# AST 51 Assignment #2 - Eclipsing Binary Star Populations

**Task:** Create a Jupyter notebook tutorial for generating a population of binary stars according to simple but realistic distributions (given below). Implement **either** inverse transform sampling **or** rejection/acceptance sampling (or both). From this, determine a population of eclipsing binary stars and identify/discuss differences with the eclipsing population
**Professor:** David V Martin
**Grade amount:** 15% of final grade
**Given**: Thursday Feb 6th, 2025
**Due date**: Thursday Feb 27nd, 2025 @ 6pm
**Early reward:** 3% per day for max 2 days, applied multiplicatively (max 100%)
**Late penalty:** 3% per day with no maximum, applied multiplicatively
**Marking metric:** Out of 100%

- Clear pedagogy including clearly written English **(25%)**
- Population synthesis is done correctly **(40%** total**)**
  - Masses (**10%**)
  - Radii (**5%**)
  - Orbital size (**5%**)
  - Eccentricity (**10%**)
  - 3D orientation (**10%**)
- Eclipsing population is thoroughly and correctly analyzed **(25%)**
- Code is well commented and explained **(10%)**

| | |
|---|---|
| 1 | Mahadevan, Pratt, Nguyen |
| 2 | Lamardo, Lam, Reeves |
| 3 | Vidal, Banks, Finger |
| 4 | Kropiwnicki, Woodruff, Pandina |
| 5 | Carson, Robinson, Mengkeat |
| 6 | Spejcher, English, Truex |
| 7 | Sibley, Dahleen, Kleanthous |
| 8 | Broni, Edelman, Lindsay |
| 9 | Zapanta, Zlotnitsky, Birn |
| 10 | Donavan, Dy, Zapanta |

## Distributions

**Primary mass:** Salpeter 1955 "Initial Mass Function" (see **https://en.wikipedia.org/wiki/Initial_mass_function**). Range between 0.1 and 3 solar mass
**Mass ratio:** $q = M_B / M_A$ is uniform between 0.1 and 1
**Radius:** $R/R_\odot = (M/M_\odot)^\xi$ where $\xi = 0.53$ for $M > 1\ M_\odot$ and $\xi = 0.8$ for $M < 1\ M_\odot$
**Period**: Log-normal with mean $\mu(\log P) = 5.03$ and standard deviation $\sigma$ (logP) = 2.28, both in days (in log space, i.e. the mean is $10^{4.8}$ days ~ 170 years)
**Eccentricity:** Halbwachs+ (2003) Fig. 11 or Raghavan+ (2010) Fig. 14
**3D Orientation:** Isotropic: $\omega$ is uniform between 0 and 360 degrees and cos(I) is uniform between -1 and 1 (you will have to explain in your tutorial why cosine of the inclination and not the inclination itself is uniform). You can ignore $\Omega$ (or set it to be uniform between 0 and 360) because it does not affect eclipses. You will have to explain **why** you can ignore $\Omega$ though.
**Eclipses:** To obtain full marks you need to account for eccentric orbits and test for both primary eclipses (star B in front of star A) and secondary eclipses (star A in front of star B), where A = primary star (more massive) and B = secondary star (less massive). You can assume circular orbits and do the analysis with this simplification, but you will not receive full marks.

**I have created a Jupyter notebook "3D orbit geometry" which you can find on Canvas that will help you significantly with this!**

**References (all on Canvas under "Resources")**

**Duquonnoy and Mayor 1991**
**Halbwachs et al. 2003**
**Raghavan et al. 2010**
**Murray and Dermott "Solar System Dynamics" Chapter 2**
**AST-191 lectures 1-3**

## FAQ

**Who is the target audience of this notebook?** Other astrophysics undergrads or grad students. You can assume they have some basic knowledge of Python, random variables and stellar binaries, but have not done this task before. If unsure, err on the side of providing more explanation than less.
**What do we hand in?** Each group will hand in **ONE** tutorial notebook. All tutorial text will be contained with the notebook.
**Am I in the same group as for assignment 1?** The groups have changed

**The end goal seems rather vague?** Yep, like in research, there is an element of you setting your own goal, rather than having something very clearly defined for you. Ultimately you should produce something that has interesting (and correct results) and demonstrates how the distribution of binaries changes when you go from a general population of binaries in the galaxy to a population of eclipsing binaries. Your project should contain lots of pretty plots and interesting commentary on how the distribution looks and changes with the eclipse requirement. There is no set minimum or maximum length.

**I'm still confused about the end goal?** That's fine, ask a question in Slack or come to office hours!

**Do I care about primary eclipses or secondary eclipses or both?** Both. This is particularly important when considering eccentricity.

**How do I sample the eccentricity if you've just given a figure as a reference?** Read the papers a bit about the eccentricity distribution. Very close binaries tend to be circular due to tides. Wider binaries can be eccentric. Be creative and come up with a simple function that seems to get the main picture of these figures. You are free to eyeball it or you could take the data from the distribution and randomly sample directly from that. Either is fine! This is a chance to be a bit creative.

**What do you mean I can ignore $\Omega$?** Play around with the notebook I've provided you. You'll see that $\Omega$ has no effect on the eclipses. You'll have to explain why though.

**Can I copy code from this workbook?** You can take bits and pieces as long as it is well explained (and not just a copy of my simple explanations).

**Do I need to implement both inverse transform and acceptance/rejection sampling?** You can do either or both. These codes have to be written from scratch, not just from a package (although you can use a package to check your results if you like).

**Do I need to read all of the supplemental reading?** No, and indeed those papers are fairly long. Murray and Dermott Chapter 2 will really help (particularly 2.1, 2.2, 2.3, 2.4 and 2.8). The research papers are the basis of the distributions I gave you so they are really for if you want some context. They may also help your tutorial have some nice astrophysics context to it.

**How is this graded?** Everyone in a group will receive the same mark, barring cases of insufficient contribution

**What if I fall ill or other unforeseen circumstances?** Communicate with me! I am here to support you.

**How is plagiarism and insufficient contribution handled?** Every student must hand in a signed (e-signature is fine) document where in a couple of sentences they state their original contribution to the project and a rough assessment of the distribution of work amongst the group. This will be anonymous. This document is on Canvas. In addition, several times throughout the year students will be randomly interviewed to discuss their assignment with the TA and to demonstrate their knowledge of the code's working.

**Am I expected to understand the entire submission?** Yes, you should understand the entire submission, even if you largely worked on a certain section of it.

**What is the punishment for insufficient contribution?** If it is believe you have made an insufficient contribution to the extent that you do not properly understand the submitted assignment, you may be asked to do either an additional coding task (to reinforce the lessons of the assignment) or, in extreme cases, asked to do the assignment again on your own. Ultimately, we are all adults and will be treated as such. But we also want to have assessment that is fair and with integrity. Violations could result in reduced individual or group marks or other disciplinary action.

**How do we hand this in? ALL** students must hand in the aforementioned "integrity form" but only one student per group (could be any) needs to hand in the actual assignment. If multiple students in a group hand in the assignment I'll consider the first one, unless otherwise told. All of this will be on Canvas.

**How are late/early marks applied?** They are applied multiplicatively. If your base score is 70% but you handed it in 2 days early, then you'd receive 70 * 1.06 = 74.2%. If you handed it in 3 days late then you'd receive 70 * 0.91 = 63.7%. In terms of timing, if you hand in within 24 hours of the due date, that is considered "on time", i.e. no early or late marks. Between 24 and 48 hours before the due date = 1 day early, so a 3% early award, and so on… Within 24 hours after the due date = 1 day late, 24-48 hours = 2 days, etc…

**Can I ask the teacher/TA/rest of the class for help?** Absolutely. Whilst your first port of call should be to discuss/ask your teammates, you are always encouraged to come to office hours and post on Slack. We won't do the assignment for you but we'll happily assist you

**Can I program using a different language?** No, using python 3 is required as it is an astronomy "industry standard" that we'd all do well to learn, even if you already know another language.

**Can we include downloaded packages?** The fundamentals of the assignment have to done "from scratch". I.e. you can use some basic numpy packages but you have to write the code to do the random sampling. For many of the aspects of this assignment there exist packages out there (e.g. randomly sampling from an IMF). You are welcome to use these in order to check the validity of your work, and that can be included in your tutorial for that purpose, but ultimately you have to do the work yourself.

**Can we use code from the assignment 1 pretty plots knowledge base?** Yes, indeed the purpose of that assignment was to provide pretty plot codes that you can use! However, you still have to comment and clearly document everything that you use.

**Should the Jupyter notebook run?** Yes, it should run on Python 3.8 (or another version of python 3 as long as it is clearly specified). If you require any packages, it should be clear what needs to be downloaded.

**Can I turn in a .py script instead of a Jupyter notebook?** No, whilst the use of .py scripts is common, for consistency of grading and the fact that Jupyter notebooks are worth learning, we require them. Jupyter notebooks are also very popular for tutorials, like the one you are writing

**Can I incorporate code that I found online?** Realistically, astronomers use StackExchange, Github and in general Google a lot. This is fine in moderation, as long as it is documented. Ultimately, to obtain full marks the code submitted should illustrate that you a) know how to write sophisticated code and b) know what each line of the code is doing.

**Can I use ChatGPT?** As a tool that helps you learn, e.g. finding bugs in your code, sure. However, you cannot use ChatGPT to write the whole assignment (it'll be pretty obvious and a

pretty poor result). Remember that the coding part of this assignment is only one aspect; many marks are awarded for the pedagogy.
**Can we be inspired by tutorials we see online?** Inspired, for sure! Copied, no.