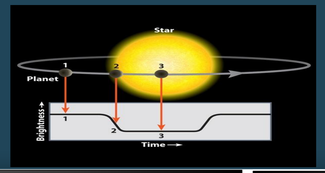
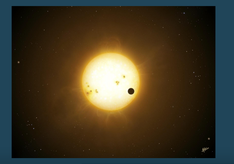


Simulation of a Transit Method for Detection of Exoplanets

Sarvesh Bhogaokar[1], Tejas Sonawane[1], Jatin Rathod[2]

[1]St. Xavier's College, Autonomous, Mumbai - 400001, India.
[2]Nehru Planetarium, Mumbai - 400018, India.



IWARA
From Quarks to Cosmos

ABSTRACT

We present the simulation of a transit method for detection of exoplanets. Our aim is to explore the easier approach to teach this concept to Undergrad and advanced high schools students and also to give them hands-on experience. For simplicity, here we only consider one planet around the target star. We simulate the star-planet system using 'Vpython' and analyse it through 'Tracker' to produce light curves and the respective data file in terms of Time and Flux. Light Curves and the Data files are analysed by Tracker and Python algorithms. We then estimate the Radius of the exoplanet and orbital radius using observables such as transit depth, transit duration, and Orbital period. Finally, these values are compared with the input parameters of the Vpython simulation to calculate the errors and to state the conclusion.

INTRODUCTION

- Transit method is one of the most effective and reliable methods to detect the exoplanets around other stars in our galaxy.
- As an exoplanet passes in front of its host star along our line of sight, it causes a periodic dimming of the system's brightness. This follows as the exoplanet blocks a portion of the host star's radiant flux. Consequently, light curves of the system are generated, which are then studied in search of periodic dips corresponding to planetary transits.
- Students only get to read about the theory part of the concept but some of them lack visualization and most importantly the hands-on experience.
- The aim of this paper is just to fill this gap where we attempt to simulate a stellar system with one planet using open source softwares - VPython & Tracker
- Students can see the system in VPython for visualization and analyse it through Tracker for understanding the lightcurves and also to generate data files. These files generated can then be used for further analysis using Python.
- This provides an opportunity for students to apply their programming skills for deriving the observables and calculating parameters using python.
- Students will also get to use various packages like astropy, scipy, matplotlib and pandas to improve their data analysis skills.

METHODOLOGY

VPython

Simulate the star-planet system using object oriented programming approach and Physics equations. This simulation requires parameters such as radius of the host star and planet, mass of the star and planet, colour of the star and orbital radius. These parameters will be needed for later comparison. The output simulation is saved as a video. The view of the video can be changed for having different inclination of the system and further analysis. Here every second of the video is considered as one day.

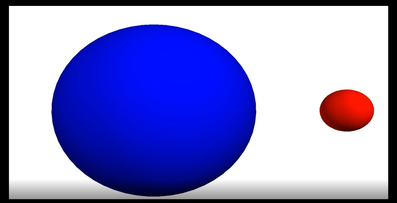


Fig.1 - Simulation of planet around star in visual python

1. **Tracker**

   This video is given as the input file in tracker and analysed using RGB profile option for producing the graph of time vs luma and generating the respective data file columns. These columns should be saved in excel and converted into the comma separated values (.csv) format.

   We find the transit width, transit duration and the orbital period of the planet for calculating

- Radius of Exoplanet

$$\frac{\Delta F}{F} = \left(\frac{Rp}{Rs}\right)^2$$

- Orbital Radius, semi major axis. Here, we only assume the case of circular orbit.

$$a = \sqrt[3]{\frac{P^2 GMs}{4\pi^2}}$$

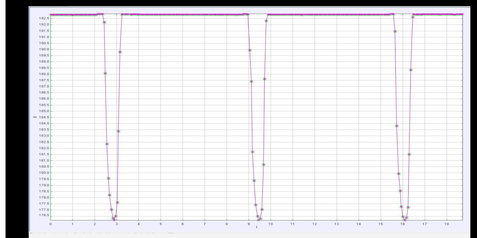


Fig.2 - Lightcurve in Tracker analysis

2. **Python Algorithms**

.csv data files can be read using pandas package and first, columns are separated and saved in variables. Since the vpython system is ideal, we add noise into the data file. We then apply the median filter from the scipy module to reduce the star's wobbling and the noise. We use the BoxLeastSquare method from astropy to plot the periodogram and obtain the orbital period at the maximum power. Further, using the Compute.stats method we derive transit depth, transit duration and calculate the radius, impact parameter, and the orbital radius('a') of the exoplanet. We also fit our data to verify the calculated parameters using astropy.model method.

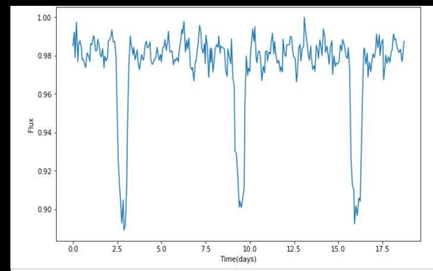


Fig. 3 - Lightcurve in Python

OBSERVATION

CASE I

Softwares	Orbital Period (days)	Transit Depth	Transit Width	Radius of planet(Rp)	Orbital Radius ('a')	Theta (Deviation from 90) deg	Impact parameter
Visual Python simulation	6.50	-	-	6.0e9	7.70e10	-	-
Python	6.58	0.051	0.57	9.0e9	1.03e10	3.88	0.257
Tracker	6.62	0.089	0.90	1.2e10	1.03e10	1.15	0.870

CASE II

Softwares	Orbital Period (days)	Transit Depth	Transit Width	Radius of planet(Rp)	Orbital Radius ('a')	Theta (Deviation from 90) deg	Impact parameter
Visual Python simulation	6.80	-	-	1.0e10	7.70e10	-	-
Python	6.58	0.110	0.42	1.4e10	1.03e10	3.88	0.257
Tracker	6.61	0.210	1.02	1.9e10	1.038e10	1.82	0.550

CASE III

Softwares	Orbital Period (days)	Transit Depth	Transit Width	Radius of planet(Rp)	Orbital Radius ('a')	Theta (Deviation from 90) deg	Impact parameter
Visual Python simulation	4.06	-	-	6.0e9	7.70e10	-	-
Python	3.89	0.073	0.29	1.1e10	7.36e9	5.50	0.256
Tracker	3.92	0.089	0.587	1.2e10	7.36e9	1.64	0.610

- Value deviation in softwares w.r.t. Visual python

Python coding method (Jupyter lab)		
Cases	Deviation of Radius of planet	Deviation of Orbital period
Case I	50.38%	86.62%
Case II	35.00%	86.61%
Case III	81.16%	90.57%
Tracker software and analysis		
Cases	Deviation of Radius of planet	Deviation of Orbital period
Case I	50.00%	86.63%
Case II	88.71%	86.65%
Case III	98.68%	90.57%

RESULTS

We clearly see that both approaches produce similar results and are pretty close to the input values of the Vpython simulation. We also saw huge deviations in orbital radius values from python as well as tracker. Orbital period in Vpython is determined with the help of a stopwatch by measuring the time taken by the planet to come to the same point where the mouse pointer is kept on the stellar surface. This method is a bit vague for research purposes but definitely suitable for making students understand the concept in a better way.

CONCLUSION

We showed this method to few of the undergraduates and high school students. We found the second year undergraduates with basic python knowledge were able to follow the python programming approach on the other hand high school students found it confusing and difficult since they don't have a strong background of object-oriented programming. Both of them followed the Tracker method. We also saw that tracker software has a limitation for analysing star-planet system size. It's a preliminary method and can be further refined for more robust model by strong computation, better tracking softwares and combining more packages to analyse more complicated datasets.

REFERENCES

1. *1. Bradley W. Caroll & Dale A. Ostlie (2014), "Chapter 7" An Introduction to Modern Astrophysics (2nd ed.)*
2. *Exoplanets-the-exoplanet-transit-method by Paul Anthony Wilson https://www.paulanthonywilson.com/exoplanets/*
3. *BLS-Tutorial https://gist.github.com/dfm/96e30d891c0d351daae9aaaef56d3e78*
4. *About Transits by NASA nasa.gov/kepler/overview/abouttransits*

AUTHORS

Sarvesh Bhogaokar

Tejas Sonawane

Jatin Rathod

Sarvesh Bhogaonkar -sarvesh.bhogaokar@xaviers.edu.in
Tejas Sonawane- tejas.sonawane@xaviers.edu.in
Jatin Rathod - rathodjatin@gmail.com

Scan for code