

Detecting Exoplanets Through Transits

Kalliope Luna Welch

Introduction

The shape of a planet crossing over its star will lessen the amount of light waves that reach Earth from the star, thus making the star “dimmer.” Professional astrophysicists have used this technique to help them discover many of today’s known exoplanets, or planets from another planetary system.[1, 2] I propose to use Machine Learning[3] to search data put forth by NASA Citizen Scientist Programs[4, 5] for any signs of a star suddenly dimming in brightness for a small period of time and then returning to its normal amplitude of light, an occurrence called a light curve.[6] The most important factor is the structure of the planet system itself; the relative sizes of the planet and star, as well as their distance apart, affect the resulting light curve.[7] Finally, there is the structure of my neural network. How this tool is built and how much training it is given will affect the result of its performance nearly as much as the the structure of the planet system. To train my neural network, I will generate model light curves.

Model

I am using the Mandel and Agol equations to generate model light curves:[8]

$$F^e(p, z) = 1 - \lambda^e(p, z) \tag{1}$$

In this equation, $F^e(p, z)$ is the star’s intensity relative to its unobscured intensity. p and z are the parameters of λ . p represents the size of the planet ($r_p =$ planet radius) as a ratio to the size of the star ($r_* =$ star radius): $p = r_p/r_*$. z represents the distance (d) between the two: $z = d/r_*$. From Mandel and Agol, we have:

if $1 + p < z$:

$$\lambda^e(p, z) = 0 \tag{2}$$

if $|1 - p| < z \leq 1 + p$:

$$\lambda^e(p, z) = \frac{1}{\pi} \left[p^2 \kappa_0 + \kappa_1 - \frac{1}{2} \sqrt{4z^2 - (1 + z^2 - p^2)^2} \right] \tag{3}$$

where

$$\kappa_0 = \cos^{-1}[(p^2 + z^2 - 1)/2pz] \tag{4}$$

and

$$\kappa_1 = \cos^{-1}[(1 - p^2 + z^2)/2z] \quad (5)$$

if $z \leq 1 - p$:

$$\lambda^e(p, z) = p^2 \quad (6)$$

if $z \leq p - 1$:

$$\lambda^e(p, z) = 1 \quad (7)$$

Figure 1 displays a cartoon of the planet transiting its home star, while Figure 2 shows the resulting light curve. The light curve was generated by adding artificial computer noise to the model above. The noise renders the light curve in a more realistic fashion. Real light curves are convoluted, as there are many sources of interference in between the analyzed star and the detecting telescope.

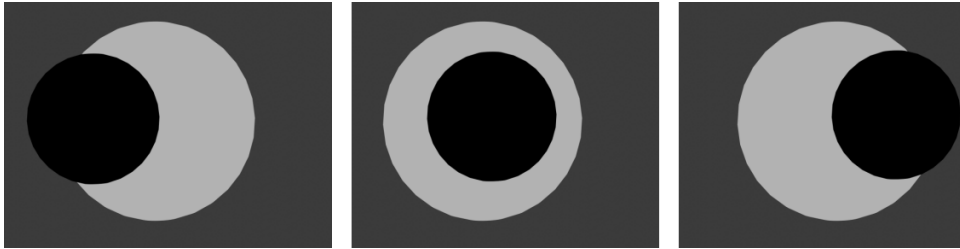


Figure 1: A planet crossing in front of a star.

Implementation

I will use my model data to train a Pytorch neural network, following the Pytorch tutorial that uses the FashionMNIST data set[9]. I will test the performance of the neural network with existing transit data for known exoplanets. Then I will use the neural network to analyze data collected from NASA's Exoplanet Watch[5, 6, 10, 11, 12, 13] and (perhaps) my own telescope in order to detect *unknown* exoplanets.

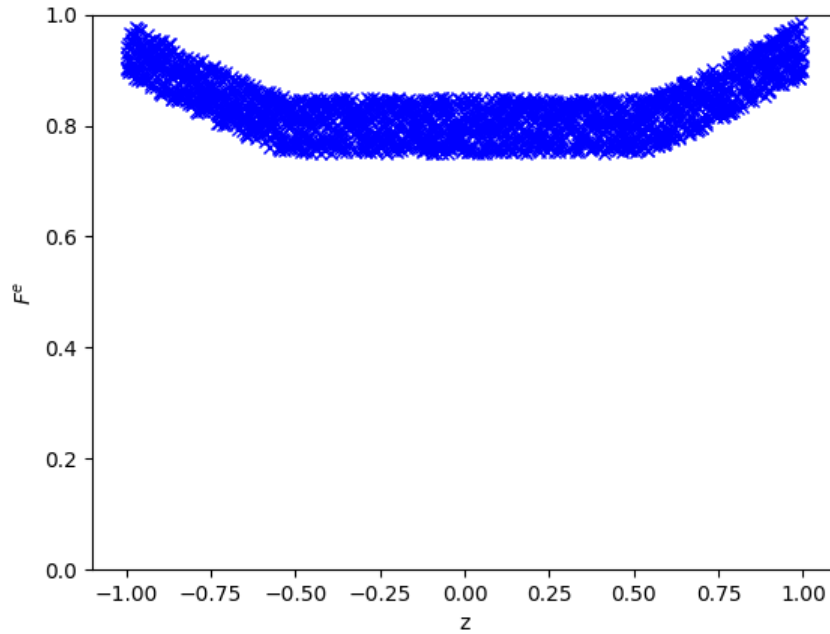


Figure 2: The simulated light curve.

My code for the simulations is given below. Figure 2 was made with Matplotlib[14].

```
import numpy as np

def K0(p, z):
    term1 = p*p + z*z - 1
    term2 = 2*p*z
    return np.arccos(term1 / term2)

def K1(p, z):
    term1 = 1 - p*p + z*z
    term2 = 2*z
    return np.arccos(term1 / term2)

def lambda_e(p, z):
```

```

    if 1+p < z:
        return 0
    if (np.abs(1-p) < z) and (z <= 1 + p):
        term1 = p*p*K0(p, z)
        term2 = K1(p, z)
        term3 = 0.5*np.sqrt(4*z*z - (1 + z*z - p*p)**2)
        return 1/np.pi * (term1 + term2 - term3)

    if z <=1-p:
        return p*p
    if z <=p-1:
        return 1

    return -1000

def Fe(p, z):
    return 1 - lambda_e(p, z)

#Example
p = 0.5
Nb = 2001
I = np.zeros(Nb, dtype=float)
zboxes = np.zeros(Nb, dtype=float)
dz = 2.0/(Nb-1)
noise_f = 0.1
for i in range (0, Nb):
    z = i*dz - 1
    zboxes[i] = z
    z = np.abs(z)
    I[i] = Fe(p, z) + np.random.random()*noise_f

```

References

- [1] Winn, Joshua N. “The Search for Exoplanets: What Astronomers Know” *The Great Courses* (2015).

- [2] Winn, Joshua N. “An Introduction to Astrophysics” *The Great Courses* (2015).
- [3] Chintarungruangchai, P.; Jiang, I. “Detecting Exoplanets through Machine-learning Techniques with Convolutional Neural Networks,” *Publications of the Astronomical Society of the Pacific* (2019).
- [4] “Citizen Science – Exoplanet Exploration: Planets Beyond our Solar System,” <https://exoplanets.nasa.gov/citizen-science/>, Accessed Nov. 10, 2023.
- [5] “Overview / What is Exoplanet Watch? – Exoplanet Exploration: Planets Beyond our Solar System,” <https://exoplanets.nasa.gov/exoplanet-watch/about-exoplanet-watch/overview/> , Accessed Nov. 10, 2023.
- [6] “Background Information / What is Exoplanet Watch? – Exoplanet Exploration: Planets Beyond our Solar System,” <https://exoplanets.nasa.gov/exoplanet-watch/about-exoplanet-watch/background/> , Accessed Nov. 10, 2023.
- [7] “Exoplanets,” <https://afh.sonoma.edu/exoplanets/> , Accessed Nov. 10, 2023.
- [8] Mandel, K.; Agol, E. “Analytic Lightcurves for Planetary Transit Searches,” arXiv: astro-ph/0210099v1, (3 Oct 2002).
- [9] “Quickstart-Pytorch Tutorial,” https://pytorch.org/tutorials/beginner/basics/quickstart_tutorial.html , Accessed Oct. 13, 2023.
- [10] “Request an Exoplanet Observation / How to Participate – Exoplanet Exploration: Planets Beyond our Solar System,” <https://exoplanets.nasa.gov/exoplanet-watch/how-tocontribute/data-checkout/> , Accessed Nov. 10, 2023.
- [11] “How to Analyze Your Data / How to Participate – Exoplanet Exploration: Planets Beyond our Solar System,” <https://exoplanets.nasa.gov/exoplanet-watch/how-to-contribute/how-to-analyze-your-data/> , Accessed Nov. 10, 2023.

- [12] “Resources / Exoplanet Exploration: Planets Beyond our Solar System,” <https://exoplanets.nasa.gov/exoplanet-watch/resources/everything-exo/> , Accessed Nov. 10, 2023.
- [13] “Get Involved / How to Participate – Exoplanet Exploration: Planets Beyond our Solar System,” <https://exoplanets.nasa.gov/exoplanet-watch/how-to-contribute/checklist/>, Accessed Nov. 10, 2023.
- [14] Hunter, J. D. “Matplotlib: A 2D graphics environment,” *Computing in Science & Engineering* 9, 3, <https://doi.org/10.1109/MCSE.2007.55> (2007), 90-95.