

Detecting Exoplanets Through Transits

New Mexico

Supercomputing Challenge

Final Report

April 10, 2024

Welch Homeschool

Team Members:

Kalliope Luna Welch

Teacher:

Cindy Welch

Project Mentor:

Paul Welch

Executive Summary

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.” This crossing of the planet in front of its star, similar to an eclipse, is called a transit. Professional astrophysicists have used this phenomenon to help them discover many of today’s known exoplanets, or planets from another planetary system.[1, 2] The aim of this project was to train and test a neural network (a machine learning tool) to find exoplanets through data analysis of transits. As a conclusion to this project, I discovered the different parameters that affect the accuracy of a neural network and even detected a few (prediscovered) exoplanets myself.

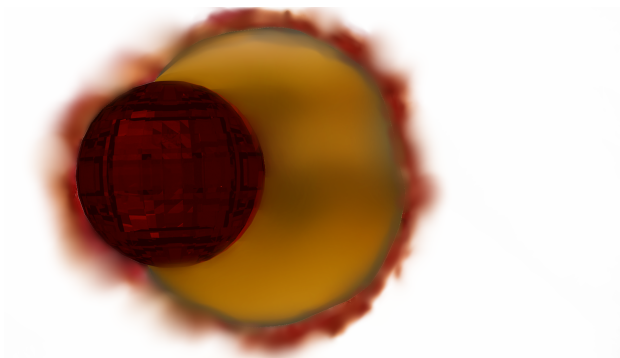


Figure 1: Model of a planetary transit.

Introduction

In 1984, the first proof of an exoplanetary object was found when the Las Campanas Observatory took a picture of Beta Pictoris, a star that had a planetary disk of ice and dust surrounding it in orbit. Not long afterwards, in 1990, the Hubble Space Telescope was launched to find more such phenomena, and in 1995, Didier Queloz and Michael Mayor discovered 51 Pegasi, the first exoplanet. In 1999, HD 209458, another exoplanet, was found through the transit method that I used in this project. I used a machine learning tool called a neural network[3] to search various star data sets 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; the result of this occurrence in a set of data is called a lightcurve. Figure 2 shows a model lightcurve, the result of a transiting exoplanet.

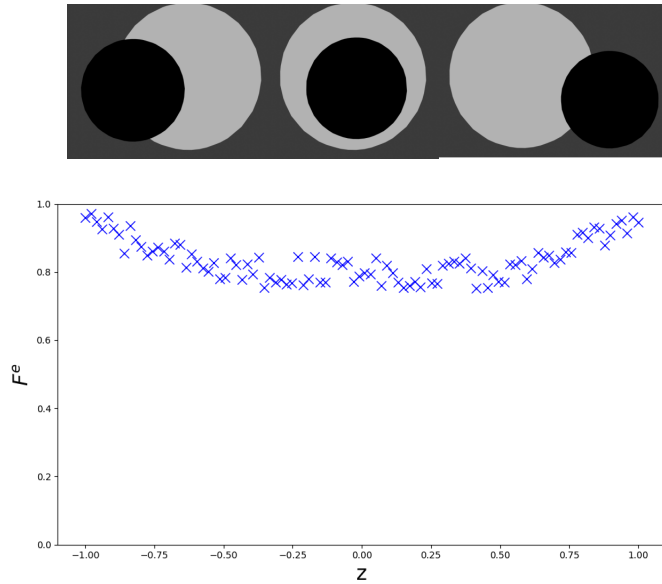


Figure 2: Diagram of a planet transiting its star (top) and the resulting model lightcurve (bottom).

A neural network was considered the best tool for this project because it will analyze the data for you, rather than forcing you to do so manually; it is therefore quicker and enables you to analyze more data than would a manual method.

Problem Statement

In this project, my goal was to discover if a neural network can detect transiting exoplanets, given certain factors. In particular, I investigated the following questions:

- 1) How does noise (objects passing in between the star and the telescope, as well as the telescope's abilities to gather data) affect network performance?
- 2) How does the size of the planet compared to the size of the star affect network performance?
- 3) How does the structure of the network affect its own performance?

Process

Model

My approach to this project was to first create model data, then use different datasets to

train and test my neural network, and finally test my network with real data, collected from real exoplanets. To create my model data, I used the Mandel and Agol equations (Equations 1-7). In this set of equations, F_e is the fraction of light that reaches the telescope as a function of p and z ; p is the ratio of the size of the planet to the size of the star, and z is the distance between the centers of the star and planet. λ_e is the amount of light obscured by the planet crossing in front of the star. I is the overall light intensity.

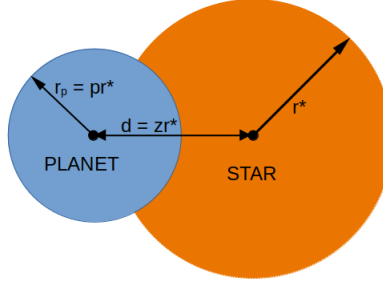


Figure 3

$$F_e(p, z) = 1 - \lambda_e(p, z) \quad (1)$$

If $1 + p < z$:

$$\lambda_e(p, z) = 0 \quad (2)$$

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

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

where

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

and

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

If $z \leq 1 - p$:

$$\lambda_e(p, z) = p^2 \tag{6}$$

If $z \leq p - 1$:

$$\lambda_e(p, z) = 1 \tag{7}$$

The lightcurve in Figure 2 was generated by adding artificial computer noise to the model above (Equation 8). The noise renders the lightcurve in a more realistic fashion. Real lightcurves are convoluted, as there are many sources of interference in between the analyzed star and the detecting telescope.

$$I = F_e(p, z) + \textit{RandomNoise} \tag{8}$$

Neural Network

Neural networks are made of several sets of hidden layers, as is shown in Figure 4. Each hidden layer contains a number of nodes, which make up the height of the hidden layers, a major part of my project.

To create the training set, I did the following:

- a) set the input parameters, including the number of datapoints per lightcurve and the number of lightcurves;
- b) selected random values for p and noise;
- c) used the model to generate lightcurve data.

If the data was a true transit, I would set a flag to mark the data as 'true transit'; on the other hand, if it was not a true transit, I would set a flag to mark the data as 'noise'. In both circumstances, the next step would be to write the data to a file.

To train my Pytorch neural network, I followed the procedure below:

- a) defined my neural network, i.e., the number and the height of the hidden layers;
- b) inputted my training dataset;
- c) fed my data into the network's input layer;

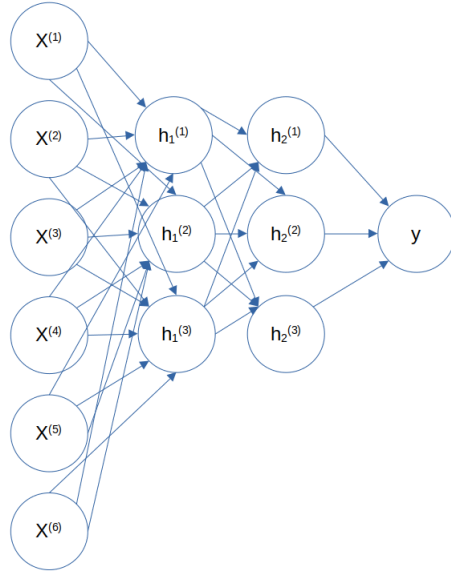


Figure 4

d) passed the data through each hidden layer;

e) obtained an estimate (i.e., does the data contain a lightcurve or not?).

If the network calculated an error across the set, it would backpropagate and optimize the parameters until the answer was correct.

Finally, I tested my neural network with real data collected from actual exoplanets, which I gathered from the AAVSO, or the American Association of Variable Star Observers.[14] Figure 5 is an example of a lightcurve covered in interfering noise. Can you tell the difference?

Results and Significant Achievements

First, I tested my neural network and made three plots of my network's accuracy based on certain parameters. Figure 6 shows that increasing the height of the network's hidden layers also increases its accuracy, up to a certain point where the accuracy stabilizes. Figure 7 shows that a higher noise factor creates a lower accuracy. One axis of Figure 8 is the accuracy of the network, another is the planet to star size ratio (p), and the third is the noise factor. In this example, my network's hidden layers had a height of 20; it was trained with 100,000 training points. Although the training sets mostly focused around $p \leq 0.2$, it was able to detect planets with larger p values, as well. The amount of noise heavily

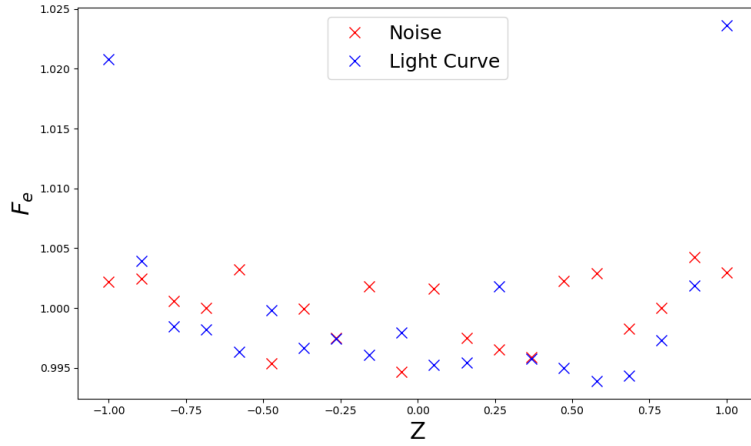


Figure 5

impacted the network's overall accuracy; as in Figure 7, increasing the noise value decreased the network's accuracy by quite a bit.

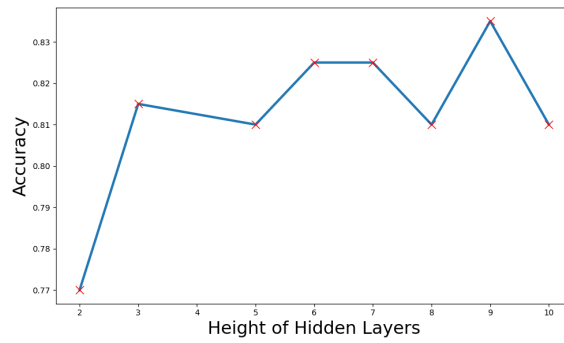


Figure 6

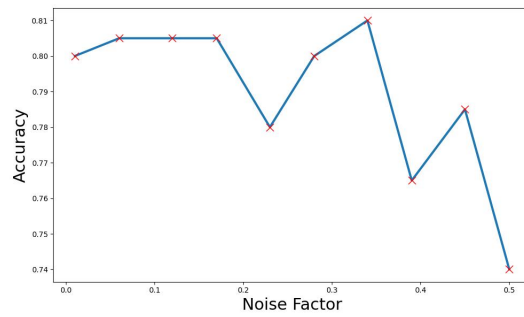


Figure 7

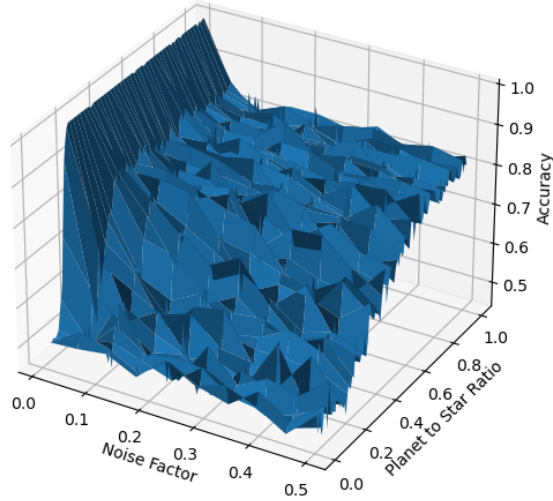


Figure 8

Figure	Height of hidden layers	Noise Factor	p
6	2-10	0.01-0.5	0.1-1
7	3	0.01-0.5	0.1-1
8	20	0.01-0.5	0.1-1

Table 1: Neural network test data parameters.

After testing my neural network, I used it to detect four real (but pre-discovered) exoplanets named TrES-2b, HAT-p-68-b, Qatar 9-b, and TOI 3757. To detect each of these planets, I had to go through a process called smoothing, in which I took the running average of a certain number of datapoints from the original lightcurves. After smoothing, I had to rescale the time axis of each lightcurve to fit the scale used in training my network. I then fit the data to a spline and sampled it, taking an estimate from the smoothed and rescaled lightcurve to give to the network. Figures 9 - 12 show four sets of lightcurves for each of the four exoplanets. Each set shows the original lightcurve, followed by the result of the smoothing and rescaling process. These exoplanets were only detected by my network after the smoothing process, showing what an important part of the procedure this is.

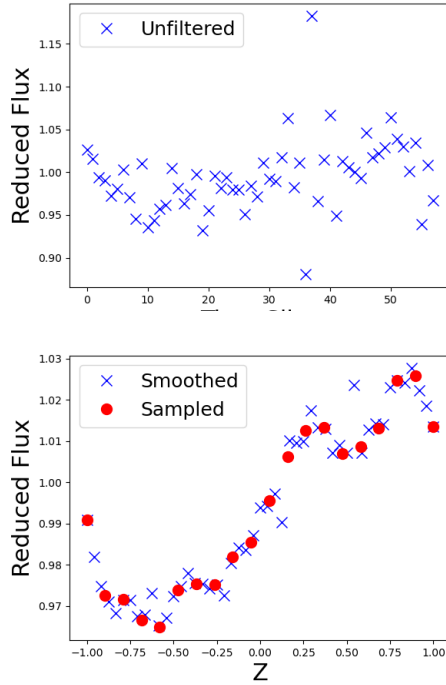


Figure 9: TrES-2b before (top) and and after smoothing (bottom).

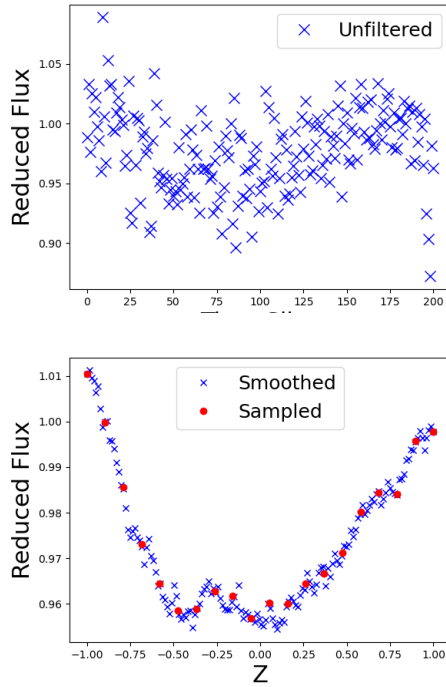


Figure 10: HAT-p-68-b before (top) and after smoothing (bottom).

As shown in Table 2, the detected exoplanets ranged from being 1.36 times bigger than Jupiter to being 1.009 bigger than Jupiter. They were discovered in time periods that ranged

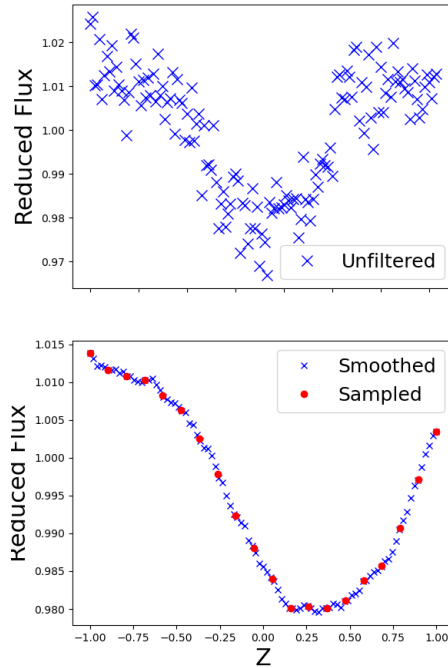


Figure 11: Qatar 9 b before (top) and after smoothing (bottom).

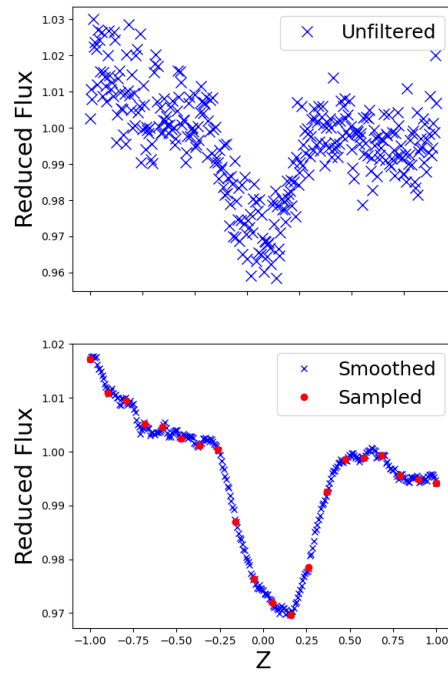


Figure 12: TOI 3757 b before (top) and after smoothing (bottom).

from 2006 to only recently in 2022. They ranged from being 591 to 702 light years from Earth. These are real exoplanets that my neural network has successfully detected. Of course, there were also some exoplanets that my neural network did NOT detect, even after

Exoplanet	Size compared to Jupiter	Distance from Earth (light years)	Discovery	Detected by NN?
TrES-2b	1.36 x	702	2006, transit	only with smoothing
HAT-p-68-b	1.072 x	659	2020, transit	only with smoothing
HD 189733 b	1.13 x	64	2005, radial velocity	not detected
Qatar 9 b	1.009 x	697	2019, transit	only with smoothing
TrES-3b	1.336 x	754	2007, transit	not detected
TOI-270 c	0.21 x	73	2019, transit	not detected
TOI-3757 b	1.071 x	591	2022, transit	only with smoothing
WASP-12 b	1.937 x	1,393	2008, transit	not detected

Table 2: Exoplanet information.

the smoothing process. These exoplanets are named Wasp-12 b, TrEs-3b, TOI 270, and HD 189733 b; their lightcurves are shown in Figures 13-16.

Figure 13 shows the lightcurve of Wasp-12 b. This is a significant planet, 1.973 times bigger than Jupiter, 1,393 light years away from Earth, discovered in 2008, and literally being torn apart by its star's gravity. I was, therefore, very disappointed by the fact that my neural network did not detect this exoplanet. This is probably due to the fact that the data shows only a fraction of the lightcurve, whereas my network was trained to detect full curves. The same is true for Figure 14 (HD 189733 b, 1.13 times the size of Jupiter, 64 light years from Earth, discovered 2005) and Figure 15 (TrES 3-b, 1.336 times bigger than Jupiter, 754 light years, discovered 2007.) As for Figure 16, TOI 270-c, I infer that a cloud or other object crossed in front of the telescope during the observation of this exoplanet, and thus made the curve strangely marred.

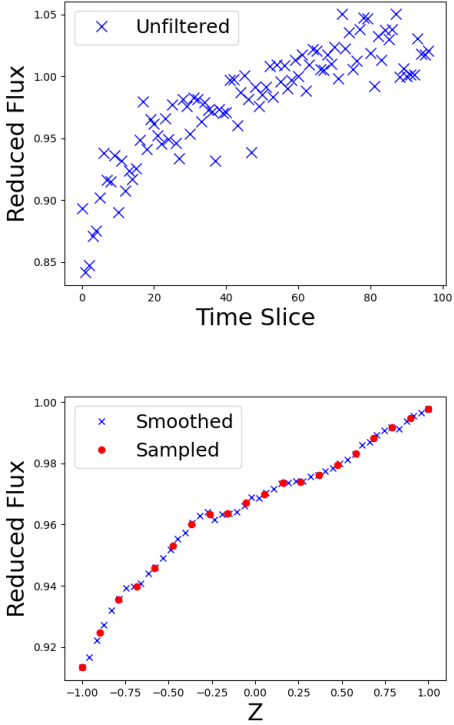


Figure 13: Wasp-12 b before (top) and after smoothing (bottom).

In this project, I have concluded how noise, p , and the structure of the network itself affect its performance. I have detected real (albeit prediscovered) lightcurves from planets outside of our solar system and tested the limits of what I could not detect. But most importantly, I have now tested and trained a tool that can discover real exoplanets.

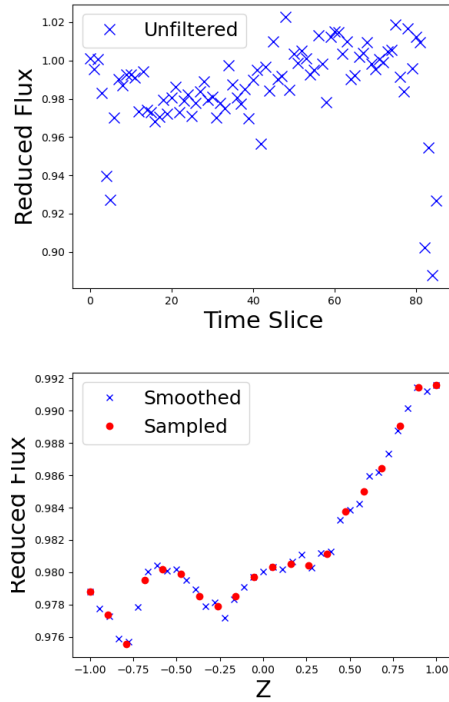


Figure 14: HD189733 before (top) and after smoothing (bottom).

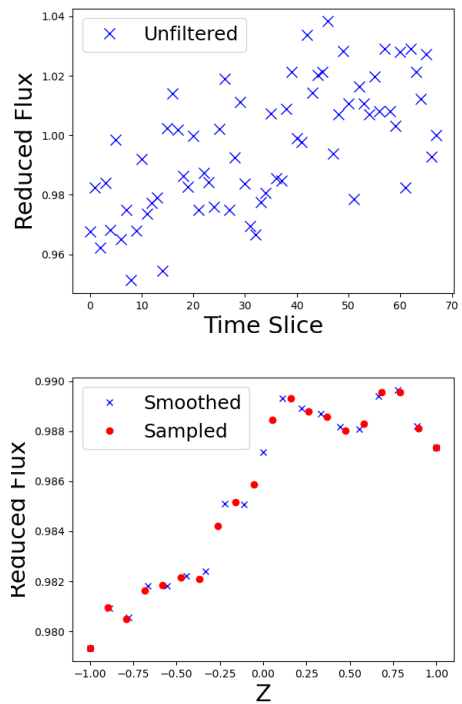


Figure 15: TrES-3 b before (top) and after smoothing (bottom).

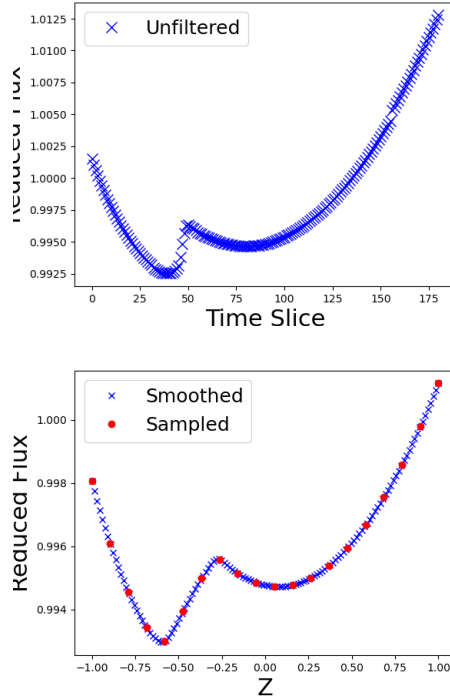


Figure 16: TOI 270-c before (top) and after smoothing (bottom).

Conclusions

From this project I was able to conclude that a neural network's ability to detect exoplanets depends on the planet system and the neural network's structure. Increasing noise decreases accuracy, whereas increasing the height of the hidden layers increases accuracy, up to a point. Larger planets will give higher network accuracy, as does smoothing the data. In the future, I hope to test my network with additional real data and find an undiscovered exoplanet using my neural network.

Appendix

My code for generating model lightcurve data is given below. All plots were made with Matplotlib[15].

```
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

```

Acknowledgements

I would like to thank judges and sponsors of the Annual Supercomputing Challenge for their helpful advice; I would also like to thank the the judges of the local and regional NENM Science Fair for useful feedback.

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-to-contribute/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] Lightcurve data for known transiting exoplanets were obtained from the AAVSO International Database: “AAVSO / Exoplanet Database Search,” <https://app.aavso.org/exosite/>, Accessed Mar. 10, 2024
- [15] 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.