Project 2: Planet watching in the TRAPPIST-1 System

Project *Comp. Methods Physics* ASU PHY494 (2017)*

March 13, 2017 – March 30, 2017

Abstract Recently it was discovered that multiple planets orbit the ultracool dwarf star TRAPPIST-1 about 40 light years from earth. All of these planets are of a size comparable to earth and three orbit in the habitable zone of their star and are therefore possible candidates for worlds with liquid water and possibly life. Their very tight orbits bring the planets into close vicinity. It should be possible for an observer to stand on one and to see multiple other planets in the night sky so close nearby that continents would be easily discernible with the naked eye. Your task is to *(1) simulate the innermost six planets of TRAPPIST-1* and *(2) analyze the visibility of nearby planets and to decide, which of these planets would be the best one to visit for planet watching*. You will write a *short report* to communicate, discuss and summarize your reasoning and your results. The work is carried out in teams of two or three students.

Due Thursday, March 30, 2017, 11:59pm.

- Students work in teams of two or three students.
- **Admissible Collaboration:** Students are allowed to talk to other students in the class about the project and exchange ideas and tips. However, sharing/copying reports or full code solutions is not allowed. **Help from other students must be acknowledged in an Acknowledgments section**. Direct help from outside the class is not allowed (except instructor/TA), e.g., you cannot ask for solutions (online or in person) but you can use books and resources on the internet to solve problems. **Cite all sources**. Code from the class can be used without explicit citation or acknowledgement.
- Each team should commit their report (see Section 3) in **PDF** format to the team's **GitHub repository**; alternatively, combining report and code in a

*Current version of this document: March 13, 2017. See Appendix 6 for a list of changes since v1 from March 13, 2017.

Jupyter notebook is also possible as long as the notebook can be read like a report (i.e., not just bullet points or short comments). If possible, also generate a PDF from your notebook and commit it together with everything else.

- The report *must* contain a section **Contributions** at the end where the contributions of all team members are summarized.
- Each team should commit and push **all code** (see Section 4) that is required to reproduce the results in the report to their **GitHub repository**. Include a text file `README.txt` that describes the commands to run calculations. The code must run in the standard anaconda-based environment used for the class. If it is a Jupyter notebook then it should be possible to *Kernel* → *Restart & Run All* and to produce all the required figures and output.

Grading will take the following into consideration:

- The code runs and produces correct output.
- The report clearly and succinctly describes the question, approach, and results and contains sufficient evidence that the requirements (see below) have been met.
- All team members contributed to the work: assessed by (1) Contributions section in the report, (2) commit history of the repository and comments in code, (3) short oral examination of team members (at instructor's discretion if deemed necessary).
- Code from outside sources (see Admissible Collaboration) and help is thoroughly attributed (Acknowledgements and References).
- BONUS: Additional work that you want to include in an appendix to the report or additional simulations for the main report will be treated as bonus material and can be awarded bonus points.
- BONUS: Elegant and fast code can be awarded bonus points.

1 Submission instructions

Submission is to your private **team GitHub repository**. Follow the link provided to you by the instructor in order for the repository to be set up: It will have the name *ASU-CompMethodsPhysics-PHY494/project-2-2017*-YourTeamName and will only be visible your team and the instructor/TA. Follow the instructions below to submit this project.

Read the following instructions carefully. Ask if anything is unclear.

1. `git clone` your project repository (change *YourTeamName* to your team's name)

   ```
   repo="project-2-2017-YourTeamName.git"
   git clone https://github.com/ASU-CompMethodsPhysics-PHY494/${repo}
   ```

   or, if you already have done so, `git pull` from within your assignments directory.

2. Create three sub-directories `Submission`, `Grade`, and `Work`.

3. You can try out code in the `Work` directory but you don't have to use it if you don't want to. Your grade with comments will appear in `Grade`.

4. Create your solution in `Submission`. Use Git to `git add` files and `git commit` changes.

   You can create a PDF file or Jupyter notebook inside the `Submission` directory as well as Python code (if required). **Name your files `project02.pdf` or or `project02.ipynb`**, depending on how you format your work. Files with code (if requested) should be named exactly as required in the assignment.

5. When you are ready to submit your solution, do a final `git status` to check that you haven't forgotten anything, commit any uncommitted changes, and `git push` to your GitHub repository. Check on *your* GitHub repository web page[1] that your files were properly submitted.

   You can push more updates up until the deadline. Changes after the deadline will not be taken into account for grading.

Work must be legible and intelligible or may otherwise be returned ungraded with 0 points.

2 Problem description

Recently, at least seven earth-sized exoplanets were discovered orbiting the ultracool dwarf star TRAPPIST-1, about 40 light years away from earth (Gillon et al., 2016, 2017). Using transit measurements of the planets in front of their host star using the Spitzer space telescope and ground-based instruments, orbital parameters for six of these planets (named TRAPPIST-1b to TRAPPIST-1g) could be determined.[2] The planets orbit very close to TRAPPIST-1, at distances between 11×10^{-3} au (astronomical units, average distance earth-sun[3]) to about 45×10^{-3} au (Gillon et al., 2017). TRAPPIST-1 is very cool (effective temperature 2600 K) and small (0.0802 of the solar masses, 0.117 of the solar radius) and hence even in these small orbits at least three planets (TRAPPIST-1e, TRAPPIST-1f, and TRAPPIST-1g) are predicted to be in the habitable zone where liquid water could exist.

[1] `https://github.com/ASU-CompMethodsPhysics-PHY494/project-2-2017-`*`YourTeamName`*

[2] A seventh planet, TRAPPIST-1h, was observed but no orbital parameters could be determined. TRAPPIST-1h will be ignored in this problem.

[3] 1 au is defined to be 149,597,870.700 km.

This discovery has spurred the imagination, with people wondering what it would be like to be on these planets (Figure 1). In particular, the close vicinity of the planets would make it possible to see multiple other planets in the sky. Some of these planets might come so close that one could, in principle, be able to see continents or even smaller landmarks and structures with the naked eye.

In this work you will be simulating the TRAPPIST-1 system and analyze the trajectories of the planets to find out which planet offers the best view on nearby other planets. In particular, use your simulations to decide if the poster in Figure 1 is scientifically accurate: *Will someone on TRAPPIST-1e be able to see five other planets in the sky so that details of their surfaces would in principle be visible with the naked eye?*[4]

2.1 TRAPPIST-1 system

Consider the following for your orbital simulation of the TRAPPIST-1 system.

Interactions The planets interact with the central star and with each other through Newton's law of gravitation.

Parameters and constants Use the orbital parameters from Table 1 in Gillon et al. (2017) (as provided in file `parameters.py` in the dictionaries `parameters.planets` and `parameters.star`) to simulate the TRAPPIST-1 system with the six innermost planets, b–g.

If you use the parameters as given in `parameters.planets` then the units for the planets are as follows: mass m in multiples of the star's mass, orbital semi-major axis a in 10^{-3} au and eccentricity as unit-less number, orbital period P in days; the planetary radius R is given in earth radii. The mass of TRAPPIST-1 in these units is 1. Importantly, the gravitational constant in these units is

$$G = 4\pi^2 \times 0.0802 \times \frac{10^{-3}}{364.25^2}. \tag{1}$$

(and is available as `parameters.G_local`). For other constants see the file `parameters.py`.

Initial conditions The planets move on elliptical orbits with the star at one focus; the ellipses are defined by their major axes a and eccentricities e. All planets move in the same plane. The following Keplerian equations determine their *initial conditions* at aphelion: $r_{\max}$ is the distance of the star to the aphelion (farthest) point of orbit; $r_{\min}$ to perihelion:

$$r_{\max} = a(1 + e) \tag{2}$$

$$r_{\min} = a(1 - e) \tag{3}$$

[4]The poster shows *six* planets but we are ignoring TRAPPIST-1h (which would be the smallest planet in the poster) in this project because we do not have enough data to simulate it.

Figure 1. A poster advertising space tourism in the TRAPPIST-1 system. It shows a hypothetical view from TRAPPIST-1e with six other planets (b, c, d, f, g, and h) visible in the sky. (Copyright ©2017 NASA, from NASA Exoplanet Exploration: Galleries: TRAPPIST-1).

The velocity v_{ap} at the aphelion is

$$v_{\mathrm{ap}} = \sqrt{\frac{GM}{a}\frac{1-e}{1+e}} \tag{4}$$

with M the mass of the star and the velocity vector $\mathbf{v}$ is perpendicular to the radial vector.

For simplicity, assume that at $t = 0$ all planets are at their respective aphelion (i.e., all planets are on a line on the same side of their star).[5]

Motion of the central star You may assume that the central star is fixed at the center of the coordinate system, which is based on the observation that the star's mass is much larger than the planet masses, $M_{\mathrm{star}} \gg m_{\mathrm{planet}}$.

For a BONUS: project (see below) you should relax this assumption and treat the central star as just another body in the simulation. However, in this case you should choose your *initial conditions* such that the *total momentum* of the whole system $\mathbf{P} = \mathbf{M}_{\mathrm{star}}\mathbf{v}_{\mathrm{star}} + \sum_{\mathbf{i=1}}^{\mathbf{6}} \mathbf{m_i v_i}$ is 0, corresponding to the center of mass of the system at rest. This is most easily accomplished by setting $\mathbf{v}_{\mathrm{star}}$ so that $\mathbf{P} = 0$.

2.2 Analysis of accuracy

We want to assess the accuracy of the orbital simulations. You should perform at least the following two tests (see also the Objectives in 5).

2.2.1 Orbital period

Test that the simulation reproduces the observed orbital periods (see values in Table 1 in (Gillon et al., 2017)).

2.2.2 Energy conservation

Newton's equations of motion for time- and velocity-independent potentials conserve the total energy. Test that the total energy of the system

$$E = T + U \tag{5}$$

is conserved, $dE/dt = 0$. T is the kinetic energy and U is the potential energy of the system

$$T = \frac{1}{2}\sum_{i=1}^{N} m_i \mathbf{v}_i^2 \tag{6}$$

$$U = \frac{1}{2}\sum_{i=1}^{N}\sum_{\substack{j=1\\ j\neq i}}^{N} U_{ij}. \tag{7}$$

[5] This assumption is somewhat justified by the observation that the ratios of periods of the planets are all close to ratios of small integers and hence a full conjunction will occur eventually. Such a resonance suggests that all planets formed farther out together and then migrated inwards (Gillon et al., 2017).

Plot the base-10 logarithm of relative error in the total energy

$$\epsilon(t) = \log_{10} \frac{E(t) - E(t=0)}{E(t=0)} \tag{8}$$

to estimate the accuracy of your integration scheme.[6]

2.2.3 BONUS: Momentum conservation

BONUS: If you are also simulating the system with the central star moving freely you know that the center of mass of the total system is initially at rest and thus the total momentum $\mathbf{P} = 0$. No external forces are at work so the total momentum cannot change, $d\mathbf{P}/dt = 0$ and hence the total momentum ought to be zero for *all* time steps. Test that in your simulation the total momentum fulfills $\mathbf{P} = 0$ for all time steps.

2.3 Analysis of "nearby" planets

We want to estimate how many *nearby* planets can be seen from each of the planets in the system at any point in time. From this time series of counts for each planet we can then calculate probabilities $P_i(\nu)$ for seeing nu nearby planets[7], assuming the observer is located on planet i.[8]

Definition of nearby planets We define *nearby* to mean *within a distance d_0 so that features of size 1000 km or greater can be distinguished with the naked eye.*

Resolution of the naked eye The naked eye can typically resolve two different objects if they are at least one arc minute apart[9], i.e., its resolution is

$$\theta_{\text{eye}} = 1' \tag{9}$$

The condition for resolving an object by naked eye is that the object has to appear under an angular separation

$$\alpha \geqslant \theta_{\text{eye}}. \tag{10}$$

The viewing angle α can be computed from the size of the object y (its actual separation) and its distance d from the observer. This allows one to find a maximum distance $d_0 = d(y_{\text{min}})$ so that features of size $y_{\text{min}} = 1000\,\text{km}$ can still be resolved. A *nearby planet* is then a planet within a distance $d \leqslant d_0$.

[6] See `integrators2.py` from class *12 ODEs* for code that you can adapt for this purpose.

[7] ν is the greek letter "nu" and not lower-case V.

[8] As a simplifying assumption, we only consider distances between planets and do not consider where on the planet surface the observer would be located. You are more than welcome to extend the project in this direction for additional bonus points.

[9] One arc minute is 1/60 of a degree: $1' = \frac{1}{60}^\circ$

3 Report

Write a report in which you address the objectives in Section 5 below in the context of the problem (Section 2). The report should contain all results (figures, tables, equations). It must contain a *title*, *author's name*, sections *Background* (problem description, definitions, any equations that you use), *Results and Discussion* (description and interpretation of results), and *Summary* (short summary of the main results).

The report *must* contain a section *Contributions* where you summarize the contributions of each team member. For example, for a team consisting of three members with initials A.B., C.D.E., and F.G., the beginning of this section could be written along the lines of

> A.B. wrote the code to initialize the planet positions and velocities with help from C.D.E. C.D.E. wrote the integration routine, A.B., C.D.E., and F.G. together wrote the simulation function `orbit()`. F.G. with help from A.B. wrote the orbit plotting code and produced figures 1 and 2. F.G. also wrote the Background section ...

(Note that all team members should have commits in the team reporsitory.)

If you had any form of outside help you must describe it in an *Acknowledgments* section.

If you use code or material from elsewhere you *must cite the source* (add a *References* section).

Any bonus material can be shown in an optional *Appendix*.

The report must be written in full sentences and read as a coherent piece of work. Figures must have legends, labels, and captions. Type set in an 11pt font with single line spacing (captions, labels, legend may have smaller font sizes but must still be legible) and leave at least 1 in margins.

Overall, a length of about four pages is expected for a written document produced with a word processor; the report should not be less than three or more than six pages long.

4 Code

For all numerical calculations use Python 3.x. You may use any of the Python packages that are part of the Anaconda 3 distribution such as `numpy` and `matplotlib`.

Include all the code that is needed to generate the results shown in your report. This can consist of Python programs, modules, a Jupyter notebook, or a mixture thereof. Include a separate file `README.txt` that explains how to run your code in order to generate the results in your report. Your code must run in order for you to be awarded full marks.

5 Objectives

Address the following objectives in your report while taking all requirements in Section 2 into account:

(a) Simulate the TRAPPIST-1 system (star and planets b to g) by solving the Newtonian equations of motions using Python code. Your code should produce the positions and velocities of all planets for all time steps.

 (i) Generate initial conditions for the six planets b to g.

 (ii) Plot the orbits for all six planets for a maximum time period of 1.5 d in the x-y plane. Add a marker for the central star to your plot. Confirm that your result is consistent with the known period P_b for planet TRAPPIST-1b (see also Section 2.2.1).

 (iii) Plot the orbits for all six planets for a maximum time period of 1000 d in the x-y plane. Add a marker for the central star.[10] *Use the trajectories from this run for further analysis.*

 (iv) Determine the accuracy of your integration algorithm by plotting as a function of time (1) the total energy $E(t)$ together with kinetic energy T and potential energy U (Eqn. 5–7) and (2) the logarithm of the relative error in the energy $\epsilon(t)$ (Eq. 8).

 Describe and justify your algorithm and your choices of parameters (e.g., your time step). Describe and discuss your figures.

(b) Bonus: Treat the central star not as fixed but as another body in the simulation.

 (i) Plot the path of the star over 1000 d and compare the extent to the star's diameter. Is the center of mass of the whole system inside the star?

 (ii) Analyze the "wobble" of the star (motion due to gravity of the planets). Assume an observer looks side on the system along the y-direction. Compute a time series of the y-component of the velocity of the star. Compare the changes to the resolution of current Doppler spectrometers (about 1 m/s). Would the wobble be detectable?

(c) Using the positions of all planets $\mathbf{r}_i(t)$ for $0 \leqslant t \leqslant 1000\,\mathrm{d}$, find the planets where an observer is most likely to see many other planets *nearby* (as defined in Section 2.3).

 (i) Given the resolution of the naked eye, Eq. 9, determine the maximum distance d_0 for two planets so that an observer could resolve large-scale landscape features of size $y \geqslant 1000\,\mathrm{km}$.

 (ii) Analyze $\mathbf{r}_i(t)$ to find for each time step the number of planets $\nu_i(t)$ that are nearby to planet i (excluding the planet itself). Plot $\nu_i(t)$ for $i = 4$ (TRAPPIST-1e) (and other planets if necessary for your discussion).

[10] Bonus: analyse the time series of the angular motions of the planets to show that their computed periods P_i match the experimentally observed ones (see Section 2.2.1).

(iii) For each planet i, calculate the probability over the 1000 d observation time to have exactly ν other planets nearby, $P_i(\nu)$.[11]

Plot the probabilities $P_i(\nu)$ for all six planets.

(d) Using your results, answer the question if the poster in Figure 1 is scientifically accurate: *Will someone on TRAPPIST-1e be able to see five other planets in the sky so that details of their surfaces would in principle be visible with the naked eye?*

Which would be the best planets in the TRAPPIST-1 system for "planet watching"?

6 History

Changes and updates to this document.

2017-03-13 initial version

2017-03-13 updated: accuracy analysis added

References

Gillon M, Jehin E, Lederer S M, Delrez L, de Wit J, Burdanov A, Van Grootel V, Burgasser A J, Triaud A H M J, Opitom C, Demory B O, Sahu D K, Bardalez Gagliuffi D, Magain P and Queloz D, 2016 Temperate earth-sized planets transiting a nearby ultracool dwarf star. *Nature* **533** 221–224, http://dx.doi.org/10.1038/nature17448.

Gillon M, Triaud A H M J, Demory B O, Jehin E, Agol E, Deck K M, Lederer S M, de Wit J, Burdanov A, Ingalls J G, Bolmont E, Leconte J, Raymond S N, Selsis F, Turbet M, Barkaoui K, Burgasser A, Burleigh M R, Carey S J, Chaushev A, Copperwheat C M, Delrez L, Fernandes C S, Holdsworth D L, Kotze E J, Van Grootel V, Almleaky Y, Benkhaldoun Z, Magain P and Queloz D, 2017 Seven temperate terrestrial planets around the nearby ultracool dwarf star TRAPPIST-1. *Nature* **542** 456–460, http://dx.doi.org/10.1038/nature21360.

[11] You may perform these calculations in any way that you like. You might, however, find it useful to look at the NumPy `numpy.histogram()` and/or Matplotlib `matplotlib.pyplot.hist()` functions.