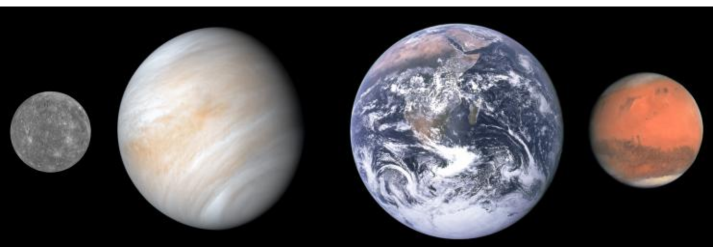
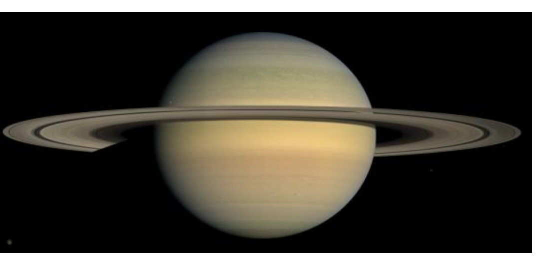
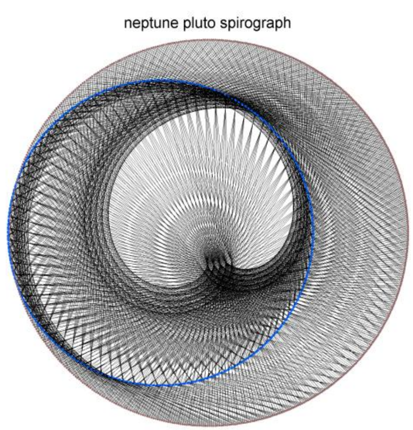
BPhO Computational Challenge

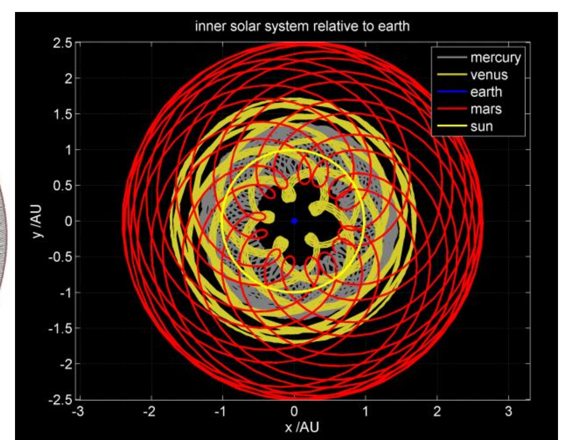


2023

Solar System Orbits







BPhO British Physics Olympiad

v1 Aug 2022

Instructions: Welcome to the **British Physics Olympiad Computational Challenge 2023.** The goal is to build computer models based upon the instructions in this document. Much can be achieved using a *spreadsheet* such as Microsoft Excel, although you are encouraged to use a *programming language* of your choice* for the more sophisticated models and graphical visualizations.

The challenge runs from **Easter 2023 till August 2023.** To submit an entry you will need to fill in a web form. There may be a small administration charge of, payable online as per other BPhO competition entries.

The deliverable of the challenge is to produce a **screencast** of ***maximum length two minutes*** which describes your response to the challenge, i.e. the graphs and the code & spreadsheets and your explanation of these. The videos must be uploaded to **YouTube**, and we recommend you set these as *Unlisted* with *Comments disabled*. **Your entry will comprise a YouTube link.** *Instructions how to do this are at the end of this presentation.* To produce the screencast, we recommend the Google Chrome add-on [Screencastify.]

You can enter the challenge **individually** or in **pairs**. If you opt for the latter, *both* of you must make equal contributions to the screencast.

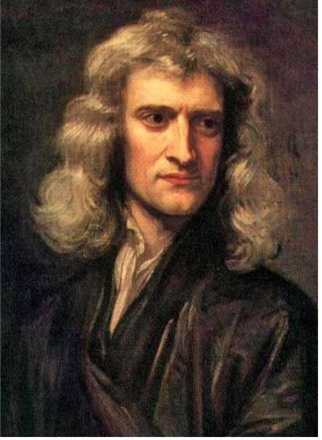
Gold, Silver or **Bronze** e-certificates will be emailed to each complete entry, and the **top five** Golds will be invited to present their work at a special ceremony. You should receive a result by December 2023. Note no additional feedback will be provided, and the decision of the judges is final.

Bronze: Initial spreadsheet-based challenge elements completed, some basic coding.
Silver: All the spreadsheet-based elements completed, and a commendable attempt at the programming-based elements. Most tasks completed to a reasonable standard.
Gold: All tasks completed to a high standard, with possible extension work such as the construction of apps (i.e. programs with graphical user interfaces), significant development of the models, attempt at extension work, short research papers etc.

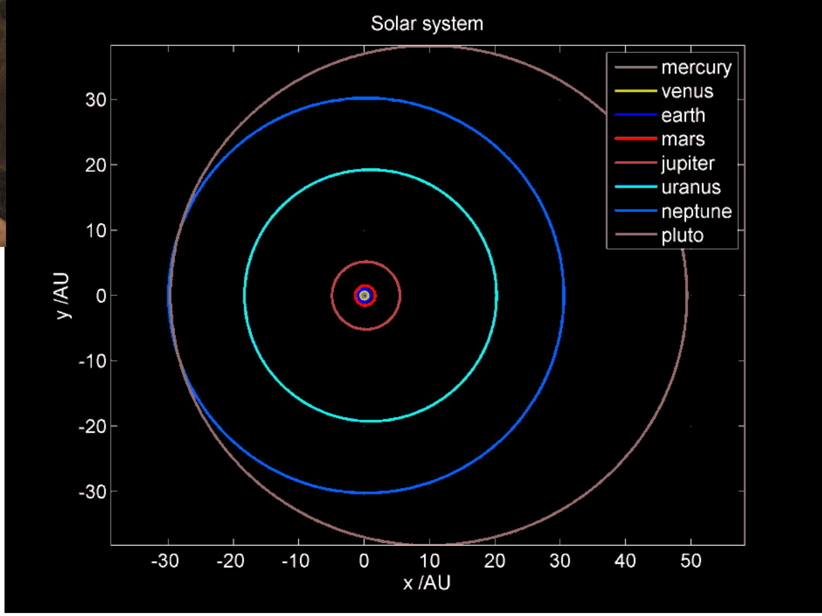
***MATLAB** or **Python** is recommended, although any system that can easily execute code in loops and plot graphs will do. e.g. **Octave, Java, Javascript, C#, C++...** Use what you can access and feel comfortable with. [Programming resources]

How to make a screencast using Screencastify and upload this to Youtube

1. Download the Google Chrome web browser
2. Download the Screencastify add-on to Chrome. The free educational version will allow up to 5 minutes of video.
3. When you are ready to make your video (have all the program windows open in advance, and prepare what you are going to say), click on the Screencastify arrow in the corner of your browser. Follow the instructions to record a screen, and a three second countdown will begin.
4. Record your video!
5. Export your video to a **.webm** or **.mp4** file. There is also a direct to YouTube upload option.
6. Upload your video to YouTube (you will need to set up an account first and establish a Channel).
7. Navigate to your video and copy to the clipboard the YouTube weblink. Submit this link in your submission form in the BPhO website.
8. It is recommended that (i) you *don't* have a presenter image in your video (you can turn off this in Screencastify) , i.e. **only have a voice-over**. Also *turn off Comments* in YouTube and make the video *Unlisted.* This means nobody can leave comments, and only those with the link will find your video.



Isaac Newton (1642-1727) developed a mathematical model of Gravity which predicted the ***elliptical*** orbits proposed by Johannes Kepler (1571-1630)



Planet and star masses

Force of gravity

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

Universal gravitational constant

$$G = 6.67 \times 10^{-11}\,\mathrm{m^3 kg^{-1} s^{-2}}$$

$$r = \frac{a\left(1-\varepsilon^2\right)}{1-\varepsilon\cos\theta}$$

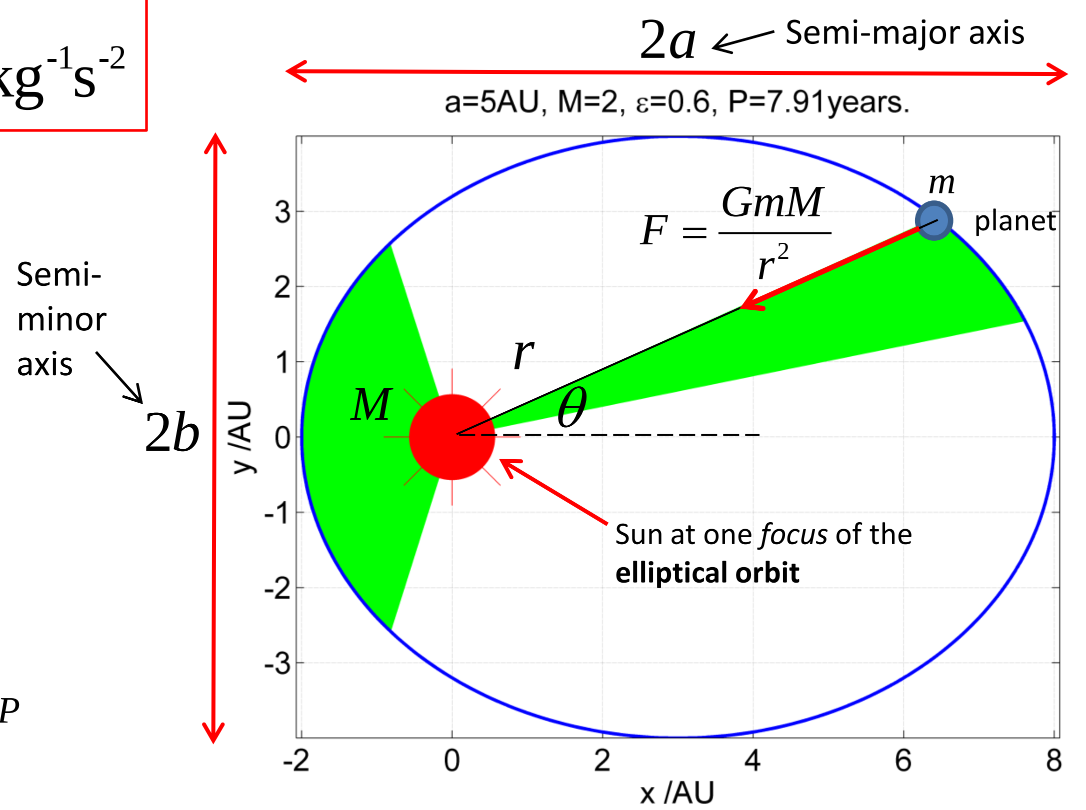
Polar equation of ellipse

$$\varepsilon = \sqrt{1-\frac{b^2}{a^2}}$$

Eccentricity of ellipse

$$P^2 = \frac{4\pi^2}{G(m+M)}a^3$$

Orbital period P



Kepler's three laws are: $G = 6.67 \times 10^{-11}\,\mathrm{m^3 kg^{-1} s^{-2}}$ $\quad M_\odot = 1.9891 \times 10^{30}\,\mathrm{kg}$

1. *The orbit of every planet in the solar system is an ellipse with the Sun at one of the two foci.*
2. *A line joining a planet and the Sun sweeps out equal areas during equal intervals of time.*
3. *The square of the orbital period of a planet is directly proportional to the cube of the semi-major axis of its orbit.*

The wording of Kepler's laws implies a specific application to the solar system. However, the laws are more generally applicable to any system of two masses whose mutual attraction is an inverse-square law.

$$\boxed{\frac{dA}{dt} = \tfrac{1}{2}\sqrt{G(m+M)\left(1-\varepsilon^2\right)a}}$$

This is a *constant* of the orbit

$$r = \frac{a\left(1-\varepsilon^2\right)}{1-\varepsilon\cos\theta}$$

Polar equation of ellipse

$$\varepsilon = \sqrt{1-\frac{b^2}{a^2}}$$

Eccentricity of ellipse

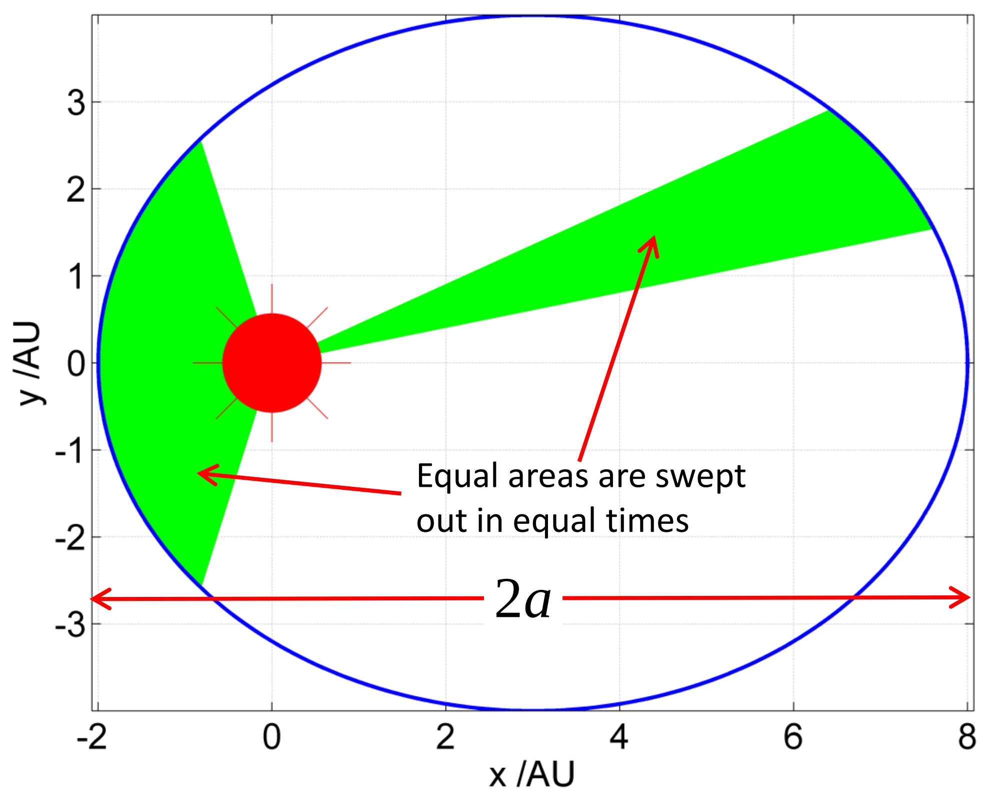
$$P^2 = \frac{4\pi^2}{G(m+M)}a^3$$

Orbital period *P*

Planet mass

Sun mass

a=5AU, M=2, ε=0.6, P=7.91years.



Johannes Kepler
1571-1630

Object	Mass in Earth masses	Distance from Sun in AU	Radius in Earth radii	Rotational period /days	Orbital period /years	Gravitational field (in terms of g = 9.81 ms^-2)
Saturn	95.16	9.58	9.45	0.44	29.63	1.07
Uranus	14.50	19.29	4.01	0.72	84.75	0.90
Jupiter	317.85	5.20	11.21	0.41	11.86	2.53
Sun	332,837	-	109.12	-	-	27.95
Neptune	17.20	30.25	3.88	0.67	166.34	1.14
Pluto	0.00	39.51	0.19	6.39	248.35	0.09
Mars	0.107	1.523	0.53	1.03	1.88	0.38
Venus	0.815	0.723	0.95	243.02	0.62	0.90
Mercury	0.055	0.387	0.38	58.65	0.24	0.37
Earth	1.000	1.000	1.00	1.00	1.00	1.00

For *our* Solar System:

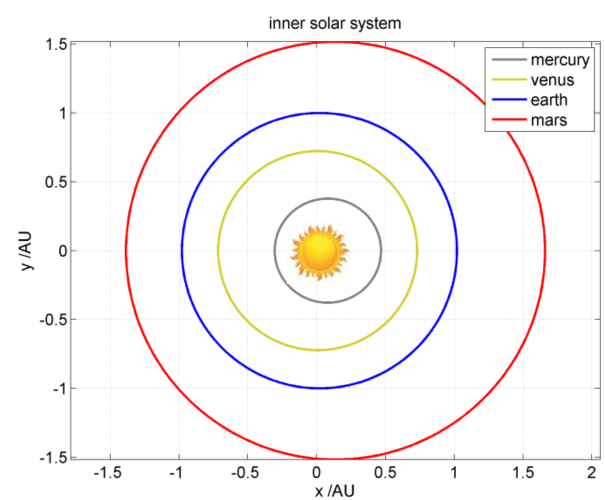
$$m \ll M_\odot$$

$$P^2 = \frac{4\pi^2}{G\left(m + M_\odot\right)} a^3$$

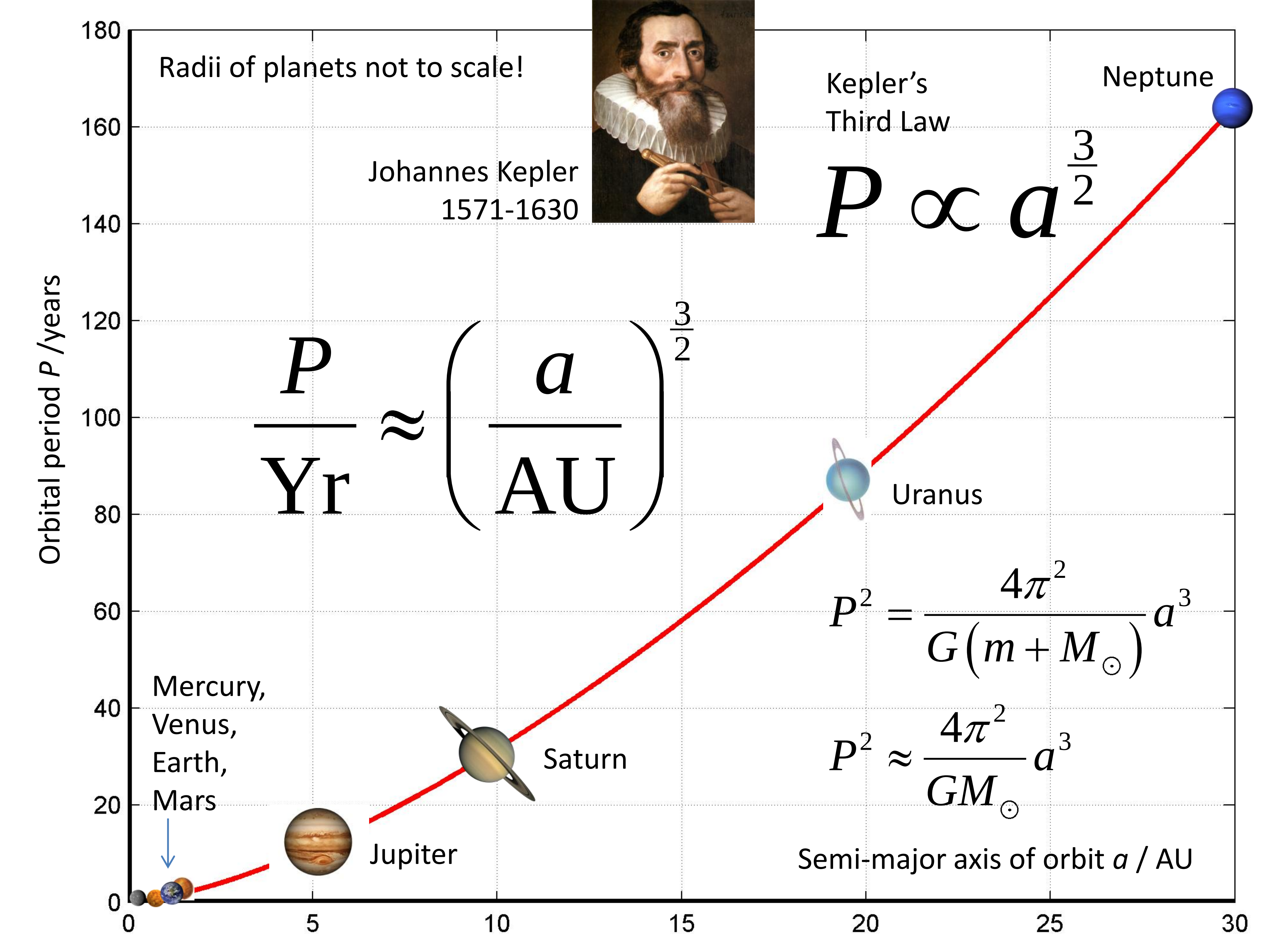
$$P^2 \approx \frac{4\pi^2}{GM_\odot} a^3$$

$$\mathrm{Yr}^2 = \frac{4\pi^2}{GM_\odot} \mathrm{AU}^3$$

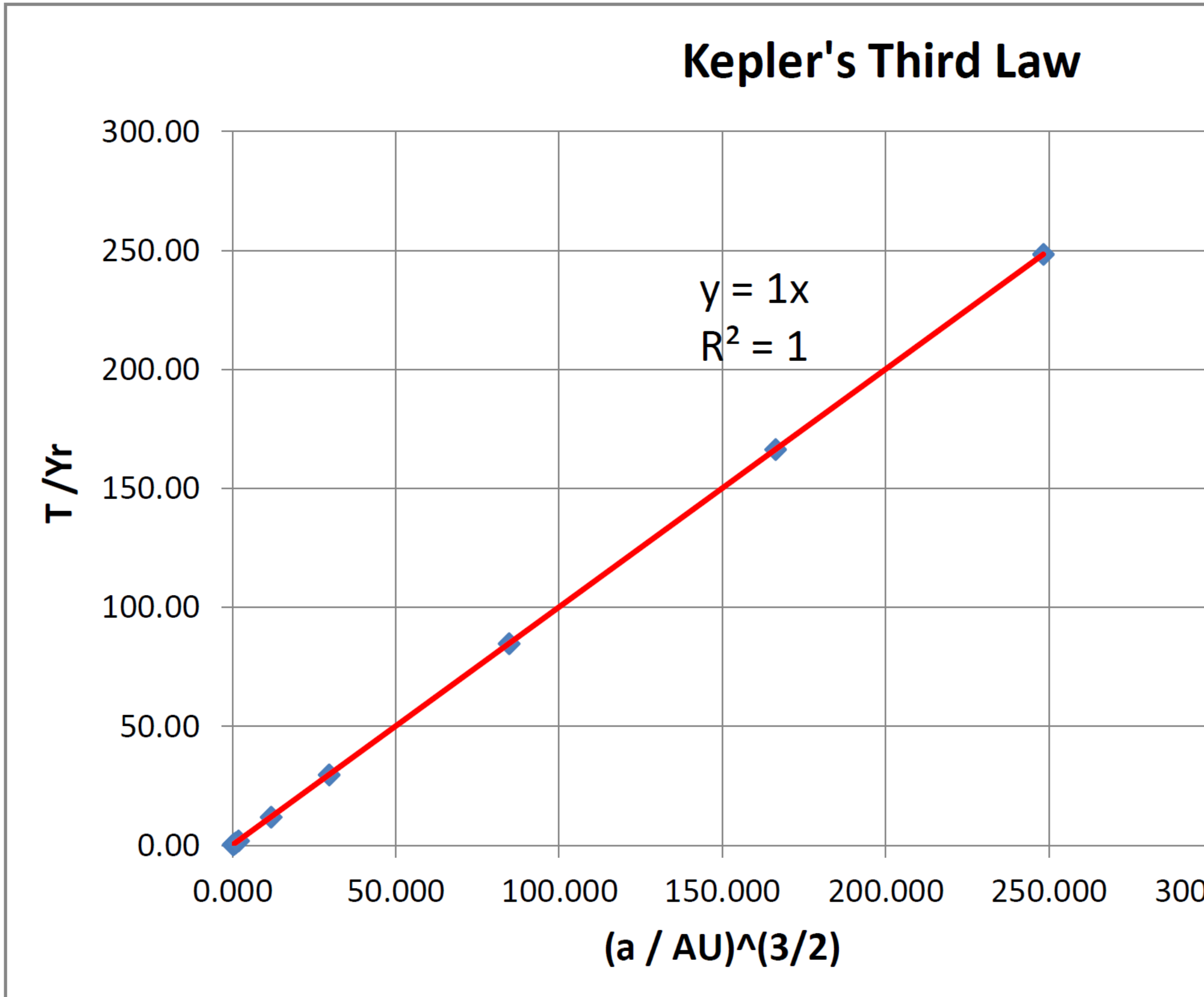
$$\boxed{\therefore \frac{P}{\mathrm{Yr}} \approx \left(\frac{a}{\mathrm{AU}}\right)^{\frac{3}{2}}}$$



$$1\mathrm{AU} = 1.496 \times 10^{11}\,\mathrm{m}$$



Radii of planets not to scale!
Johannes Kepler
1571-1630
Kepler's
Third Law
$P \propto a^{\frac{3}{2}}$
Neptune
$\frac{P}{\mathrm{Yr}} \approx \left(\frac{a}{\mathrm{AU}}\right)^{\frac{3}{2}}$
Uranus
$P^2 = \frac{4\pi^2}{G\left(m + M_{\odot}\right)} a^3$
$P^2 \approx \frac{4\pi^2}{GM_{\odot}} a^3$
Mercury,
Venus,
Earth,
Mars
Jupiter
Saturn
Semi-major axis of orbit a / AU
Orbital period P /years
0
20
40
60
80
100
120
140
160
180
0
5
10
15
20
25
30



$$\frac{T}{\mathrm{Yr}} = \left(\frac{R}{\mathrm{AU}}\right)^{\frac{3}{2}}$$

A *very strong* correlation of Kepler III to orbital data for planets in our solar system!

Challenge #1: Replicate this Kepler III correlation in Excel or Python or MATLAB

Challenge #2: Plot elliptical orbits of the planets using Excel, Python, MATLAB etc....

Assume all orbits are **ellipses** with the Sun at the (left) focus. Let this sun position be the origin of a Cartesian coordinate system, and assume the sun is stationary.

$$r = \frac{a\left(1-\varepsilon^2\right)}{1-\varepsilon\cos\theta}$$

$$\varepsilon = \sqrt{1-\frac{b^2}{a^2}} \quad \therefore b = a\left(1-\varepsilon^2\right)$$

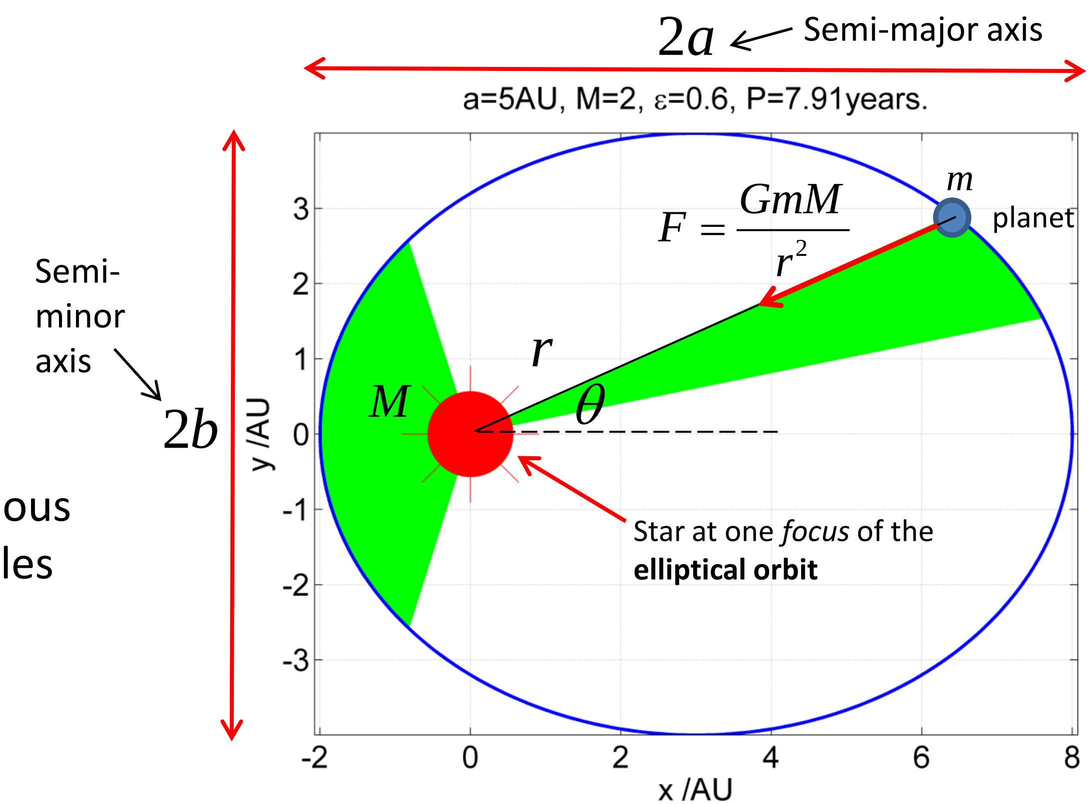
$$P^2 = \frac{4\pi^2}{G(m+M)}a^3$$

$$x = r\cos\theta, \;\; y = r\sin\theta$$

$$\theta = 0....2\pi$$

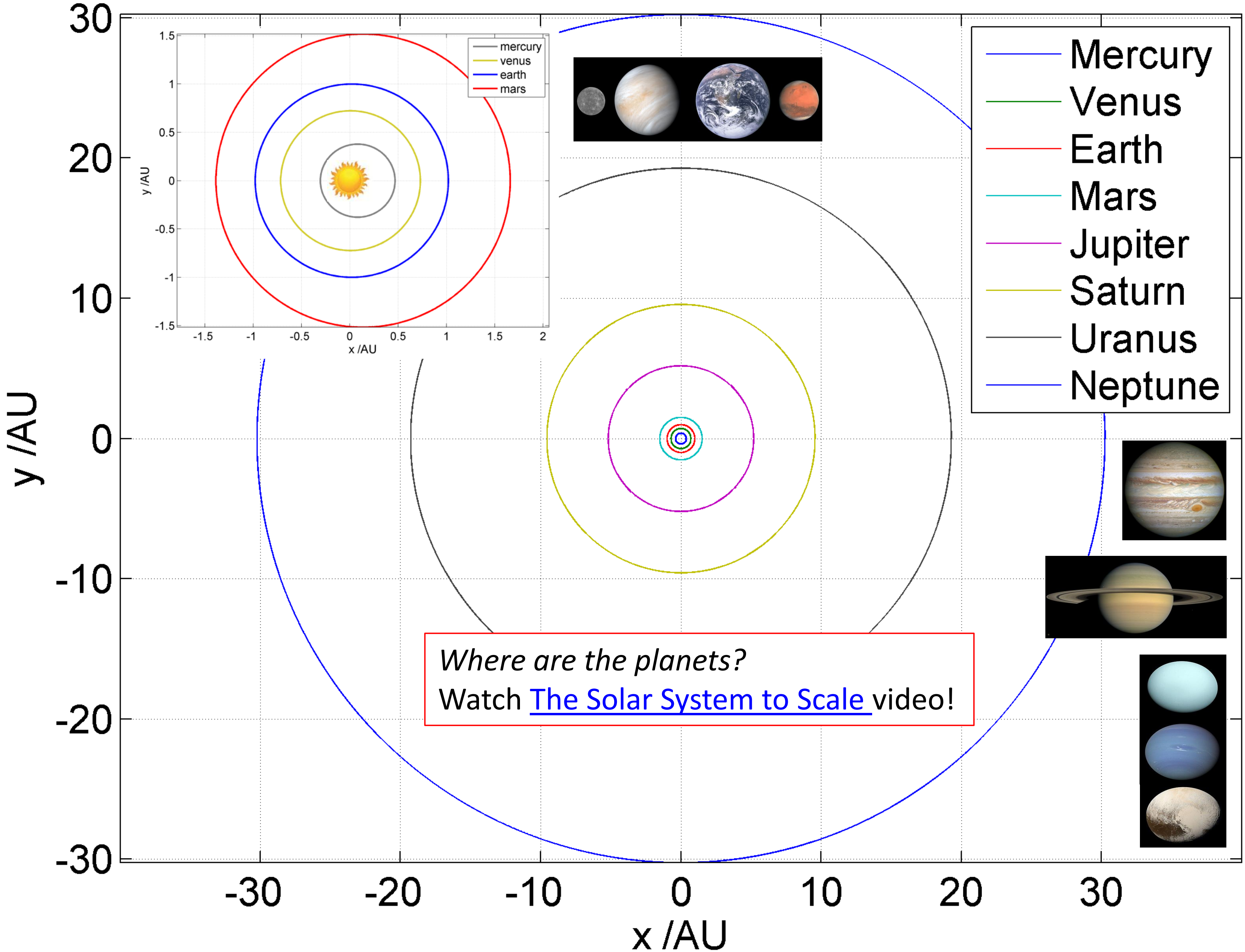
Plot the inner five planets on a separate scale to the outer planets

We will assume at this point all elliptical orbits are in the same plane ... but this is not quite true!



Use the data in the table on the previous slide. Use a 1,000 linearly spaced angles θ for each orbit.

Use an axis scale of AU



mercury
venus
earth
mars
y /AU
x /AU
Mercury
Venus
Earth
Mars
Jupiter
Saturn
Uranus
Neptune
y /AU
x /AU
Where are the planets?
Watch [The Solar System to Scale](#) video!

Challenge #3: Create a 2D animation of the solar system orbits

Use an axis scale of AU

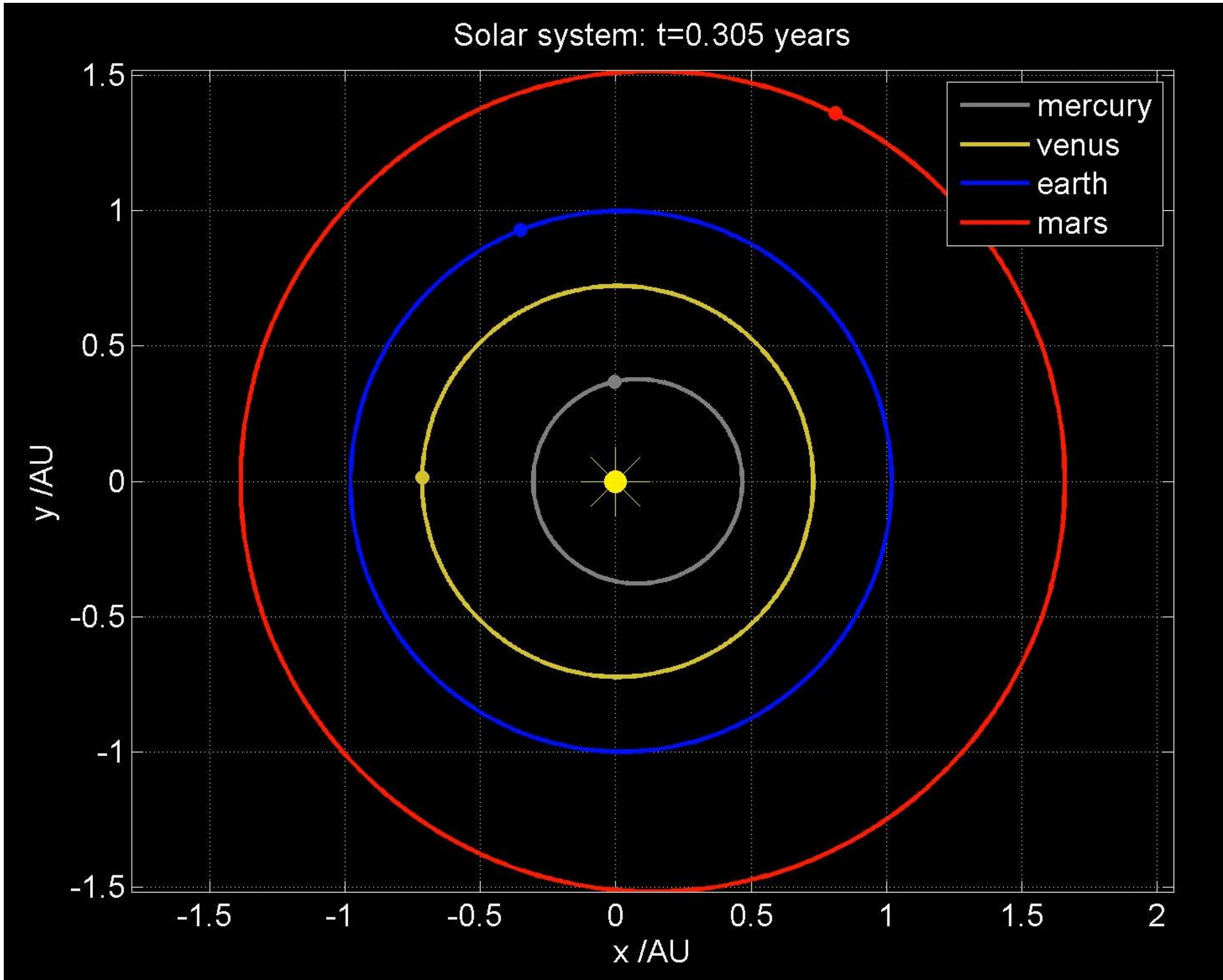
Plot the inner five planets on a separate scale to the outer planets

For the *inner* planets, set a frame rate such that one orbit of the Earth takes a second i.e. **one year is one second.** For the *outer* planets, **set the orbit of *Jupiter* to take one second**.

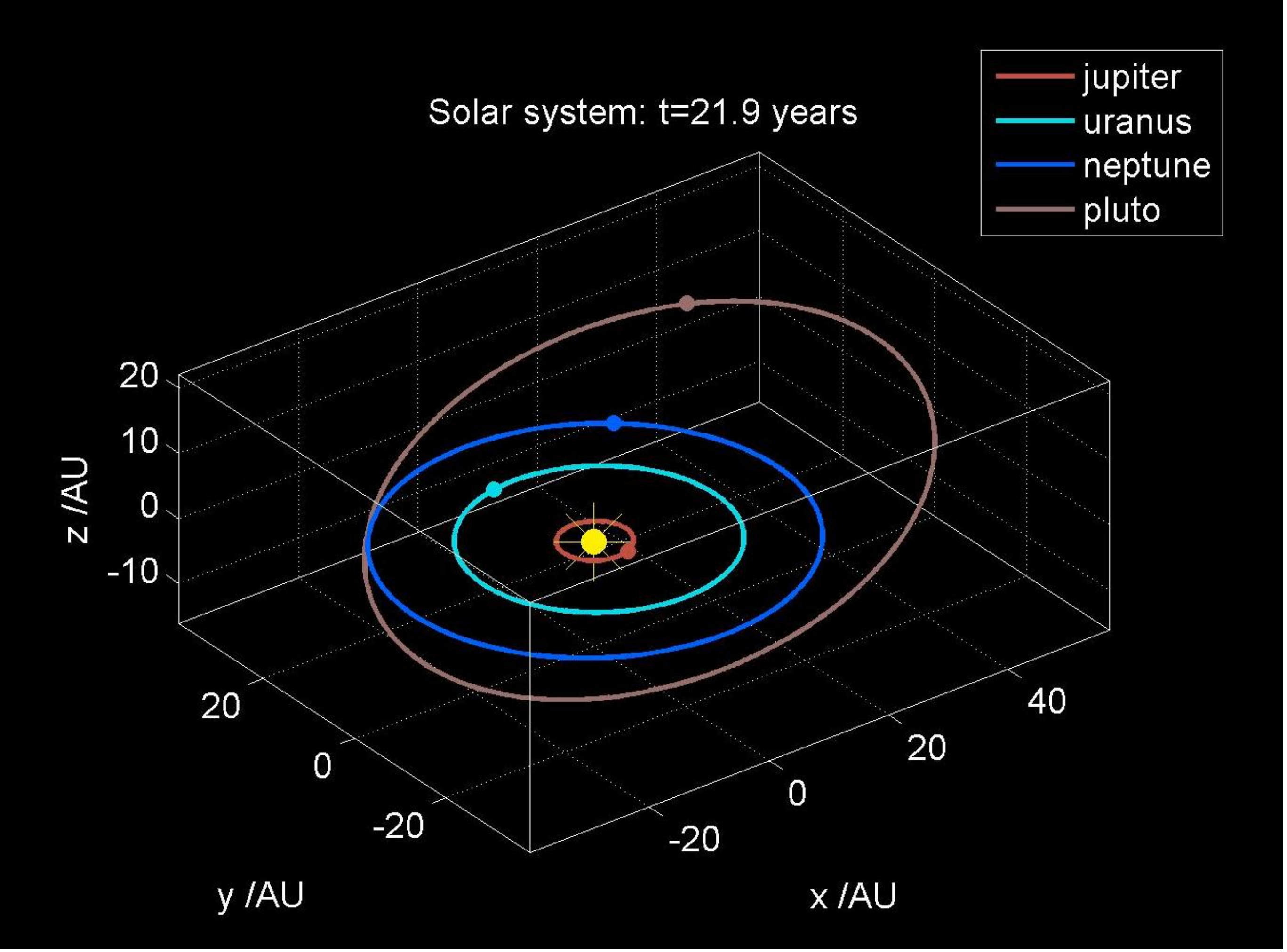
$$x = r\cos\theta, \;\; y = r\sin\theta$$

$$\theta = \frac{2\pi t}{P}$$

Run the simulation till the outermost planet completes at least one orbit.



[YouTube example video]



Challenge #4: Create a 3D animation of the solar system orbits

YouTube example video

Use the elliptical inclination angle β. *(See next slide).* Most orbits won't change much, but Pluto is the exception! The coordinate change is:

$$\boxed{x' = x\cos\beta \qquad z' = x\sin\beta \qquad y' = y}$$

β is the orbital inclination /degrees. In all cases the semi-major axis pointing direction is

$$\mathbf{d} = d_x\hat{\mathbf{x}} + d_y\hat{\mathbf{y}} + d_{\mathbf{y}}\hat{\mathbf{z}} = \cos\beta\hat{\mathbf{x}} + \sin\beta\hat{\mathbf{z}}$$

Object	$M/M_\oplus$	a /AU	ε	θ_0	β	$R/R_\oplus$	T_{rot} / days	P /Yr
Sun	332,837	-	-	-	-	109.123	-	-
Mercury	0.055	0.387	0.21	*	7.00	0.383	58.646	0.241
Venus†	0.815	0.723	0.01	*	3.39	0.949	243.018	0.615
Earth	1.000	1.000	0.02	*	0.00	1.000	0.997	1.000
Mars	0.107	1.523	0.09	*	1.85	0.533	1.026	1.881
Jupiter	317.85	5.202	0.05	*	1.31	11.209	0.413	11.861
Saturn	95.159	9.576	0.06	*	2.49	9.449	0.444	29.628
Uranus†	14.500	19.293	0.05	*	0.77	4.007	0.718	84.747
Neptune	17.204	30.246	0.01	*	1.77	3.883	0.671	166.344
Pluto†	0.003	39.509	0.25	*	17.5	0.187	6.387	248.348

$$r = \frac{a\left(1-\varepsilon^2\right)}{1-\varepsilon\cos\theta}$$

$$\varepsilon = \sqrt{1-\frac{b^2}{a^2}}$$

$$P^2 = \frac{4\pi^2}{G(m+M_\odot)}a^3$$

β is the orbital inclination /degrees. In all cases the semi-major axis pointing direction is

$$\mathbf{d} = d_x\hat{\mathbf{x}} + d_y\hat{\mathbf{y}} + d_{\mathbf{y}}\hat{\mathbf{z}} = \cos\beta\hat{\mathbf{x}} + \sin\beta\hat{\mathbf{z}}$$

You could begin with all zero, or perhaps a random angle for each planet's orbit.

$$\begin{aligned}
M_\odot &= 1.9891\times10^{30}\ \text{kg}\\
R_\odot &= 6.960\times10^{8}\ \text{m}\\
M_\oplus &= 5.9742\times10^{24}\ \text{kg}\\
R_\oplus &= 6.37814\times10^{6}\ \text{m}\\
1\text{AU} &= 1.495979\times10^{11}\ \text{m}
\end{aligned}$$

* For the current orbital polar angle θ_0 (and indeed more accurate values for solar system parameters) see the website of the Jet Propulsion Laboratory (JPL) `http://ssd.jpl.nasa.gov/`

†These planets rotate clockwise about their own internal polar axis. ("Retrograde"). All the other planets rotate anti-clockwise about their own internal axis. All the planets orbit the sun in an anticlockwise direction.

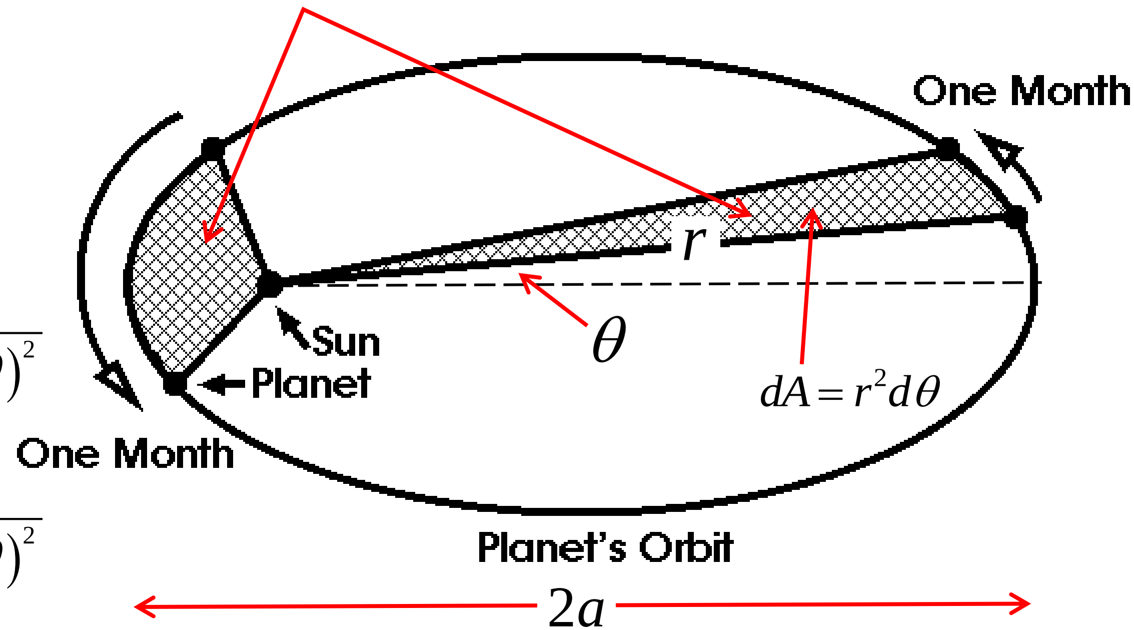
Calculating orbit angle vs time

$$\boxed{\frac{dA}{dt}=\tfrac{1}{2}\sqrt{G(m+M)\left(1-\varepsilon^2\right)a}}$$

This is a *constant*

Orbit time can be determined from polar angle using Kepler II:

Equal areas swept out in equal times



$$r^2\frac{d\theta}{dt}=\sqrt{G(m+M)\left(1-\varepsilon^2\right)a}$$

$$\therefore \int_{\theta_0}^{\theta} r^2 d\theta = t\sqrt{G(m+M)\left(1-\varepsilon^2\right)a}$$

$$\therefore t=\frac{a^2\left(1-\varepsilon^2\right)^2}{\sqrt{G(m+M)\left(1-\varepsilon^2\right)a}}\int_{\theta_0}^{\theta}\frac{d\theta}{\left(1-\varepsilon\cos\theta\right)^2}$$

$$\therefore t=\frac{a^2\left(1-\varepsilon^2\right)^2}{\sqrt{G(m+M)\left(1-\varepsilon^2\right)a}}\int_{\theta_0}^{\theta}\frac{d\theta}{\left(1-\varepsilon\cos\theta\right)^2}$$

$$\therefore t=\sqrt{\frac{a^3\left(1-\varepsilon^2\right)^3}{G(m+M)}}\int_{\theta_0}^{\theta}\frac{d\theta}{\left(1-\varepsilon\cos\theta\right)^2}$$

From Kepler III: $P^2=\dfrac{4\pi^2}{G(m+M)}a^3$

$$\boxed{t=P\left(1-\varepsilon^2\right)^{\frac{3}{2}}\frac{1}{2\pi}\int_{\theta_0}^{\theta}\frac{d\theta}{\left(1-\varepsilon\cos\theta\right)^2}}$$

Evaluate this *numerically*

Note when: $\boxed{\begin{array}{l}\varepsilon \ll 1\\ t\approx P\left(\theta-\theta_0\right)\end{array}}$

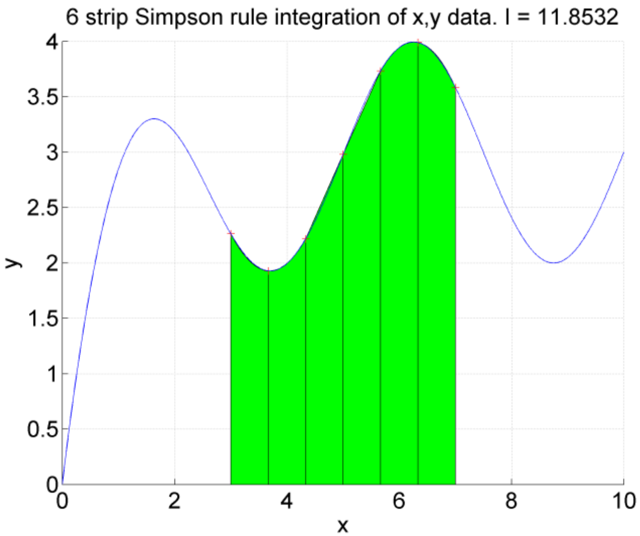
Challenge #5: Calculate orbit angle vs time for an *eccentric* orbit (e.g. pluto) and compare to a circular version with the same period.

To evaluate the angle integral, use **Simpson's rule**, which approximates the integrand of an integral with a series of quadratic curve segments.

$$t = P\left(1-\varepsilon^2\right)^{\frac{3}{2}} \frac{1}{2\pi}\int_{\theta_0}^{\theta} \frac{d\theta}{\left(1-\varepsilon\cos\theta\right)^2}$$

$$\varepsilon \ll 1$$

$$t \approx P\left(\theta-\theta_0\right)$$



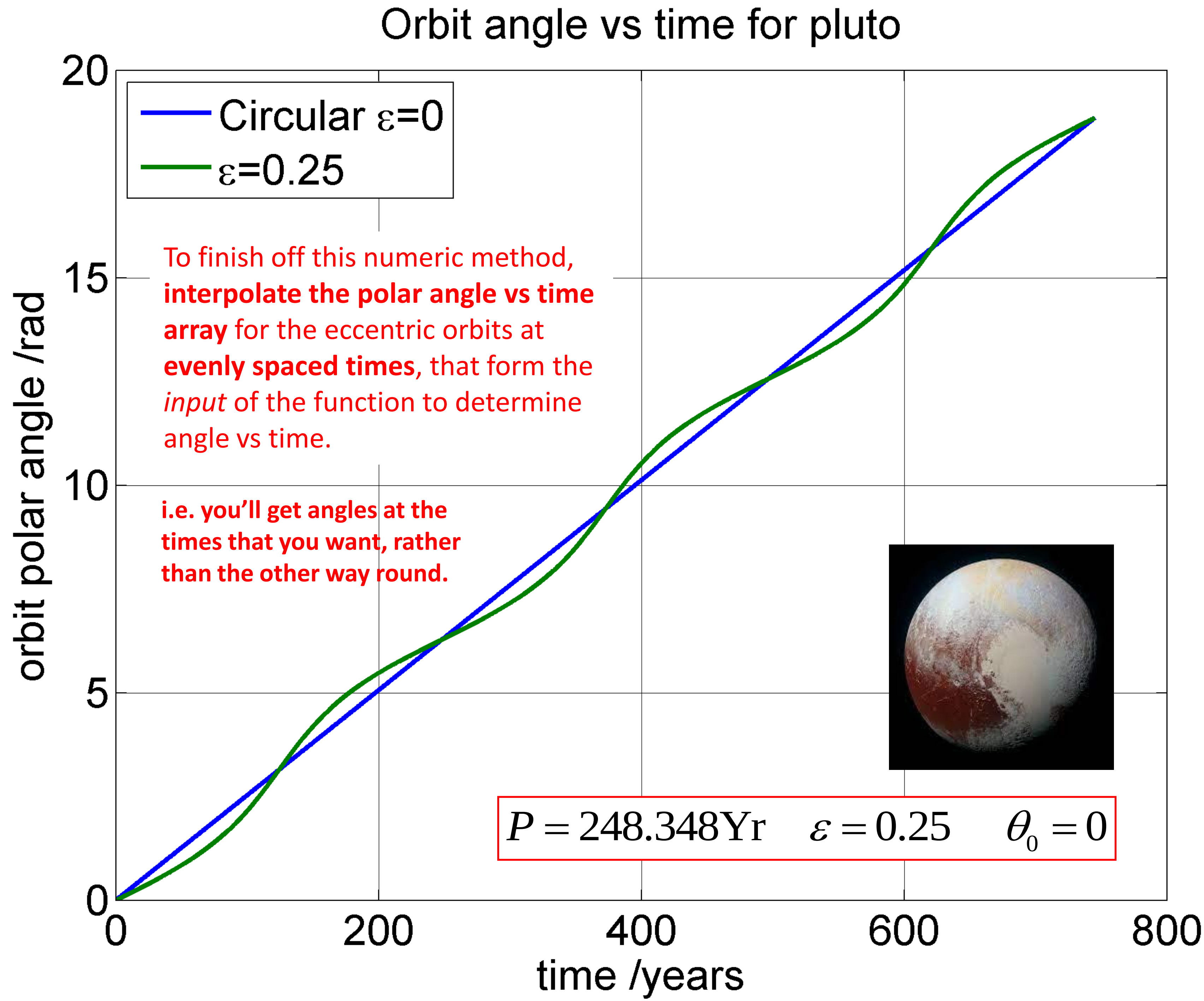
$$\int_a^b f(x)dx \approx \tfrac{1}{3}h\left\{y_0 + 4y_1 + 2y_2 + 4y_3 + 2y_4 + \ldots 4y_{N-1} + y_N\right\}$$

$y_n = f(a+nh)$ $\quad$ h is the strip width $h = \dfrac{b-a}{N}$

Determine time vs angle for three periods of Pluto's orbit, using $d\theta = h =$ 1/1000.

You'll have to evaluate the integral over a *range* of polar angles, which amounts to a **cumulative sum**. Many languages have functions (such as `cumsum` in MATLAB) that can perform efficient operations with *arrays*.

$$P = 248.348\mathrm{Yr} \quad \varepsilon = 0.25 \quad \theta_0 = 0$$

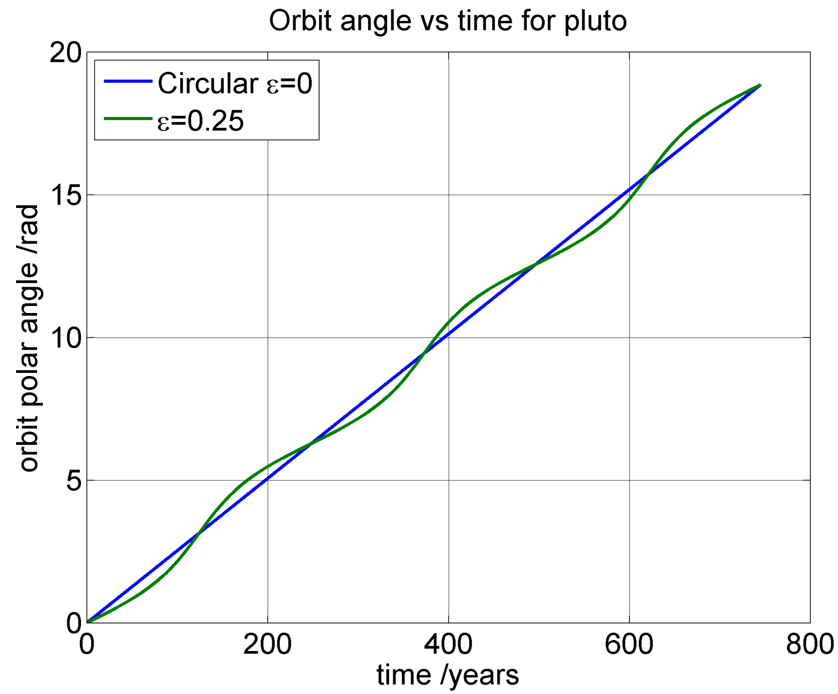


Orbit angle vs time for pluto
Circular ε=0
ε=0.25
To finish off this numeric method, **interpolate the polar angle vs time array** for the eccentric orbits at **evenly spaced times**, that form the *input* of the function to determine angle vs time.
i.e. you'll get angles at the times that you want, rather than the other way round.
$P = 248.348\text{Yr} \quad \varepsilon = 0.25 \quad \theta_0 = 0$
orbit polar angle /rad
time /years
0
5
10
15
20
0
200
400
600
800

MATLAB example code for determining polar angle vs time for an elliptical orbit

```
%Numeric method to compute polar angle vs orbit time
function theta = angle_vs_time( t, P, ecc, theta0 )

%Angle step for Simpson's rule
dtheta = 1/1000;

%

%Number of orbits
N = ceil( t(end)/P );

%Define array of polar angles for orbits
theta = theta0 : dtheta : ( 2*pi*N + theta0 );

%Evaluate integrand of time integral
f = (1 - ecc*cos(theta) ).^(-2);

%Define Simpson rule coefficients  c = [ 1, 4, 2, 4, 2, 4, ....
L = length(theta);
isodd = rem( 1:(L-2),2 ); isodd( isodd==1 ) = 4; isodd( isodd==0 ) = 2;
c = [1, isodd, 1];

%Calculate array of times
tt = P*( (1-ecc^2)^(3/2) )*(1/(2*pi))*dtheta*(1/3).*cumsum( c.*f );

%Interpolate the polar angles for the eccentric orbit at the circular orbit
%times
theta = interp1( tt, theta, t, 'spline' );
```

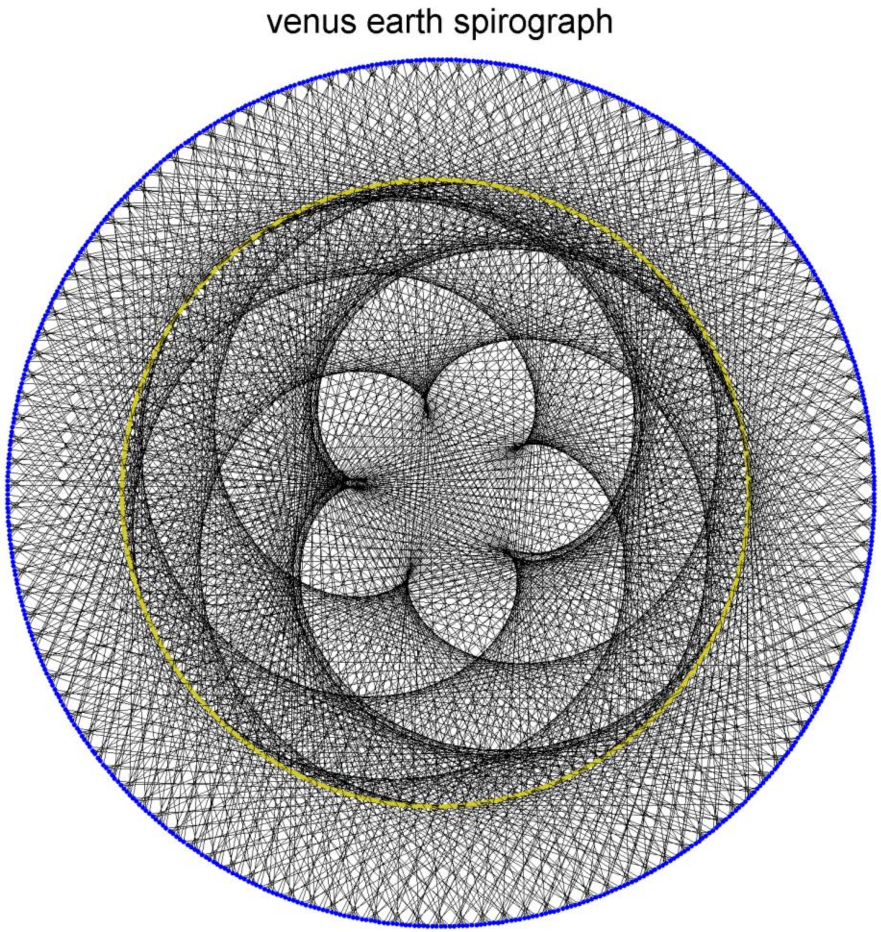
Note time is an input

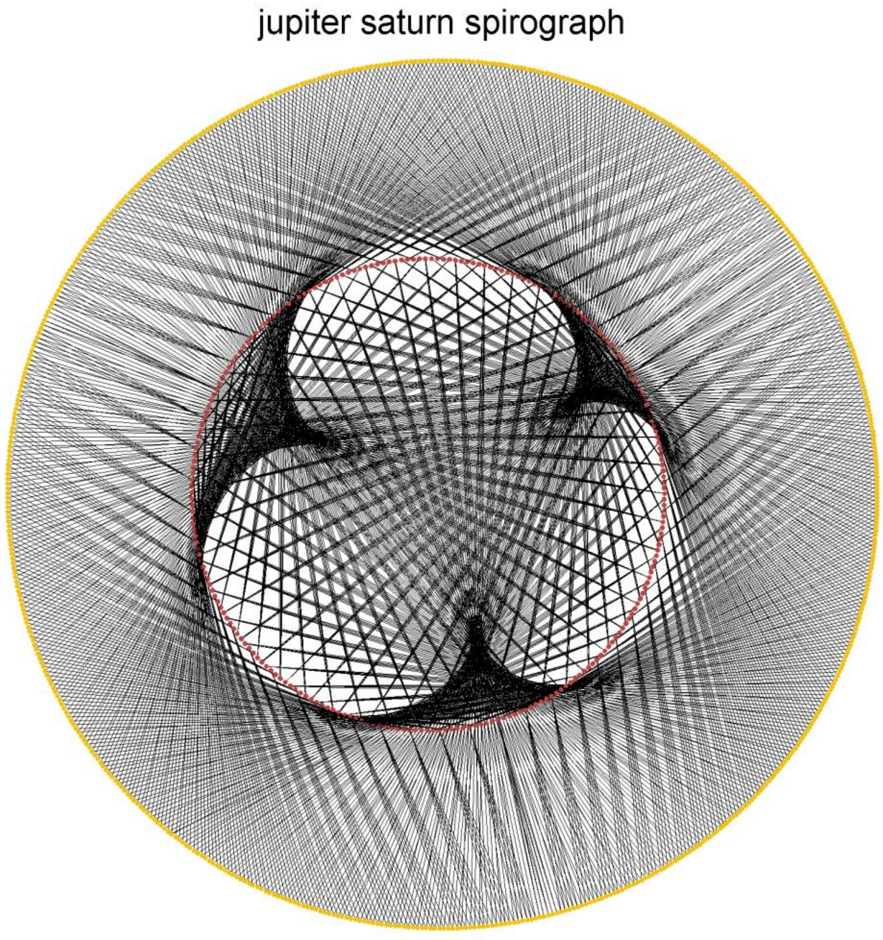


Challenge #6: Solar system spirograph! inspired by: https://engaging-data.com/planetary-spirograph

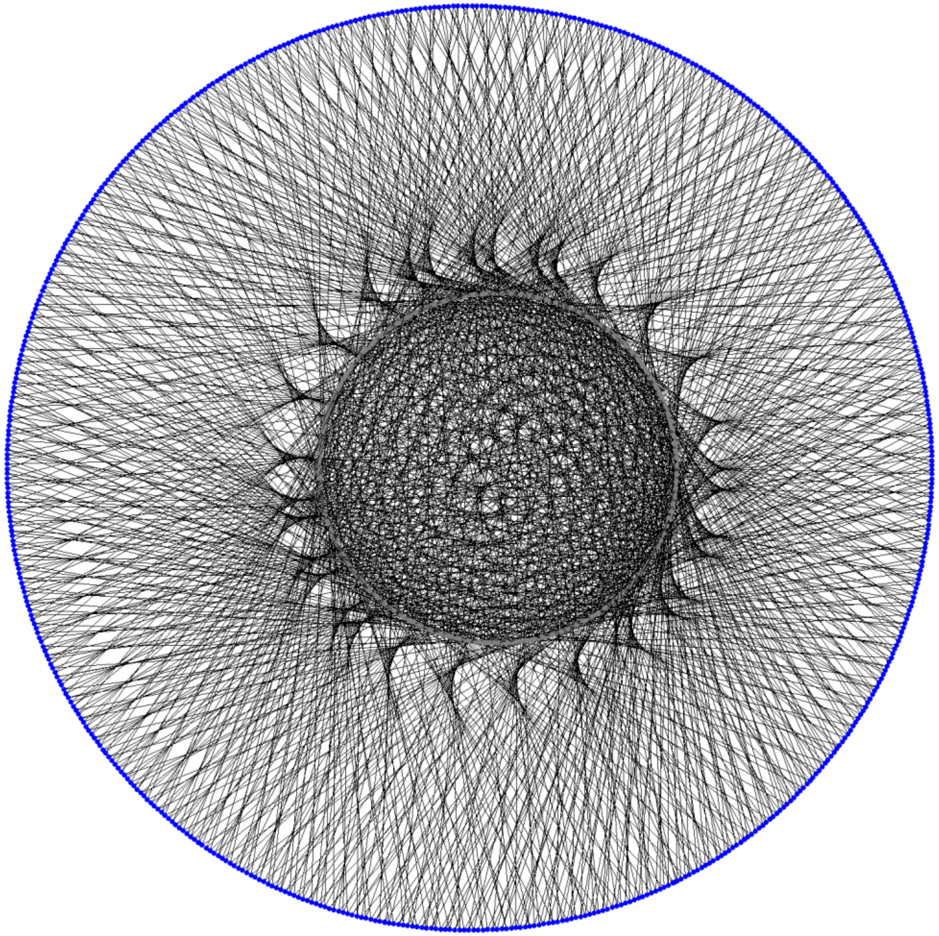
Choose a pair of planets and determine their orbits vs time. At time intervals of Δt, draw a line between the planets and plot this line. Keep going for N orbits of the outermost planet.

N = **10 ,** $\boldsymbol{\Delta t}$ **=** N x **maximum orbital period /1234**, might be sensible parameters.

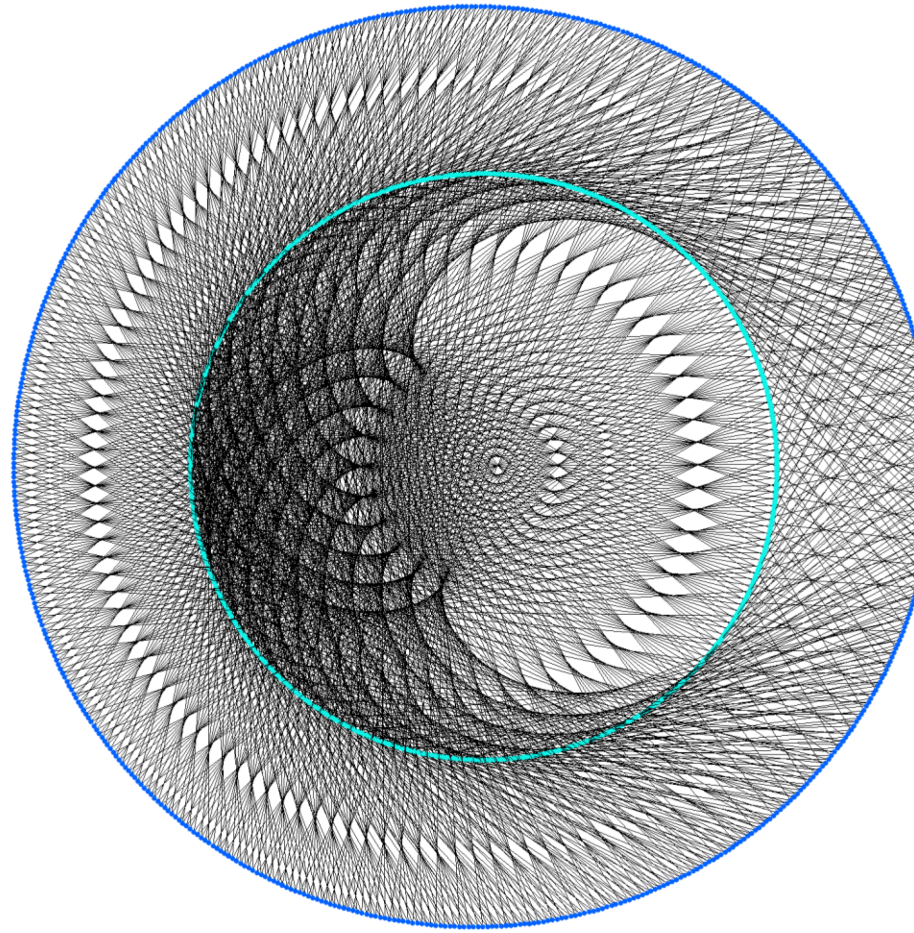




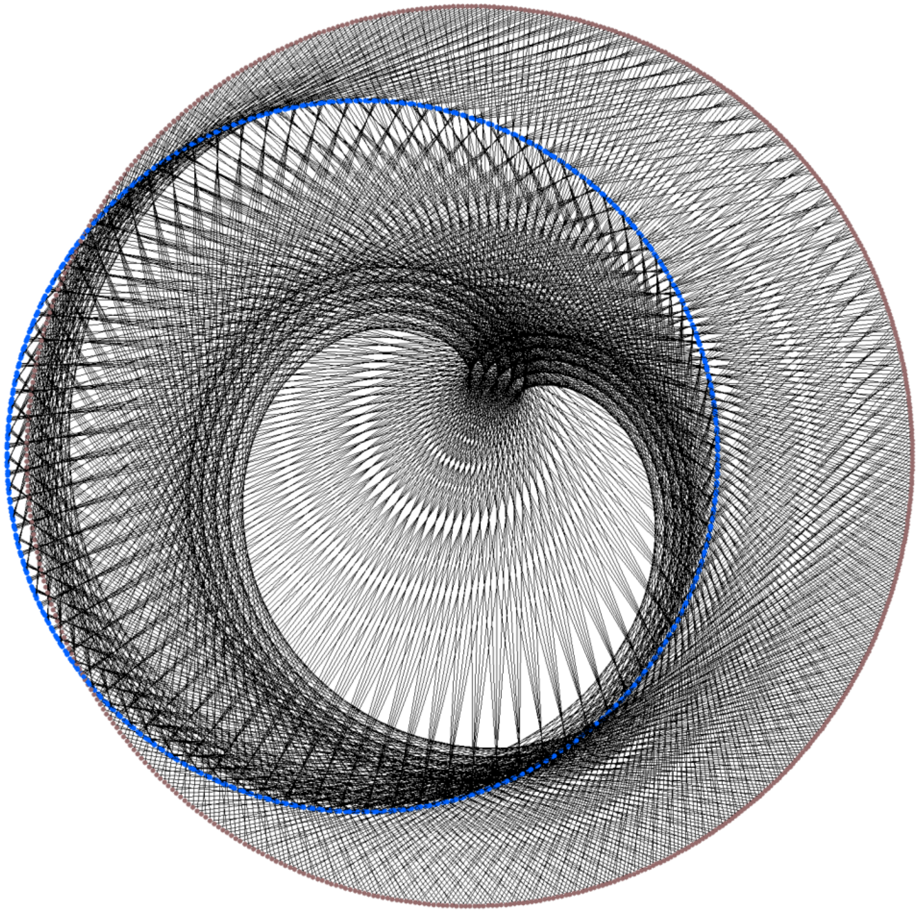
mercury earth spirograph



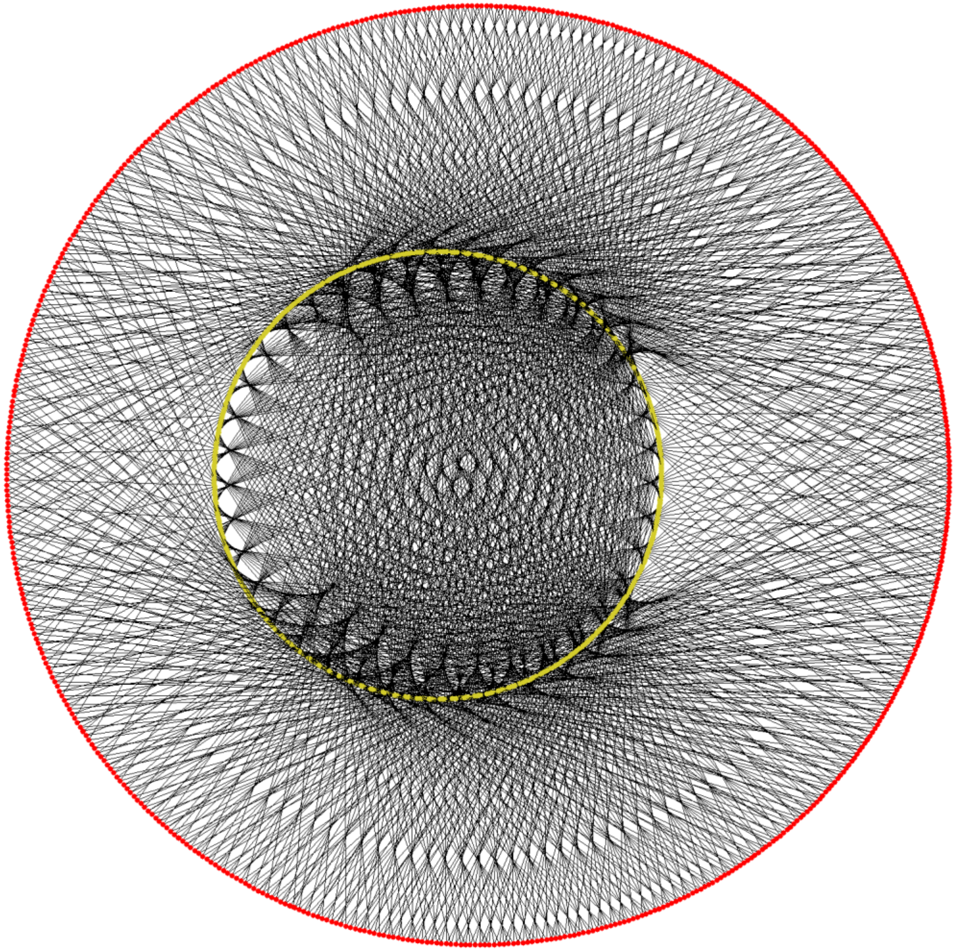
uranus neptune spirograph



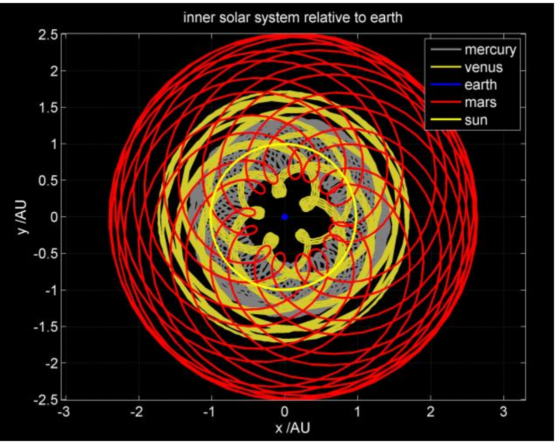
neptune pluto spirograph

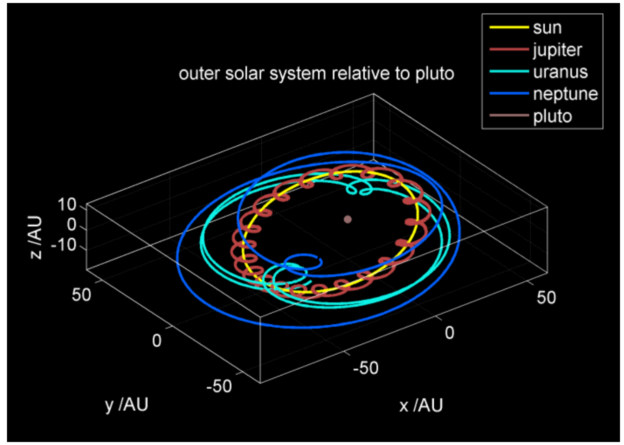


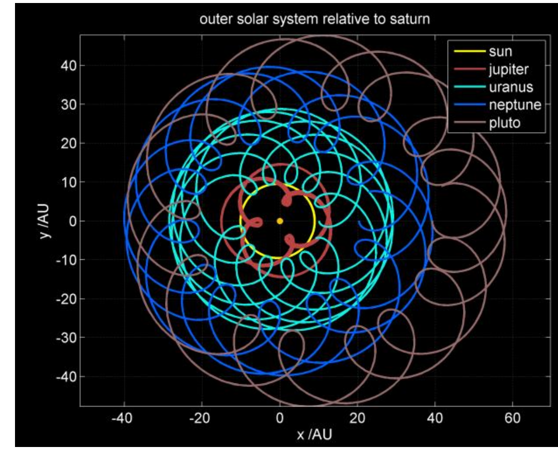
venus mars spirograph

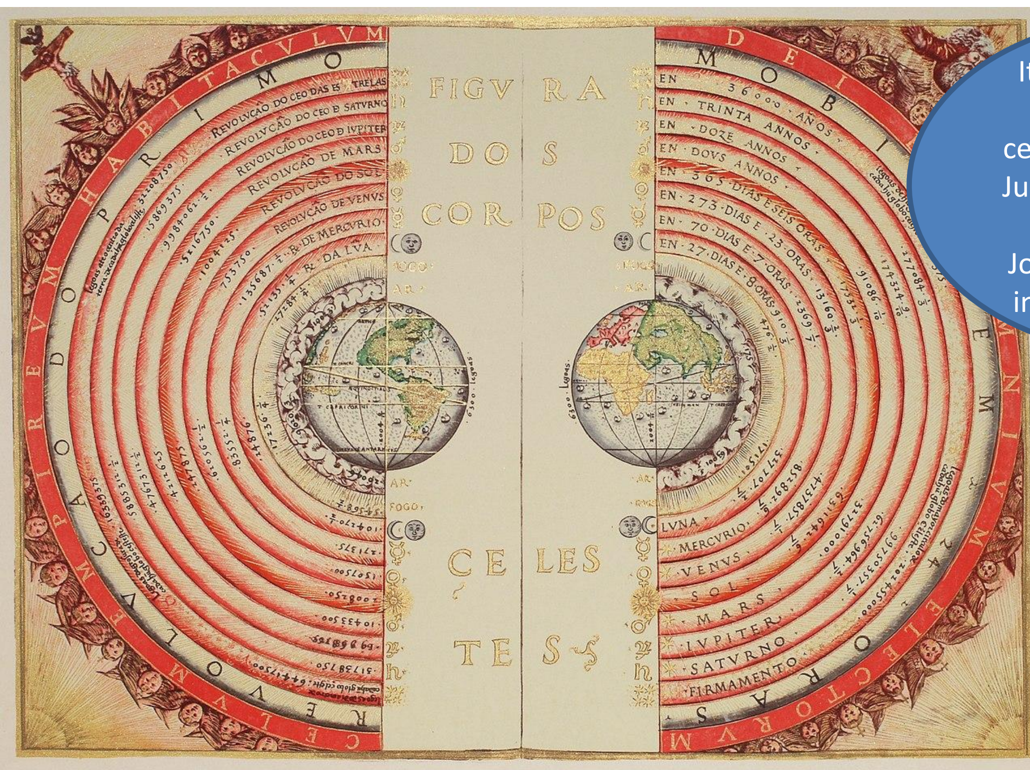


Challenge #7: Use your orbital models to plot the orbits of the other bodies in the solar system, with a chosen object (e.g. Earth) at a *fixed position at the origin of a Cartesian coordinate system.* i.e. choose a coordinate system where your chosen object is at (0,0,0).



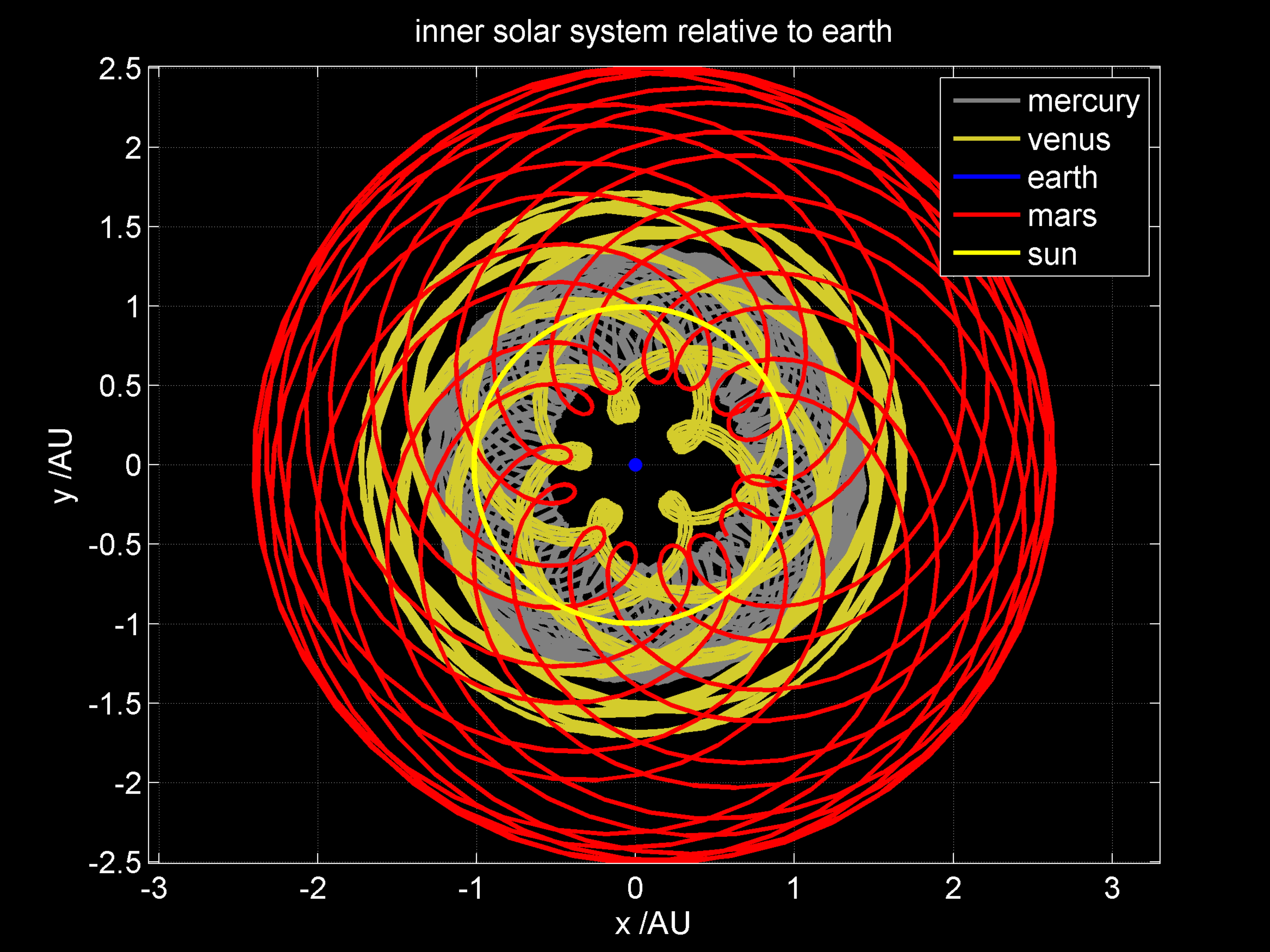


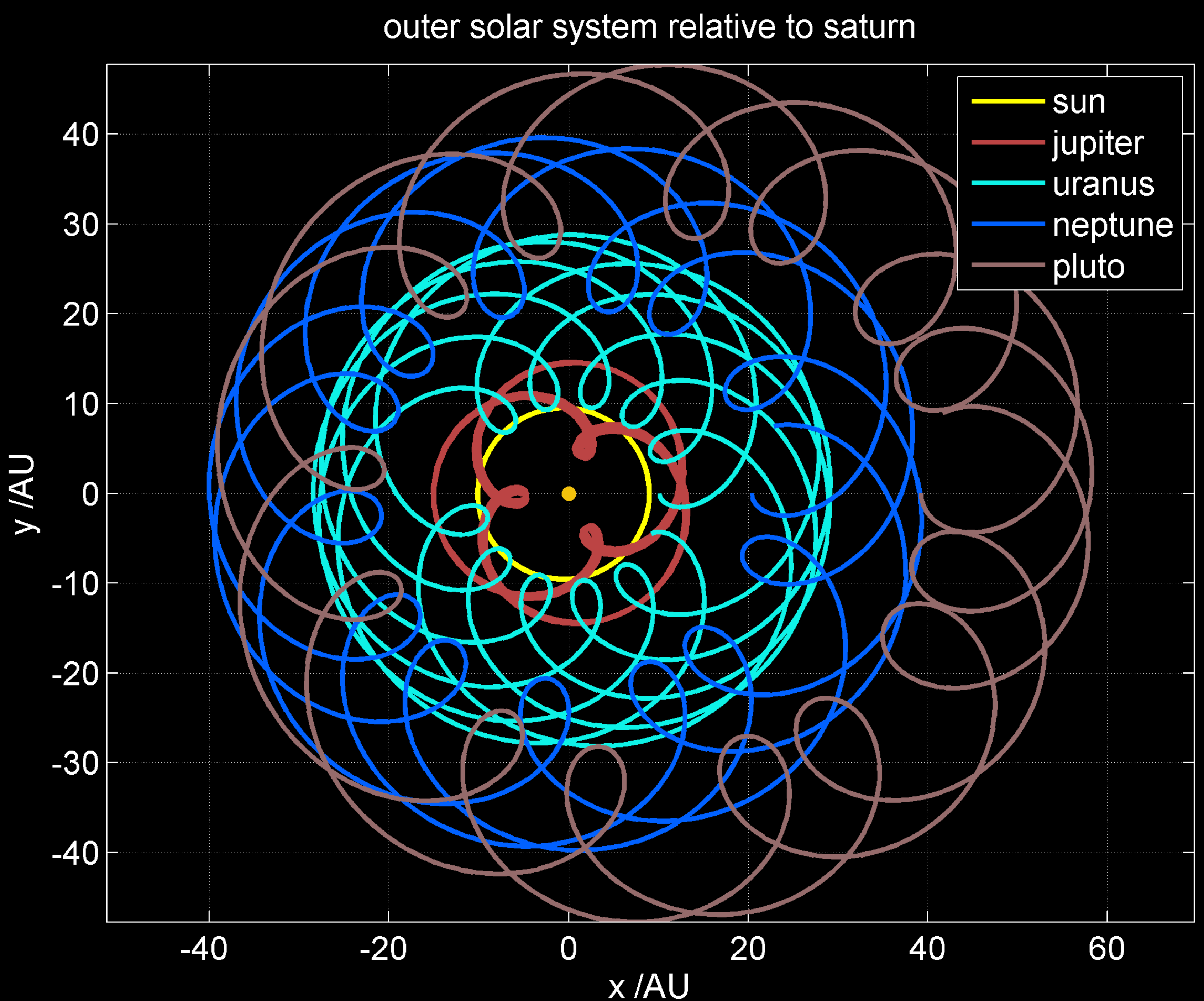




It is perfectly fine for the Earth to be the centre of the Universe! Just don't expect those nice ellipses that Johannes will discover in about 1500 years...

Claudius Ptolemy
(100-170 AD)


inner solar system relative to earth
mercury
venus
earth
mars
sun
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
y /AU
-3
-2
-1
0
1
2
3
x /AU


outer solar system relative to saturn
sun
jupiter
uranus
neptune
pluto
y /AU
x /AU
40
30
20
10
0
-10
-20
-30
-40
-40
-20
0
20
40
60

Extension opportunities:

• Find data about **exoplanets** around stars *other than the Sun*. Stick to single-star systems rather than binaries, as planetary orbits may *not be elliptical* in more-than-two body systems (!) Simulations of many-body systems will be discussed in the online course that precedes this challenge. Recordings of the seminars are available on the BPhO website.

• Write a **graphical user interface** (GUI) for the orbital model and encode this as an 'app'. Coding up an iOS/Android smartphone app will particularly impress the judges!

• Write up your model as a **short paper.** (Aim for about 10 sides of A4, two columns). If you have never written a paper before, download a few from the *Physics Education* journal. *The Epidemiology of Eyam* might be a good start... A good opportunity to learn LaTeX – which is the typesetting language used to write most technical papers and books in the physical sciences. Including *Science by Simulation* *

Don't forget to include any extension projects in your video, as this is the only way you will gain credit for your work in the BPhO Computational Challenge.
I'm afraid we cannot accept any other files. **Submit only the YouTube link to your two-minute screencast.**

* *ScibySim* was created in Scientific Word. There are lots of other LaTeX-based tools available. Find one that works for you!