Simulating Planetary Motion using Python

A Data Management Plan created using UCT DMP

Creator: Georgina Gomes

Affiliation: University of Cape Town

Template: UCT Student Generic DMP

Project abstract:

This project is about understanding and visualising how planets move in space due to gravity. The purpose was to simulate the motion of an earth-like planet around the sun using mathematical laws and the computing power of Python. The project starts by showing how Newton's Law of Gravitation keeps the planets in orbit around the sun, and then combines this with Newton's second law to start simulating planetary motion. The simulation of motion is broken down into logical steps using numerical methods such as the Verlet integrator to help calculate the velocity of a planet based on gravity, which allows the resultant position to be calculated. 2D and 3D plots are used to give a visualisation using Matplotlib. This presents a powerful representation of how celestial bodies move in space with a final simulation of the entire solar system. The project outlines clearly the steps, equations and knowledge required to reach the final simulation, with Jupyter Notebook, allowing for readability.

ID: 7156

Start date: 19-01-2020

End date: 19-06-2020

Last modified: 21-03-2025

Simulating Planetary Motion using Python - Student Full DMP

1. Project Details

PROJECT NAME - Replicate the title of your project, dissertation or thesis exactly as it appears in your proposal document.

Simulating Planetary Motion with Python

PERSONAL DETAILS - Indicate the name(s) and student number(s) of the student(s) who will be involved in this project, dissertation or thesis.

Georgina Gomes GMSGEO001, Jemima Lucas, Imogen Beesley, Rachael Bomphrey

SUPERVISOR(S) DETAILS - Indicate who will supervise this project, dissertation or thesis. If you do not yet have a supervisor, leave this section blank.

2. Project Summary

RESEARCH SUMMARY - Briefly summarise your study. Include the study's objectives, design, and methods.

This project logically simulates the planetary motion of an Earth-like planet around the Sun, the physical laws behind this motion, and finally the solar system. Throughout the simulations, the Sun is assumed to remain stationary at the origin in order to simplify the problem. All computations use SI units. Each group member (4 total) was a third year Mathematics BSc student who had at least 6 months Python programming experience before starting this project. We had been taught Python using JupyterLab and so naturally we chose JupyterLab to develop this project. Newton's Law of Gravitation and Newton's second law were used firstly to explain why the planets in our solar system stay in orbit around the Sun instead of flying into space. Then, numerical methods were used, namely the Verlet Integrator, to plot the trajectory of each planet, using matplotlib and numpy. The Verlet Integrator was chosen in particular because it is a symplectic integrator meaning it is designed to preserve the structure/laws of real physical systems. The function earth_orbit was created using the methods named above, taking initial (x, y) position and velocity components as inputs, as well as the number of iterations and total time for the simulation to run. It returns an array of (x, y) components for position, an array of (x, y) components for velocity and a plot of the trajectory. The closed orbit of an Earth-like planet is visually confirmed with a plot generated by the earth_orbit function. A function named totalE was written which takes the same inputs as earth_orbit and outputs total energy, kinetic energy, and gravitation potential energy for different positions in an Earth-like planet's orbit. The function also plots 2 graphs, one displays total, kinetic and potential energy (indicated by different colours) against time, and the other plots only total energy against time. The y-scale is purposely made much smaller in the second plot to reveal small variations in total energy over time. This function is used with the same values used to verify Earth's closed orbit to show that the variations in total energy over time as Earth orbits the Sun, are so small they can be considered negligible i.e. energy is conserved. The average orbital speed of the Earth was proved visually and computationally as the only compatible speed of Earth that will give a near circular closed orbit, via the totalE function. The effect of a lower and higher average speed than 29.8km/s is tested with the totalE function to show the respective elliptical and hyperbolic trajectories and energy against time. The study then mathematically derives the conditions under which a hyperbolic trajectory starts and uses the totalE function to find the first average speed value that causes positive total energy. The totalE function is used again to work out the first average speed that leads to negative total energy i.e. elliptical orbits. The range of average speeds for each trajectory are then used in the earth_orbit function to visualise how the trajectory changes as speed changes. A function aph_per was written which returns four arrays: aphelion points, perihelion points, orbital eccentricities and average orbital speeds. The aphelion and perihelion points arrays are used to plot the perihelion and aphelion points on differing orbit trajectories. A dataframe is created from the corresponding orbital eccentricities and average speed arrays to show clearly under the plot the corresponding values of each trajectory. A function traj_type was written which takes the average orbital speed and position and returns a boolean which represents whether the trajectory is hyperbolic or not. This function is used the determine the escape speed of an Earth-like planet from the sun's gravitational pull. The function cart_2_pol was written which converts cartesian coordinates to polar coordinates. The function kepler_2 was written which takes the output of cart_2_pol and returns an array of the areas of each sector of the ellipse trajectory created in each time step and dt. A function da_dt was written which returns da/dt and t, and this is used to show that dA/dt is constant for a range of average orbital speeds. A function three_body was written which returned a plot of the orbits of the three bodies. Different input masses were used to visualise a stable case, a restricted case and a chaotic case. Finally, the function sol_sys was written, and was used to plot a 3D simulation of the solar system.

3. Description of the Data

DATA REUSE DESCRIPTION - If you are re-using data from third-party sources in your study, record pertinent details here such as the source of the data, the extent of the data, usage rights or restrictions pertaining to the data, and how it incorporates into your study.

- I am using existing data in my study.

All numerical data, equations, and logic were used from lecture notes at University of Bristol Mathematics BSc course (UCAS code G100) This project reuses parameter data and some coding techniques from two third party sources to help simulate the three body problem and planetary motion. The first data source was a Medium.com article by Ben Eagan named "Python N-Body Orbit Simulation". (Link: https://bceagan.medium.com/python-n-body-orbit-simulation-be3fb6356579). The article provided code inspiration and simplified simulating planetary motion which was extremely helpful for the latter part of the project. Core concepts and code structures (e.g. integration method for updating positions and velocities) from this article were used to generate the foundational code to simulate firstly a stable three body simulation and then to simulate the solar system as a whole. We expanded upon the code to simulate our desired outcome. The source was used solely for academic purposes and is fully credited in the project. The content was publicly posted, with no explicit reuse license provided, which means the content is copyrighted by the author and reuse falls under fair use for academic purposes, which is how it was used here. The second data source was a public youtube video from Volker Dörr called "Restricted 3-Body Problem With Osculating Orbits". (Link: https://www.youtube.com/watch?v=qIVe_xEv6zQ). The video gave visual insight of what to expect from our restricted three body problem simulation output and also provided examples of the appropriate parameter values to use. The video is publicly accessible on youtube, under YouTube's Standard License. It was used solely for this project and academic purposes, providing conceptual guidance i.e. fair use. The author of the youtube video is credited where appropriate in the project.

ORIGINAL DATA DESCRIPTION - If you are collecting your own data, describe the data you are gathering for your study. Briefly describe the type, scope and amount of the data you are producing.

- I am not collecting my own data.

4. Formats and Quality Control

QUALITY CONTROL - Describe what measures you are taking to ensure the data you collect are of high-quality.

Quality control of the two third party data sources used was ensured by cross checking with other references and by confirming physical outputs against known mathematical/physical laws. The code from Ben Eagan's Medium.com article was verified by checking the outputs with known physical laws, specifically energy conservation. For the youtube video from Volker Dorr, we actually made contact with him personally to chat about his simulation, as well as comparing his output with others. Null values were not applicable to our simulation but we verified that our outputs made logical sense according to mathematical laws and expected results as per previous experiments. The visualisation of each calculation via the plots also verified numerical and physical accuracy. All mathematical equations used are globally accepted. Triangulation was achieved within the project by comparing the output of multiple different velocity/timestep configurations throughout, and confirming that the behaviour was as expected. We analytically derived parts of our data using globally accepted mathematical principles too.

FILE FORMATS - Indicate the formats in which your data will be collected and processed. Clarify whether you will use specialised, proprietary software to produce and access your data and whether you will convert to open, accessible formats for long term access and preservation? In the case of physical objects (such as artworks or models) indicate whether these will be digitised or otherwise preserved for accessibility.

All data for this project was generated, processed, and presented within a single Jupyter Notebook file, .ipynb format (stands for IPython Notebook), which combines Python code, equations, visual outputs, and explanatory text. It is an open, documented and human readable format based on JSON, maintained by Project Jupyter which is open source. It is a widely used format that is compatible with all major operating systems by default (Windows, macOS, Linux). In order to access/run the notebook file, one must have Python (version 3.x) and Jupyter Notebook installed locally on their system. The user can optionally install Anaconda also, whereby all required packages will be included by default. However, if Anaconda is not installed, "pip install" (a command that becomes available once Python is installed) will be needed to install the packages used in the notebook: matplotlib, pandas, numpy. To open the file once this software has been installed, one can use a variety of platforms such as JupyterLab, VS code, Google Colab etc. As no external datasets or physical objects were used, and all results were generated programmatically, no digitisation or additional preservation measures were necessary. The .ipynb file format is preserved in its native form for long-term accessibility (it is widely supported by open source tools), and can be exported to open formats such as HTML or PDF for presentation purposes if required.

5. Data Management, Documentation and Curation

STORAGE AND BACKUP - Describe how your data is being stored and backed-up. If you are using a data service provider, provide details on how long they will retain the data.

The project was developed and stored using the University of Bristol's OneDrive cloud storage system, which provided secure, version-controlled access for all group members. Shared access to the document was enabled, allowing all four contributors to edit and save changes to the Jupyter Notebook. However, real-time collaborative editing of Jupyter Notebook file is not natively supported by OneDrive. In order to avoid version conflicts and to avoid a group member overwriting / working on the same part of the project as another, we dedicated particular sections of the project to each member, and only one person could edit at a time. After a member made changes, we would review them on a zoom call and then the next person would work on their part. In addition to cloud storage, each group member maintained a local copy of the project on their individual operating systems as a secondary backup. The final version of the project was submitted via email through Outlook. OneDrive, as a managed institutional service, retains data according to the University of Bristol's cloud storage policy, which includes secure backups and data retention even in the case of accidental deletion. After graduating from University of Bristol, each pupil receives an alumni email which allows you to retain access to your University of Bristol One Drive files. One Drive for Business (i.e. University of Bristol One Drive) will retain files indefinitely as long as the account is still active and the file is not deleted, with regular cloud backups ensuring data safety. If by chance the file is deleted, it goes to the recycle bin for 30 days, within which time period it can be restored. After 30 days, it goes to a second stage recycle bin for another 30 days, then after that it is deleted permanently. Additionally, if the user account is deleted, the One Drive data is retained for 30 days, after which it is permanently deleted also.

CURATION (MANAGING AND STORING) DATA - Describe how you organise and manage your data. Specify any file-naming conventions or community data standards you have adopted.

The project data was managed using a single Jupyter Notebook file stored on the University of Bristol's OneDrive cloud system. The file naming format was Project2BeesleyBomphreyGomesLucas.ipynb, which included the project number and the surnames of each group member. The project was one of 3 cap stone project options given by our professor for a Mathematical Programming module, hence why it was named Project2 here instead of "SimulatingPlanetaryMotion". Inside the notebook file, the project is clearly into 3 "CORE" sections with 3 "Extension" sections. Markdown headings, with numbered subheadings and subtitles, were used to make the project easy to navigate and each section was collapsible, making it easy to go quickly through the large file. All outputs were embedded within the notebook, keeping everything in a single document. No additional datasets were used, all data produced was done so programmatically. The final version of the notebook was saved on OneDrive and submitted via email through Outlook. Each group member kept a copy of the project locally on their hard drive too in case we lost access to our UoB OneDrive.

METADATA STANDARDS AND DATA DOCUMENTATION - Describe what metadata and documentation you are producing about the data you are generating, collecting or re-using.

All data in this project was generated programmatically within a Jupyter Notebook using defined parameters and mathematical constants. Metadata and documentation were embedded directly in the notebook using markdown cells, providing detailed explanations/derivations of the equations used, initial conditions, step-by-step logical computation, and visual outputs. It is easy to follow what each part of the project is doing, making the project useful for future referencing. Each section of the notebook was clearly labeled with headings, and all code blocks were annotated with inline comments to describe the purpose of the code and the variables used. SI units were used throughout and stated for all physical quantities to ensure clarity and reproducibility. No external datasets were collected, but reused sources were referenced within the notebook for transparency. While no formal metadata schema was applied, the structure of the notebook and the descriptive nature of each computation, code comments, and output plots served as effective internal documentation for understanding and reusing the data.

6. Data Security and Confidentiality of Potentially Disclosive Information

SECURITY - Indicate to what extent your data can be considered sensitive or at-risk. Describe how you will control access to your data. Indicate whether you anticipate a need for encryption or password-controlled access, and if so, how you will enforce that access.

- My data is not sensitive or at-risk

All content was generated programmatically using open-source packages and public scientific principles. No personal data, identifiable information, or institutional data was collected, stored, or processed. Access to the project during development was controlled via the University of Bristol's OneDrive cloud storage, where shared access was limited to the four group members. Although encryption or password-protected files were not necessary due to the non-sensitive nature of the content, the use of institutional cloud storage provided a secure environment with managed access and version control. The final submission was shared via secure university Outlook email accounts. As the project contains only academic content intended for assessment, no additional security measures such as encryption or restricted distribution were required.

ETHICS AND PRIVACY - Describe, as per your Ethics Clearance form or other similar documentation, any ethical or privacy issues that your data are subject to (if any). Summarise the main risks to the confidentiality and security of information related to human participants, the level of risk, and how this risk will be managed. If your project did not require ethical clearance, you may ignore this section.

This project on Simulating Planetary Motion contains no ethical issues as there are no human or animal participants

7. Data Sharing and Open Access

DATA OWNERSHIP - If you are reusing existing datasets, note down any restrictions the data providers have indicated regarding data sharing. Otherwise, leave blank.

- I have used existing data in my study and I have noted down the relevant restrictions as pertains to data sharing(details below).

I have not used "datasets" per se in this project but the third party data used has been referenced and was publicly available with no explicit license, deeming it fine to use for educational purposes under the "fair use" policy.

DATA LICENCE - Indicate under which licence you intend to share your research data. If you are not sharing your data, provide the appropriate justification as per the UCT Research Data Management guidelines.

- I will not be sharing my data

This project was completed as part of a taught BSc Mathematics degree at University of Bristol and was not funded by a research grant or funded by a formal research project. Therefore, the project is not classified as "publicly funded research data" under open data mandates. This means that there is no obligation to share this project as a public interest, hence why it was not/has not been shared.

DATA PUBLICATION - Indicate where you intend to publish your research data at the end of your project.

I will not be publishing my project.

8. Relevant Institutional or Study Policies

Indicate the relevant departmental, unit, or institutional policies that influence your data management activities.

The data management for this project was guided by the University of Bristol's Research Data Management Policy. Key aspects of the policy relevant to our project are: good data stewardship, sufficient metadata and documentation, data retention and access. We followed these principles by: storing the project on University of Bristol managed One Drive, maintaining clear metadata & documentation within the project, and ensuring each group member retained a local copy.