TEEN ASTRONOMY Café – To Go!

Teacher Guide for the Breaking the Solar System Instant Pack

Would the Solar System look the same if everything was 10 times more massive? What if they were closer together? In this Teen Astronomy Cafe — To Go! Instant Pack, students use simulations to model gravitational interactions between planetary objects and apply data from hot Jupiters to simulate exoplanet systems. Students then predict the stability of the Solar System over time and evaluate how sensitive the stability of the Solar System is to changes in the mass of the Sun, orbital properties of the planets, or length of time. This activity is conducted in a Python Notebook, a web-based interactive computational environment that contains code, text, and plots.

Learning Objectives

Students will be able to:

- Model gravitational interactions between planetary objects in the Solar System and other planetary systems.
- Predict the stability of the Solar System and other hypothetical systems over time.

NGSS Standards

Building Towards NGSS Performance Expectations

- HS-ESS1-4: Use mathematical or computational representations to predict the motion of orbiting objects in the solar system.
- HS-PS2-4: Use mathematical representations of Newton's Law of Gravitation and Coulomb's Law to describe and predict the gravitational and electrostatic forces between objects.

Disciplinary Core Ideas	**HS-ESS1.B: Earth and the Solar System** Kepler's laws describe common features of the motions of orbiting objects, including their elliptical paths around the sun. Orbits may change due to the gravitational effects from, or collisions with, other objects in the solar system. **HS-PS2.B: Types of Interactions** Newton's law of universal gravitation provides the mathematical model to describe and predict the effects of gravitational forces between distant objects. Forces at a distance are explained by fields (gravitational) permeating space that can transfer energy through space.

Science and Engineering Practices	**Developing and Using Models** Students develop data-centric Solar System models to illustrate gravitational force relationships between bodies. Students manipulate and test the models to predict the stability of a system over time. **Using Mathematical and Computational Thinking** Students test the mathematical expressions, computer programs, and algorithms that govern their simulations to compare the outcomes with what is known about Solar System interactions today.
Cross Cutting Concepts	**Scale, Proportion, and Quantity** Algebraic thinking is used to examine exoplanet and Solar System data to predict the effect of a change in one variable (e.g. masses, semi-major axes, and eccentricities) on another. **System and System Models** Computational models of the Solar System and exoplanet systems are developed to simulate and predict gravitational and orbital interactions within the systems. **Stability and Change** Changes within the Solar System simulation demonstrate how rates of change can be quantified and modeled over very short or very long periods of time.

Suggested Timing

- Breaking the Solar System Phenomenon & Presentation (50 minutes)
- Python Notebook intro and how to run a cell (5 minutes)
- Activity 1: Simulate a Single Planet (5 minutes)
- Activity 2: Simulate a Hot Jupiter Exoplanet (10 minutes)
- Activity 3: Simulate the Solar System (20 minutes)
- Activity 4: Break the Solar System (20 minutes)

Breaking the Solar System Presentation

This part of the Teen Astronomy Cafe — To Go! Instant Pack includes a slideshow presentation that is accompanied by a recorded presentation video. The purpose of the slideshow is to provide the audience with background information on the Solar System and its stability and to excite and motivate the learners with the content. A lesson-level phenomenon is suggested to precede the presentation slides and is recommended as a student-driven active learning strategy to initiate curiosity and questioning. Instructions for incorporating the phenomenon are included below.

Breaking the Solar System Phenomenon

1. To begin the lesson, arrange a Driving Question Board with the lesson driving question, “How stable is the Solar System?”
2. To motivate curiosity about and engagement with this topic, observe the phenomenon of The Case of the Backwards Orbiting Asteroid as a whole class.
    - Teacher background information: This video shows Jupiter’s trojan asteroids orbiting ahead of and behind the planet. Asteroid BZ509, nicknamed Bee-Zed, was discovered on 26 November 2014 by astronomers at Haleakala Observatory. The asteroid’s Hawaiian name is *Ka‘epaoka‘āwela*, meaning the “mischievous (orbit) of Jupiter.” Asteroid BZ509 orbits the Sun with Jupiter in a retrograde and complex orbit, with Jupiter’s gravity keeping the asteroid in a stable orbit. Because the asteroid’s retrograde orbit is in the opposite direction to that of objects that formed in the early Solar System, it is suggested that the asteroid may have an interstellar origin.
3. Additional resources can be administered to highlight the presence of interstellar objects in our Solar System and free-floating exoplanets:
    - ‘Oumuamua: First Known Interstellar Visitor is an “Oddball”
    - Borisov: Gemini Observatory Captures Multicolor Image of First-ever Interstellar Comet
    - Largest Collection of Free-Floating Planets Found in the Milky Way
4. After viewing the video, have students interact with Asteroid BZ509’s orbit in comparison to other orbital properties of the planets using the JPL Orbit Viewer. Students can also explore ‘Oumuamua and Borisov on the orbit viewer.
    - To add Asteroid BZ509 to the orbit viewer, click on the menu → Add/Remove Object → Add the specified object: 514107.
    - To add ‘Oumuamua to the orbit viewer, click on the menu → Add/Remove Object → Add the specified object: 1I/2017 U1
    - To add Borisov to the orbit viewer, click on the menu → Add/Remove Object → Add the specified object: C/2019 Q4
5. Students should brainstorm questions based on their experience of observing the video and interacting with the simulator to build off of the driving question, “How stable is the Solar System?” These questions should be recorded visibly for all students to see on the Driving Question Board.
6. The class should work together to organize the questions into similar categories and revisit them throughout the slideshow presentation and/or Python Notebook.
7. After the Python Notebook activity, refer back to the Driving Question Board and facilitate student discourse as they work together to answer, refine, or ask new questions.

Python Notebook General Information

- Start by going over the operation of a Python notebook: To execute or run a selected cell, click the little play button or hit [Shift + Enter] on your keyboard. Some cells may take a few seconds to render, so be patient!
- If something doesn’t seem to be working correctly (e.g., it can’t find resources such as tools.ipynb, or the first simulation where students don’t have to enter in any values fails), try restarting the notebook (Runtime → Restart).
- To run all the cells at once, go to the “`Runtime`” menu and select the option to “`Run all`.”
- To reset each simulation, run the corresponding cell again. This helps when exploring the function of each parameter.
- Encourage students to copy-paste the work they’ve already done to make things go faster.

Activity 1: Simulating a Single Planet

The first activity walks students through the four steps to running a simulation of a planetary system. Students add a star's name and mass, then add a planet's mass, name, and semi-major axis to their simulation. Part 1.4 runs the simulation with a specified end time of 10 years. The final simulation code for Activity 1, in cell 32, is a useful code to copy and paste for future activities in this notebook.

Activity 2: Simulate a Hot Jupiter Exoplanet

Students will choose a (single) exoplanet from the list hyperlinked in green above the red instructions.

- Students will need to record the stellar mass, planet mass (in Jupiter masses), semi-major axis, and eccentricity, and write them on the worksheet.
- Remember that any orbital elements not in the table should be 0 as per the instructions.
- Students can copy the single-cell simulation example from the previous code cell and paste the code into this section. Students will then need to edit the orbital properties of the copied code to correspond with their selected exoplanet.

The discovery methods for hot Jupiters are pulsar timing (changes in the expected arrival of pulsar signals, which are usually exceedingly regular), radial velocity (velocity variations in the spectral line of a host star due to a planet tugging on it as it orbits), and transits (dips in a star's brightness as planets pass in front of it).

```
In [2]: # Create simulation
        mysim = initialize_simulation()

        # Add a star
        add_star(mysim, name='HD 118203', mass=1.84)

        # Add some planets
        add_planet(mysim, name='HD 118203 b', mass=2.79, a=0.07, e=0.31)

        # Run the simulation and show the movie
        movie = run_simulation(mysim,end_time=5)
        movie
```

Figure 1: An example of code for a Hot Jupiter in Activity 2.

Activity 3: Simulate the Solar System

Students enter the orbital elements for the Sun and giant planets to simulate the Solar System in its current configuration.

- Encourage the students to create an empty copy of `add_planet()` with variables that they can copy four times over without having to retype everything for each new planet.
- Also, the `end_time` will need to be changed.
- Students should add the orbital elements to their cell once they have figured out what they need to do to create the simulation.
- Answers to the questions in the activity:
  Does Neptune make one orbit around the Sun? No
  How do you expect the orbital elements will change on long timescales? Not very much. The Solar System is very stable in its current configuration.

```
In [3]: # Initialize the simulation
        mysim = initialize_simulation()

        # Add the Sun here
        add_star(mysim, name='Sun', mass=1.0)

        # The inner terrestrial planets
        add_planet(mysim, name='Mercury', mass=0.00017, a=0.38, e=0.22, i=7.1, omega=30,  Omega=48,  f=201)
        add_planet(mysim, name='Venus',   mass=0.0026,  a=0.74, e=0.02, i=3.4, omega=91,  Omega=7,   f=347)
        add_planet(mysim, name='Earth',   mass=0.0031,  a=1.00, e=0.01, i=0.0, omega=335, Omega=133, f=86)
        add_planet(mysim, name='Mars',    mass=0.00034, a=1.51, e=0.09, i=1.9, omega=292, Omega=49,  f=281)

        # Add the giant planets here
        add_planet(mysim,name='Jupiter',mass=1,      a=5.2 ,e=0.05,i=1.3,omega=275,Omega=101,f=268)
        add_planet(mysim,name='Saturn', mass=0.3,    a=9.5 ,e=0.05,i=2.5,omega=339,Omega=114,f=202)
        add_planet(mysim,name='Uranus', mass=0.046,  a=19.2,e=0.05,i=0.8,omega=97, Omega=74, f=225)
        add_planet(mysim,name='Neptune',mass=0.054,  a=30.1,e=0.01,i=1.8,omega=274,Omega=132,f=302)

        # Pluto-Charon, just for fun. :)
        add_planet(mysim, name='Pluto-Charon', mass=0.000007, a=39.5, e=0.25, i=17.1, omega=114, Omega=110, f=69)

        # Run the simulation
        movie = run_simulation(mysim, end_time=100)
        movie
```

Figure 2: An example of Solar System orbital elements for Activity 3.

Optional Discussion Questions:

- Does the Solar System look different than students expected (in terms of masses, distances, etc.)? Most people think the planets are much closer together than they are.
- There are more than just planets in the Solar System. There's the asteroid belt between Mars and Jupiter, the Kuiper Belt beyond Neptune (where Pluto-Charon lives), and the Oort cloud at the outer edge of our Solar System, plus comets, etc. Are any of these objects important to include? Why or why not?
    - No; they're all really low mass and don't contribute to the dynamics of the planets in the Solar System today. You'd have to make a really large change (factors of hundreds to thousands in mass) to break the Solar System.
- Pluto used to be a planet and got demoted to a dwarf planet in 2006. Why is Pluto not a planet?
    - Look at its orbit. It's inclined, eccentric, and very low mass compared to the other planets. It acts as more of a "tracer particle" of Neptune's dynamical history rather than as a "real" planet.
- How do the hot Jupiters we explored with the first simulation compare/contrast to the Solar System? What is the farthest-out hot Jupiter, and how does that compare to Mercury?
    - In the table, the longest-period hot Jupiter orbits in ~9 days, or at about 0.1 AU. An astronomical unit, or AU, is a unit of length equal to the average distance of Earth to the Sun. Mercury orbits in 88 days at 0.4 AU.
    - If your group is really ahead, have them put a hot Jupiter in the Solar System and see what happens. You can run for 10,000 years, which means the hot Jupiter orbits over 400,000 times. (Spoiler, nothing really bad happens for the average hot Jupiter. Even choosing the most eccentric doesn't incite instability, although it does pump the eccentricity of the inner planets slightly.)

Activity 4: Breaking the Solar System

This is where it starts to get a little more complicated, but also a lot more fun. Students must start with the worksheet to help codify what they want to investigate. It is important to discuss these ideas *before* starting the coding activity. There should be a solid answer for every question before letting students move to the next question. Please encourage students to come up with their own creative answers (as long as they can justify them) without explicitly giving them answers.

Encourage your groups to set both a time scale (less than 10 million years [Myr], which is what the activity instructs) and a criterion based on the orbits.

The reason that explicitly defining this is so important is that another group might try the same simulation with a different criterion and get a different answer. There's no one right answer, so we need to know the underlying assumptions to compare.

Now we get to break things!

- Challenge students to find the smallest change they can make to break the Solar System. It's easy to make Jupiter super large and explode everything.
- Have the students copy the stable Solar System they created in the previous section to this section and delete what they don't want to simulate.
- Note that you can use expressions in the function inputs to make changes a little easier to follow (for instance, mass=1*10 or a=5/2; see examples below).
- It can be helpful to use the "skip to end" button in the animation and step back frame by frame to see the catastrophic instability.
- Because of the way that the backend of the code has been written, simulating to different `end_times` may introduce numerical artifacts into the orbit (meaning that if you see a planet crossing at 10 Myr and simulate the same system out to 11 Myr to see if something happens, the planets may not cross). Instability is susceptible to this numerical error, and unfortunately, this is the way the code has to work (unless we want things to take 10x longer).

Optional Discussion Questions:

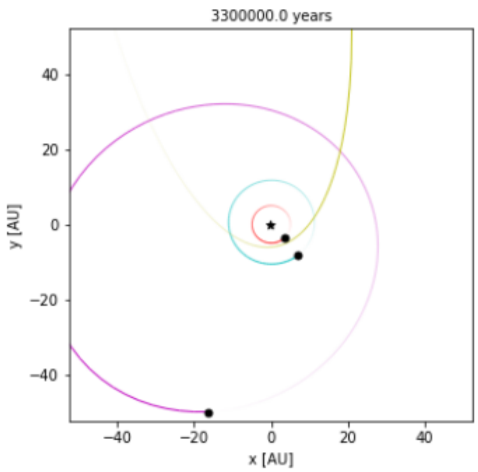
- Observe which planet undergoes instability and speculate why.
    - It's almost never Jupiter. This is because the gravitational force more easily "tosses around" a less massive planet (equal and opposite forces, but with different mass).
- What does instability typically look like? Does the semi-major axis or eccentricity (or something else) change most dramatically? Can you pinpoint in the movie or plot when a system is about to experience catastrophic instability?
    - It's frequently eccentricity that goes first, and a planet will gain eccentricity before getting ejected. You can see that two-planet orbits get close to each other before an ejection.
- How would you want to change the simulation if they had more time or more computational resources? What might you want to investigate?
- Think about how you could apply this to other planetary systems. How might we use these sorts of simulations to understand observations?
    - Simulations of stability can help us place upper limits on the masses of planetary systems (if the planets are too massive, the gravitational force causes instability) or help us understand the lifetime of an observed system.
        - The exoplanet system HR8799 has four extremely large planets (~5–10 Jupiter masses). How must the planets be arranged such that they are stable (are they close or far apart)?
            - The planets are super spread out and are separated by more than 10 AU
        - The exoplanet system Kepler-11 has six planets within 0.5 AU. Are the planets massive (gas giants) or not?
            - The planets are all less than ~10 Earth masses, so quite small, comparatively.

Example: Sun with 10x smaller mass

In [2]:
```python
# Parameter being tested: Small Sun

# Initialize the simulation
mysim = initialize_simulation()

# Add the Sun here
add_star(mysim, name='Sun', mass=1./10)

# Add the giant planets here
add_planet(mysim, name='Jupiter', mass=1,      a=5.2 , e=0.05, i=1.3, omega=275, Omega=101, f=268)
add_planet(mysim, name='Saturn',  mass=0.3,    a=9.5 , e=0.05, i=2.5, omega=339, Omega=114, f=202)
add_planet(mysim, name='Uranus',  mass=0.046,  a=19.2, e=0.05, i=0.8, omega=97,  Omega=74,  f=225)
add_planet(mysim, name='Neptune', mass=0.054,  a=30.1, e=0.01, i=1.8, omega=274, Omega=132, f=302)

# Run the simulation
movie = run_simulation(mysim,end_time=10E6)
movie
```

```
Progress: 33%  3300000/10000000 [00:04<00:09, 714037.37it/s]

Neptune had a catastrophic instability detected at time 3300000.0 years
Its initial orbital elements at time=0 are: a=30.1 and e=0.01
Its orbital elements at time=3300000.0 are: a=96.99 and e=0.94
```

Out[2]:

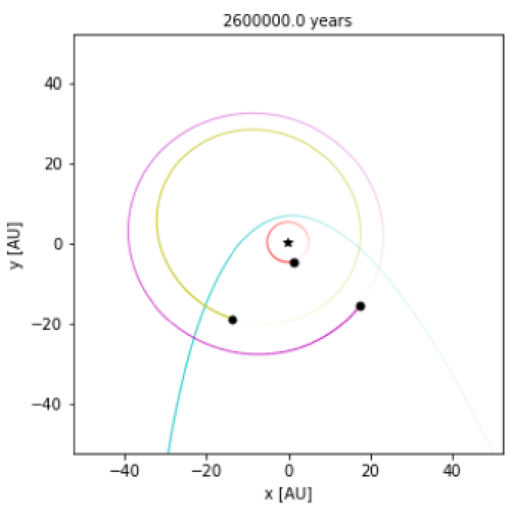


Example: Jupiter with 6x mass

In [3]:
```
# Parameter being tested: Large Jupiter

mysim = initialize_simulation()

# Add the Sun here
add_star(mysim,name='Sun',mass=1)
add_planet(mysim,name='Jupiter',mass=1*6,    a=5.2 ,e=0.05,i=1.3,omega=275,Omega=101,f=268)
add_planet(mysim,name='Saturn', mass=0.3,    a=9.5 ,e=0.05,i=2.5,omega=339,Omega=114,f=202)
add_planet(mysim,name='Uranus', mass=0.046,  a=19.2,e=0.05,i=0.8,omega=97, Omega=74, f=225)
add_planet(mysim,name='Neptune',mass=0.054,  a=30.1,e=0.01,i=1.8,omega=274,Omega=132,f=302)

# Run the simulation
movie = run_simulation(mysim,end_time=10E6)
movie
```

```
Progress: 26%  2600000/10000000 [00:10<00:29, 250021.69it/s]

Saturn had a catastrophic instability detected at time 2600000.0 years
Its initial orbital elements at time=0 are: a=9.5 and e=0.05
Its orbital elements at time=2600000.0 are: a=658.92 and e=0.99
```

Out[3]:

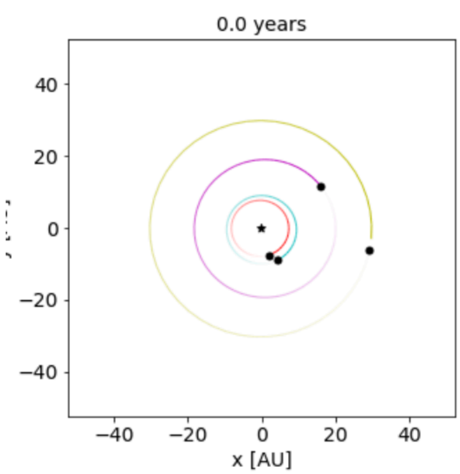


Example: Jupiter semi-major axis x1.5

```
# Parameter being tested: semi-major axis x 1.5

# Add your simulation below
# Initialize the simulation
mysim = initialize_simulation()

# Add the Sun here
add_star(mysim, name='Sun', mass=1)

# Add the giant planets here
add_planet(mysim, name='Jupiter', mass=1.0, a=7.8, e=0.05, i=1.3, omega=275,  Omega=101,  f=268)
add_planet(mysim, name='Saturn',  mass=0.3,  a=9.5, e=0.05, i=2.5, omega=339,  Omega=114,  f=202)
add_planet(mysim, name='Uranus',  mass=0.046,  a=19.2, e=0.05, i=0.8, omega=97, Omega=74, f=225)
add_planet(mysim, name='Neptune',   mass=0.054, a=30.1, e=0.01, i=1.8, omega=274, Omega=132,  f=302)


# Run the simulation
movie = run_simulation(mysim, end_time=10E6)
movie
```

Progress: 1% 100000/10000000 [00:00<00:43, 225154.13it/s]

```
Uranus had a catastrophic instability detected at time 100000.0 years
Its initial orbital elements at time=0 are: a=19.2 and e=0.05
Its orbital elements at time=100000.0 are: a=444.08 and e=0.984
```



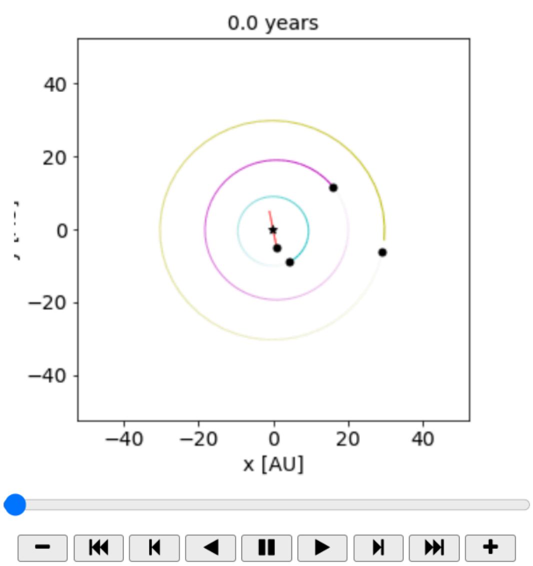
Once Loop Reflect

Example: Inclined (90 degrees) Jupiter (this one makes one of the coolest movies)

```
# Parameter being tested: Jupiter Inclined 90 degrees

# Add your simulation below
# Initialize the simulation
mysim = initialize_simulation()

# Add the Sun here
add_star(mysim, name='Sun', mass=1)

# Add the giant planets here
add_planet(mysim, name='Jupiter', mass=1.0, a=5.2, e=0.05, i=90, omega=275,  Omega=101,  f=268)
add_planet(mysim, name='Saturn',   mass=0.3,   a=9.5, e=0.05, i=2.5, omega=339,  Omega=114,   f=202)
add_planet(mysim, name='Uranus',   mass=0.046,  a=19.2, e=0.05, i=0.8, omega=97, Omega=74, f=225)
add_planet(mysim, name='Neptune',    mass=0.054, a=30.1, e=0.01, i=1.8, omega=274, Omega=132,  f=302)


# Run the simulation
movie = run_simulation(mysim, end_time=10E6)
movie
```

Progress: 5% 500000/10000000 [00:02<00:55, 170216.66it/s]

```
Neptune had a catastrophic instability detected at time 500000.0 years
Its initial orbital elements at time=0 are: a=30.1 and e=0.01
Its orbital elements at time=500000.0 are: a=37.82 and e=0.914
```

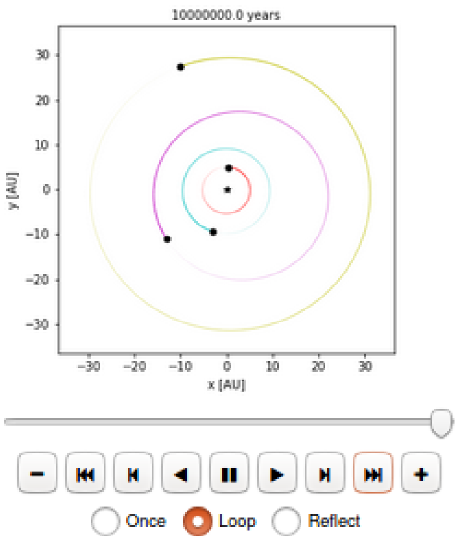


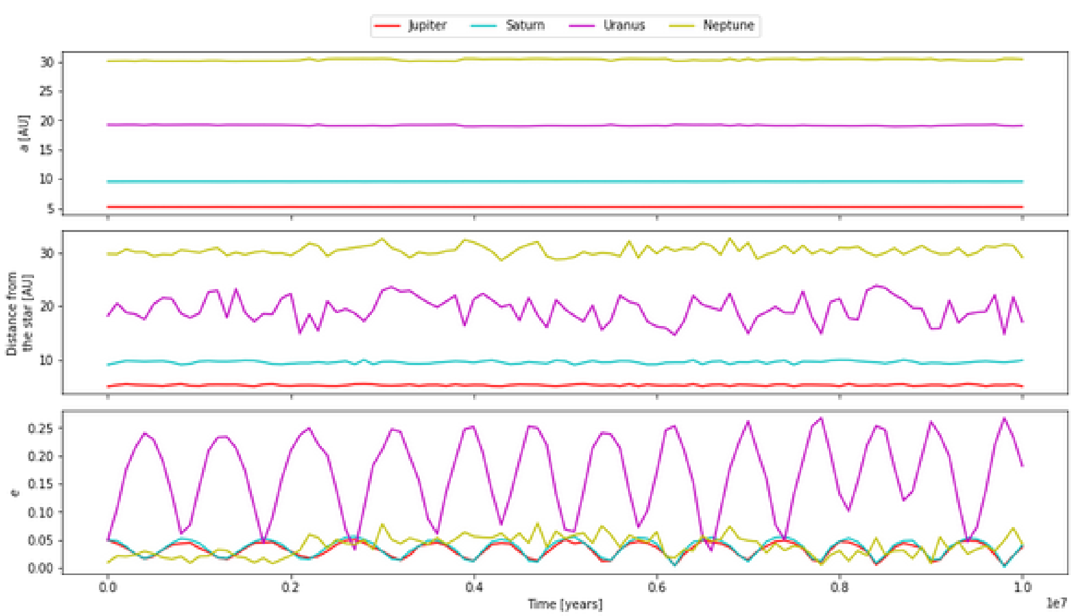
Example: All planets start aligned with a 3x massive Saturn
(While the system doesn't lose planets, Uranus has a 5x higher eccentricity)

In [10]:
```
# Parameter being tested: alignment

mysim = initialize_simulation()

# Add the Sun here
add_star(mysim,name='Sun',mass=1)
add_planet(mysim,name='Jupiter',mass=1,     a=5.2 ,e=0.05,i=1.3, omega=0, Omega=0, f=0)
add_planet(mysim,name='Saturn', mass=0.3*3, a=9.5 ,e=0.05,i=2.5, omega=0, Omega=0, f=0)
add_planet(mysim,name='Uranus', mass=0.046, a=19.2,e=0.05,i=0.8, omega=0, Omega=0, f=0)
add_planet(mysim,name='Neptune',mass=0.054, a=30.1,e=0.01,i=1.8, omega=0, Omega=0, f=0)

# Run the simulation
movie = run_simulation(mysim,end_time=10E6)
movie
```

Progress: 100% 10000000/10000000 [00:38<00:00, 256845.27it/s]

Out[10]:



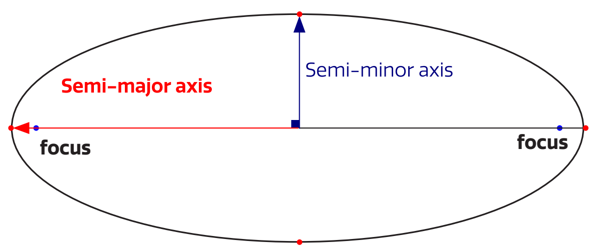
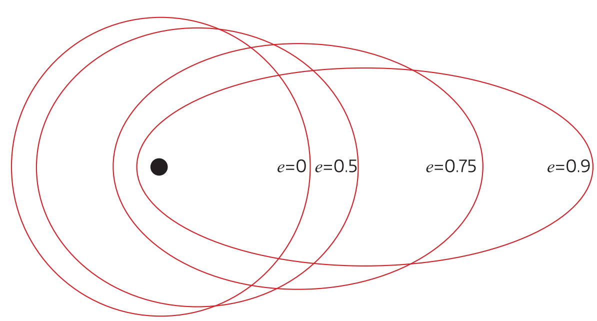


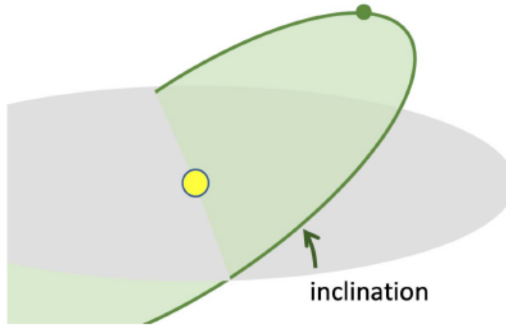
TEEN ASTRONOMY Café – To Go!

Student Worksheet for the Breaking the Solar System Instant Pack

Answer Key

Orbital Parameters

Parameter	Symbol	Code Name	Description	Range/Units
Mass	*M*	`mass`	Mass of the star/planet	stars: $M_\odot$ planets: M_{Jup}
Semi-major axis	*a*	`a`	Average size of the orbit  	AU
Eccentricity	*e*	`e`	How circular or elliptical the orbit is  	0–1 (no units; 0=circle, 1=line)

Inclination	i	`i`	Tilt of the orbit relative to a reference 	0–90 (degrees)
Argument of pericenter, Longitude of ascending node	ω, Ω	`omega`, `Omega`	Angles that describe the orientation of a planet's orbit	0–360 (degrees)
True anomaly	f	`f`	Angle that describes a planet's current position along its orbit	0–360 (degrees)

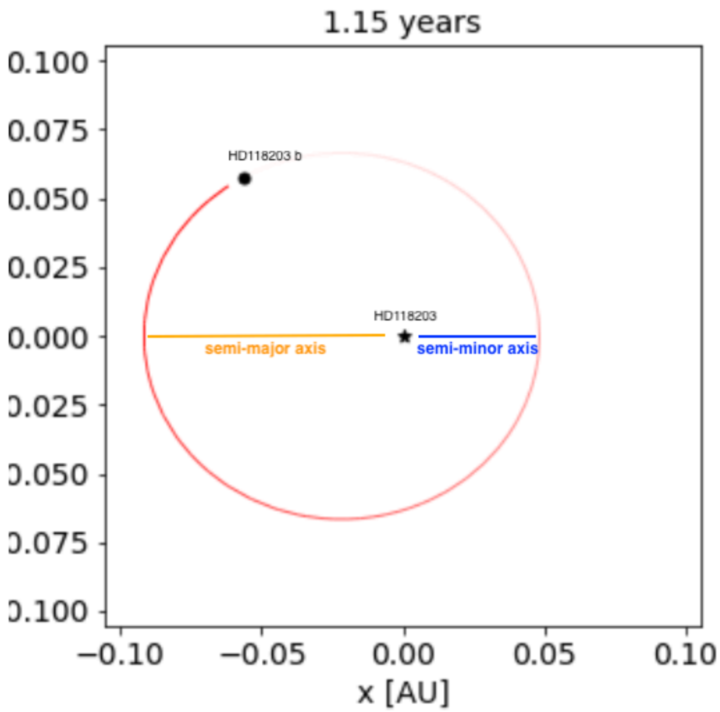
Activity 2: Simulating a Hot Jupiter Exoplanet

1. Which hot Jupiter did you choose?
   Hot Jupiter selections vary.

2. Write down the orbital properties you'll need for your simulation.
   Example:

```
In [2]: # Create simulation
mysim = initialize_simulation()

# Add a star
add_star(mysim, name='HD 118203', mass=1.84)

# Add some planets
add_planet(mysim, name='HD 118203 b', mass=2.79, a=0.07, e=0.31)

# Run the simulation and show the movie
movie = run_simulation(mysim,end_time=5)
movie
```

3. Sketch the orbit you saw in your simulation below. Be sure to include axis labels and units!
Example:



Activity 3: Simulating the Solar System

Below is a table of orbital parameters for objects in our Solar System.

Star	**Mass ($M_\odot$)**	**a**	**e**	**i**	**ω**	**Ω**	**f**
Sun	1	–	–	–	–	–	–
Planet	**Mass (M_{Jup})**	**a**	**e**	**i**	**ω**	**Ω**	**f**
Mercury	0.00017	0.38	0.22	7.1	30	48	201
Venus	0.0026	0.74	0.02	3.4	91	7	347
Earth	0.0031	1.00	0.01	0.0	335	133	86
Mars	0.00034	1.51	0.09	1.9	292	49	281
Jupiter	1.0	5.2	0.05	1.3	275	101	268
Saturn	0.30	9.5	0.05	2.5	339	114	202
Uranus	0.046	19.2	0.05	0.8	97	74	225
Neptune	0.054	30.1	0.01	1.8	274	132	302
Pluto Charon	0.000007	39.5	0.25	17.1	114	110	69

4. What do you notice about the properties of the planets in our Solar System?
   Answers here will vary. Students should realize the masses of the giant planets are larger than those of the inner terrestrial planets, with Jupiter having the largest mass. The average size of the orbit increases the further the planet is away from the Sun. Students should note that most of the planets have low eccentricities and inclinations, placing the planets roughly on the same plane.

Activity 4: Breaking the Solar System

5. What does it mean to "break" the Solar System?
   Answers here will vary. Breaking the Solar System means disrupting the planets from their current orbits. This is what we frequently mean by stability: the orbits will remain unchanged over long timescales. Instability ("breaking" the system) can look like planet orbits crossing, eccentricities growing, planets being ejected from the system, etc. Typically, instability occurs because perturbations between pairs of planets are large enough to substantially modify the orbit around the Sun.

6. Which parameters do you think the Solar System's stability is most sensitive to? Think in terms of the orbital elements, time, etc.
The masses, semi-major axes, and eccentricities are the most sensitive. This is fairly intuitive if you think about the gravitational force, which depends on masses and distances between bodies. Masses are (hopefully) fairly obvious. Changing the semi-major axis will shrink/enlarge an orbit, meaning that neighboring planets can be closer. Similarly, eccentricity makes for a more elliptical orbit, so an eccentric planet will get closer to the orbits of its neighbors.

Secondarily, orbital elements like `omega`, `Omega`, and `f`, which control the relative orientation of the orbits, can contribute to stability. For instance, imagine what would happen if you started all of the planets in conjunction (aligned in their orbits): the mutual perturbations would be at their strongest.

Inclination is also a fun one. Having an inclined planet tugs the rest of the planets upward. This can cause the planetary system to start warping.

Also, time is a factor to consider when defining stability. The Solar System might remain "stable" for a few orbits if Jupiter is 10 times as massive as it is today, but that same simulation might become unstable if we simulate for 10,000 orbits.

7. Evaluate if the following Solar System objects need to be included in your simulations. Explain your reasoning for each object.

   a. Sun: Yes, every solar system needs a star.
   b. Inner planets: Not really (but they can be). These planets are so small and insignificant that they won't easily break the Solar System. Even increasing the masses of the four terrestrial planets by 100x doesn't change anything over 10 million years (Myr). Also, including the inner planets will really slow the simulations down. I would encourage the students to not include the inner planets, but, if they understand the "risks", go ahead. This will typically require a more dramatic change to see instability in a short simulation time.
   c. Outer planets: Yes. The massive planets (Jupiter especially) control most of the dynamics in our Solar System.
   d. Pluto-Charon: Nope. It's too small to do anything. But, it can make for a fun "test particle" that will get thrown about by Neptune.

8. In your groups, decide on the **specific** criteria you will use to determine whether the Solar System "breaks" during a simulation and record the criteria below.
Some possible answers include:
   - Having an ejected planet within 10 Myr. (This is the really straightforward one that the code encourages).
   - Seeing the eccentricity change by more than 50% in 10 Myr.
   - Seeing planets cross orbits in 10 Myr.

Record what you learn from the simulations in the table below.
Answers will vary in the table below. See the teacher guide for example scenarios.

Parameter tested. How did you test it (e.g., what values did you use)?	Describe what happens to the Solar System.	Did you break the Solar System? How long did it take?