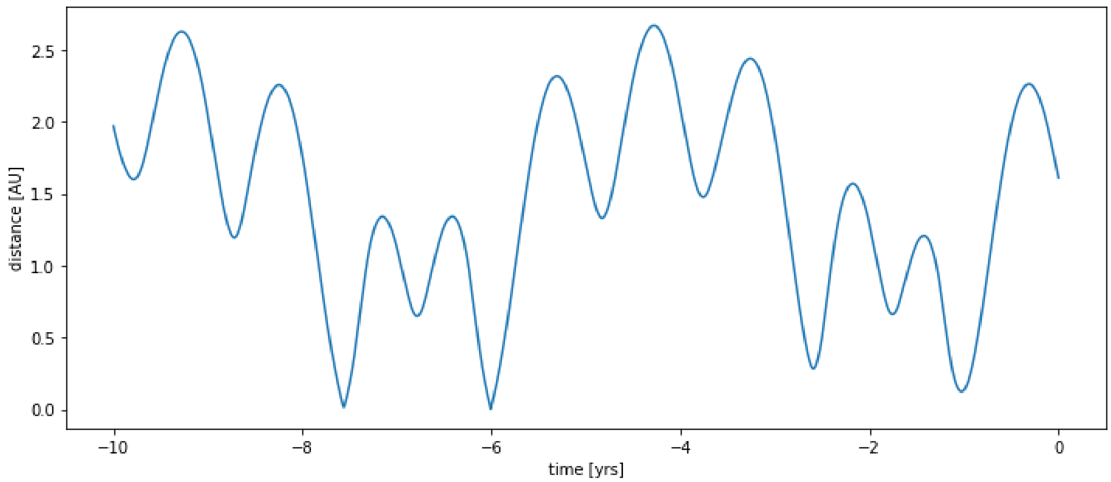


In [8]:
```
fig2 = plt.figure(figsize=(12,5))
ax2 = plt.subplot(111)
ax2.set_xlabel("time [yrs]")
ax2.set_ylabel("distance [AU]")
distance = np.sqrt(np.square(x[5]-x[2])+np.square(y[5]-y[2]))
plt.plot(times/year, distance);
closeencountertime = times[np.argmin(distance)]/year
print("Minimum distance (%f AU) occured at time: %f years." % (np.min(distance),closeenc

encounterdate = datetime.datetime(2017,11,10,15,30) + datetime.timedelta(days=365.25*clo
encounterdate.strftime("%Y-%m-%d %H:%M")
print(encounterdate)
plt.savefig('ClosestApproachWithoutForces.png')
```

```
# The closest approach does not appreciably change with solar raditaion accounted for.
```

```
Minimum distance (0.001966 AU) occured at time: -6.005601 years.
2011-11-09 02:24:19.765977
```




```
In [101]: xypostable = Table([x[5], y[5]], names=('x_pos', 'y_pos'))

In [103]: xypostable.write('xypostable.csv', overwrite=True)

In [ ]:
```

Position Subtraction

November 14, 2017

```
In [10]: import rebound
         import numpy as np
         import matplotlib.pyplot as plt
         import datetime
         from astropy.table import Table
         %matplotlib inline
```

```
In [11]: xypostable = Table.read('/Users/ke9885jd/Documents/Research_Code/Various CSV Files/xypo
```

```
In [12]: xypostable_forces = Table.read('/Users/ke9885jd/Documents/Research_Code/Various CSV Fil
```

1 Calculate difference in position between force and no force

1.1 X difference

```
In [13]: x_diff = xypostable['x_pos'] - xypostable_forces['x_pos']
```

1.2 Y diff

```
In [14]: y_diff = xypostable['y_pos'] - xypostable_forces['y_pos']
```

2 Difference Table

```
In [15]: diff_in_pos = Table([x_diff, y_diff], names=('Diff_in_xpos', 'Diff_in_ypos'))
```

```
In [17]: diff_in_pos
```

```
Out[17]: <Table length=10000>
         Diff_in_xpos Diff_in_ypos
           float64      float64
         ------------ ------------
                  0.0          0.0
                  0.0          0.0
                  0.0          0.0
                  0.0          0.0
                  0.0          0.0
                  0.0          0.0
                  0.0          0.0
```

```
            0.0         0.0
            0.0         0.0
            0.0         0.0
            ...         ...
            0.0         0.0
            0.0         0.0
            0.0         0.0
            0.0         0.0
            0.0         0.0
            0.0         0.0
            0.0         0.0
            0.0         0.0
            0.0         0.0
            0.0         0.0
```

```
In [18]: np.max(diff_in_pos['Diff_in_xpos']), np.min(diff_in_pos['Diff_in_xpos'])
```

```
Out[18]: (0.0, 0.0)
```

```
In [19]: np.max(diff_in_pos['Diff_in_ypos']), np.min(diff_in_pos['Diff_in_ypos'])
```

```
Out[19]: (0.0, 0.0)
```

```
In [12]: diff_in_pos.write("diff_in_pos.csv")
```

```
In [ ]:
```

```
In [ ]:
```