

Houston, We Don't Have a Problem: Designing Tools to Develop Intuition

Dr. Kaela M Martin, Embry-Riddle Aeronautical University, Prescott

Kaela Martin is an Assistant Professor of Aerospace Engineering at Embry-Riddle Aeronautical University, Prescott Campus. She graduated from Purdue University with a PhD in Aeronautical and Astronautical Engineering and is interested in increasing classroom engagement and student learning.

Dr. Elif Miskioglu, Bucknell University

Dr. Elif Miskioglu is currently an Assistant Professor of Chemical Engineering at Bucknell University. She graduated from Ohio State University in 2015 with a PhD in Chemical Engineering, and is interested in student learning in engineering. In particular, her work focuses on various aspects of students' development from novice to expert, including development of engineering intuition, as well as critical thinking, problem-solving, and communication skills.

Maciek Czyz

Maciek Czyz is a senior studying aerospace engineering at Embry-Riddle Aeronautical University. He has been there for two years, after previously completing a degree in physics. He is involved in several organizations and projects on campus, one of which is a NASA Space Grant to develop a tool to improve student intuition in space mechanics. Maciek will be interning at NASA's Jet Propulsion Laboratory this summer, and plans to complete graduate school before becoming an astronaut and working on the moon.

Houston, We Don't Have a Problem: Designing Tools to Develop Intuition Regarding Orbital Mechanics

Abstract

Students studying aerospace engineering are expected to be capable of calculating an orbital trajectory based on initial conditions or data but that does not mean they have an intuitive feel for how each parameter affects the final orbit. By manipulating aspects of a spacecraft such as its launch conditions, thrust vector, or time of flight and immediately seeing the results of their manipulation, students can develop a sense of how various conditions affect spaceflight and make use of that intuition in the classroom and workplace.

The purpose of this project is to develop a tool that allows for easy to use and easy to understand demonstrations of orbital mechanics. We start with a simple n-body propagator, using our solar system as the model, to establish baseline trajectories. After using this software, students should have a deeper understanding of how the elements of spacecraft and mission design work together to create an orbital trajectory. Students should also be capable of designing a basic deep space mission using the tools provided.

Introduction

A common faculty complaint is that students turn in solutions that are ridiculous without considering if the answer is even feasible. These complaints seem more apparent when students think of software as a black box that always spits out reasonable answers. Today's engineers must be able to use technology in an intelligent and critical manner and determine if their solution is feasible or infeasible. We aim to help develop this intuition by means of a simulation tool.

Intuition development is frequently linked to experience and can result in better decisions in complex situations [1]. Intuition can be described as subconsciously following a set of rules [2]-[4]. A number of theories attempt to explain intuition development. Fuzzy-Trace theory states that learning evolves from literal interpretation (verbatim) to non-literal interpretation (gist). Intuition is closely connected to gist transfer where learners create links across unique concepts and are less likely to be distracted by unimportant information [5]. The Dreyfus Model interprets learning in six levels starting from a novice who simply follows the rules to a master who can solve problems instantaneously and is absorbed in the task. The fifth level of the Dreyfus model, expert, begins when a task is performed intuitively but is not solved instantaneously [6].

This project aims to create a tool which students studying orbital mechanics could use to develop an intuition about spacecraft trajectories. We define engineering intuition as the ability to assess the feasibility of solutions. We have previously measured intuition by evaluating student responses to problems where a decision (would you approve this mission?) followed a calculation. High intuition was indicated by making the correct decision through accurate reasoning. Our previous work suggests that engineering intuition may be best discernible through "stretch" problems. A "stretch" problem is one that requires information beyond the problem statement but is related to prior knowledge which is easily found by other resources such as the

internet or a textbook [7]-[9]. We have not performed a study to determine if this tool will enhance student's learning, but previous studies on virtual experiments have shown learning gains [10]-[13].

Issues with developing intuition often first arise in the decision regarding what problem to use to engage intuition [14]. If the problem is too difficult or requires too much outside information, few will be motivated to solve the problem. For example, in previous studies of aerospace engineering students in an introductory space mechanics class, a gap in intuition was observed between aeronautics and astronautics students. The aeronautics students who do not see how the material is important to their career (i.e. designing airplanes) demonstrate less intuition than the astronautics students on astronautical problems [9]. On the other hand, a problem that is too obvious or has a clear "plug and chug" solution does not require higher-order processes. As developing an appropriate problem can be difficult, creating a broad tool was necessary. The tool can be used by the instructor to implement different problems in space mechanics and allows students to test out different cases for each problem. Furthermore, the tool needed to be written in a programming language which is common in aerospace engineering. Starting with similar problems to the "stretch" problems in the past, we created a broad tool for orbital mechanics students to develop a better understanding of an orbit's characteristics based on its state and orbital elements. Developing this intuition creates a deeper understanding of each aspect of the state and orbital elements and what role they serve in spacecraft navigation.

Tool Development

MATLAB was used to develop this program as it is a common programming language for aerospace engineers. MATLAB also offers advantages with its library of aerospace functions and the App Designer, which simplified the graphical user interface design considerably. This tool is similar to existing programs such as GMAT (General Mission Analysis Tool) and STK (Systems Toolkit). This program is designed to be simpler, so students just beginning to learn orbital mechanics can use it right away and understand what they're doing.

After running the simulation, the program displays a three-dimensional plot of a spacecraft in orbit around a target body given user-defined initial conditions. A three-dimensional plot was chosen because it allows the user to quickly determine if the orbit is behaving in a rational way, by not intersecting with the planet for instance. The user is required to input information about the desired central body, the spacecraft's initial conditions, and the time of flight. The central body is currently limited to the Earth/Moon system, with plans to include other planetary bodies in the future. The code is currently limited to seven bodies (including the spacecraft) to limit the processing power required to propagate. Initial conditions can be either a state vector or classical orbital elements (semi-major axis, eccentricity, inclination, longitude of the ascending node, argument of periapsis, and true anomaly). The state vector consists of X, Y, and Z position and X, Y, and Z velocity in an ecliptic body-centered inertial coordinate frame. The simulation dates are limited between January 1, 2019 and December 31, 2019 in one day increments. This restriction allows for a large sample of starting conditions and flight times while limiting the amount of data required to download.

Once initial conditions are established the orbit is calculated using n-body dynamics. The dynamics problem consists of the spacecraft and primary body at minimum and can include up to five additional bodies. The problem is therefore at minimum a two-body problem and at maximum a seven-body problem. The dynamics are based on relative n-body motion to solve each case numerically. The Runge-Kutta propagation method used in MATLAB's ode45 function is used to propagate the dynamics.

Developing the program began with creating and testing several functions in MATLAB. The code consists of two functions which convert between state vector and classical orbital elements, a function to lookup constants associated with each planet and moon, and multiple plotting functions to compare various plotting methods.

Next, data was manually downloaded for each of the planets and up to four of their moons from the JPL Horizons website. The website provides a web-based limited interface to JPL's HORIZONS system which can be used to generate ephemerides for solar-system bodies [15]. The data is stored as text files which the program can call to retrieve the data it needs.

The user interface was developed first as MATLAB command window prompts, then migrated to the App Designer. App Designer is an environment used for building MATLAB apps. It simplifies the process of laying out the visual components of a user interface. It includes a full set of standard user interface components, as well as a set of gauges, knobs, switches, and lamps as shown in Figure 1. App Designer integrates the two primary tasks of app building – laying out the visual components and programming app behavior [16].



Figure 1: App Designer Components

The user interface was designed in steps. Each step builds off the previous one and provides more for the user to learn about orbital mechanics.

The first step is for the user to input the initial condition for the simulation. The first user interface in Figure 2, shown with values for the International Space Station (ISS), converts between state vectors and classical orbital elements. Starting with this conversion is an important first step because it is desirable for the user to understand how these two methods of describing a spacecraft's orbit are connected. The user can input values for the state vector and see the resulting classical orbital elements or vice versa. Playing with these values allows the user to begin making connections about which values affect each other.

State Vector			Orbital Elements		
X	1292	km	SMA	6780	km
Y	4753	km	ECC	0.0006787	
Z	-4666	km	INC	51.59	deg
VX	-7.043	km/s	LAN	206.1	deg
VY	-0.909	km/s	AOP	80.43	deg
VZ	-2.878	km/s	TA	160.9	deg

Figure 2: User Interface for ISS Initial State Vector and Orbital

After the user inputs the initial conditions, a three-dimensional plot appears. This plot is updated when the user changes values of the state vector or orbital elements. Figure 3 shows the plot with the values for the ISS from Figure 2. The plot shows the Earth at the center, and the initial position and velocity of the spacecraft with respect to the Earth are shown as a red point and arrow, respectively. This visual representation gives the user perspective about where a spacecraft would be if it were in orbit around the Earth. The reason the Earth was chosen is because it gives the user a familiar starting point and has only one moon, which provides a starting point for many-body propagations. The lessons learned here can be easily transferred to an orbit around another planet. On this plot the Earth is shown to scale, so if the spacecraft is not visible, it may be within the planet.

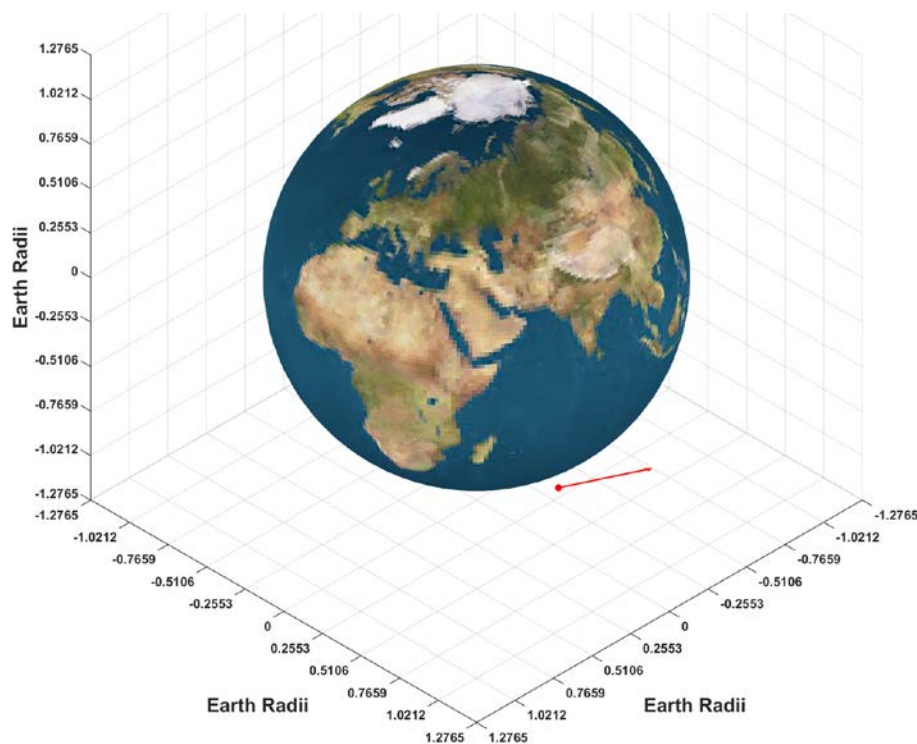


Figure 3: Earth with Initial Spacecraft Position and Velocity

Next the program implements a short propagation using just the Earth and spacecraft to give quick insight about the orbit. When the user is satisfied with their initial position and velocity, they can create a propagation plot by selecting a time on the slider and clicking the plot button shown in Figure 4. The slider allows the user to choose between 1 hour and 120 hours (5 days). This limit prevents a long run time while displaying where the spacecraft will be in the immediate future. The velocity arrow is replaced with a path the spacecraft follows for the user-defined time. The spacecraft's initial position is shown with a red point and its final position is shown with a cyan star. By comparing the velocity arrow in Figure 3 to the expected propagation in Figure 4, the user can see how their initial velocity affects the spacecraft's orbit. The Moon is not included in the dynamics at this point in the calculations, but it will be added later.

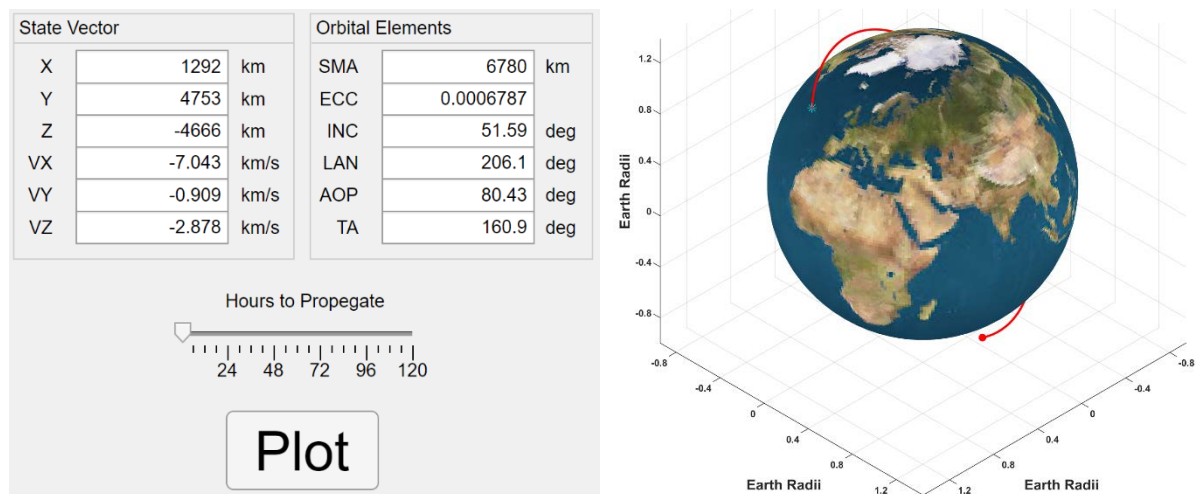


Figure 4: User Interface and Spacecraft Quick Propagation

The final step adds the Moon's gravity, resulting in a four-body propagation. The user can choose whether they want the Moon displayed in the plot. In some cases, like the ISS example, displaying the Moon would make the spacecraft's orbit difficult to see. In other cases, such as orbits which extend further from the Earth, displaying the moon would be desirable as it would show the spacecraft's path near the moon. Figure 5 shows a comparison between three-body (Earth, Sun, Spacecraft) and four-body (Earth, Sun, Moon, Spacecraft) propagation for a fictional spacecraft which passes near the moon. In the three-body propagation in Figure 5a the spacecraft moves towards the Earth, whereas in the four-body propagation the spacecraft moves towards the Moon. The Earth appears much smaller in this plot due to the inclusion of the Moon's orbit

Conclusion

A new tool has been created for students to visualize a spacecraft's orbit under the gravitational influence of the Earth, Sun, and Moon. Students can choose the spacecraft's initial conditions and subsequently gain valuable insight into complicated planetary dynamics. Understanding a spacecraft's initial conditions is an important first step to developing intuition towards orbital mechanics. After using this program for some time, the user should be able to create reasonable orbits from scratch in similar orbital mechanics programs. They would begin to understand fundamentals, like the relationship between a state vector and orbital elements, velocities necessary to maintain an orbit, and the influence of gravity from surrounding bodies.

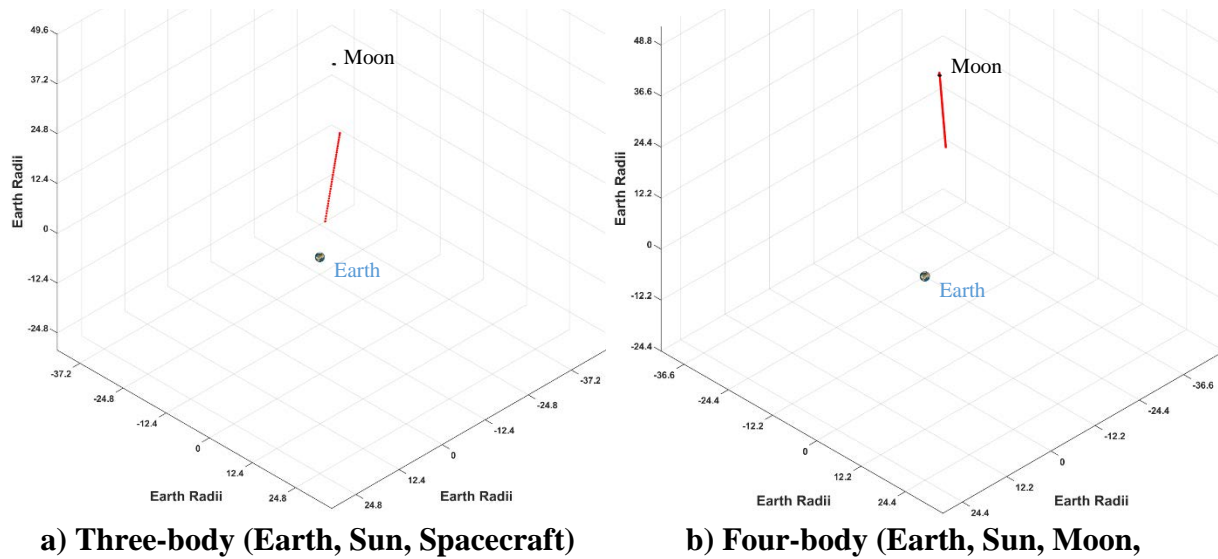


Figure 5: Three-Body and Four-Body Spacecraft Motion

By starting with these first principles, the foundation is laid for understanding more complex trajectories. After using this program, users should be capable of explaining whether an orbit's initial conditions make sense and be prepared to apply this knowledge to more challenging concepts.

Future Work

The next addition to the program is to give the user a choice of primary bodies, allowing the user to see how an orbit which works fine for one planet behaves differently at another planet. This comparison would further demonstrate how gravity plays a role in spacecraft trajectory. Work is being done to include the eight planets and the Pluto/Charon center of mass. The Pluto/Charon center of mass is included because it offers an interesting case where the point being orbited is in empty space. Each planet would include up to its four most massive moons to limit computational resources necessary to complete propagations.

Another desirable quality would be an animation instead of a static plot, allowing the user to watch the spacecraft orbit over time instead of simply seeing the result. Along with the animation, a constantly updating view of the spacecraft's state and orbital elements could be included. Adding the ability to implement a maneuver could also be something to explore. A maneuver would allow the user to make changes during the orbit and see the impact of changing the spacecraft's velocity or direction.

The software has yet to be tested with users. Students will be given a sample problem to solve using the tool, then asked to solve another problem with less information, forcing them to draw from previous observations. We expect to test the program in the spring of 2019 based on previously used problems, and the results may lead to other changes or additions to the program which have yet to be considered. The users' input will be held in high priority for deciding which features to add next. In the future, we plan to pair this tool with an instrument to measure engineering intuition which we are currently developing.

References

- [1] D. G. Elms and C. B. Brown, "Intuitive decisions and heuristics—an alternative rationality," *Civil Engineering and Environmental Systems*, vol. 30, pp. 274-284, 2013.
- [2] P. Raskin, "Decision-Making by Intuition--Part 1: Why You Should Trust Your Intuition," *Chemical Engineering*, vol. 95, p. 100, 1988.
- [3] G. Gigerenzer, *Short cuts to better decision making*. London: Penguin, 2007.
- [4] D. Kahneman, *Thinking, Fast and Slow*. New York, NY: Macmillan, 2011.
- [5] C. R. Wolfe, V. F. Reyna, and C. Brainerd, "Fuzzy-trace theory," *Transfer of learning from a modern multidisciplinary perspective*, p. 53, 2005.
- [6] S. E. Dreyfus and H. L. Dreyfus, "A Five-Stage Model of the Mental Activities Involved in Directed Skill Acquisition," *California Univ Berkeley Operations Research Center*, 1980.
- [7] E. E. Miskioglu and K. M. Martin, "Reasonable or Ridiculous? Engineering Intuition in Simulations," in *Proceedings of the 123rd ASEE Annual Conference & Exposition*. New Orleans, LA, June 2016.
- [8] E. E. Miskioglu and K. M. Martin, "Is the Answer Reasonable or Ridiculous? Common Factors among Students Who Display High Engineering Intuition on Technology-aided Solutions," in *Proceedings of the 124th ASEE Annual Conference & Exposition*, Columbus, OH, June 2017.
- [9] E. E. Miskioglu and K. M. Martin, "Work in Progress: Got Intuition? Exploring Intuition in Response to Technology-aided Problem Solving," in *Proceedings of the 125th ASEE Annual Conference & Exposition*, Salt Lake City, UT, June 2018.
- [10] S. K. Chaturvedi and K. A. Dharwadkar, "Simulation and visualization enhanced engineering education development and implementation of virtual experiments in a laboratory course," *ASEE Annual Conference and Exposition*, 2011.
- [11] T. F. Wiesner and W. Lan, "Comparison of student learning in physical and simulated unit operations experiments," *Journal of Engineering Education*, vol. 93, 2004, pp. 195-204.
- [12] N. Finkelstein, et al., "When learning about the real world is better done virtually: A study of substituting computer simulations for laboratory equipment," *Physical Review Special Topics-Physics Education Research*, vol. 1, 2005.
- [13] C. Hundhausen, P. Agarwal, R. Zollars, and A. Carter, "The design and experimental evaluation of a scaffolded software environment to improve engineering students' disciplinary problem-solving skills," *Journal of Engineering Education*, vol. 100, no. 3 2011, pp. 574-603.

- [14] J. Metcalfe and D. Wiebe, "Intuition in insight and noninsight problem solving," *Memory & Cognition*, vol. 15, no. 3, 1987, pp. 238-246.
- [15] Jet Propulsion Laboratory "HORIZONS Web-Interface," *NASA Jet Propulsion Laboratory California Institute of Technology*. [Online].
Available: <https://ssd.jpl.nasa.gov/horizons.cgi>. [Accessed: Nov. 12, 2018].
- [16] D. Garrison "App Designer" *Mathworks*. [Online]
Available: <https://www.mathworks.com/videos/app-designer-117921.html>. [Accessed: Jan. 3, 2019].