Tutorial 2

Time Series & Data Analysis

NASSP – UCT

Due 9 March 2020 - 11am

You may use any programming language.

Submit both the code (can be as Jupyter notebook) and a pdf on cloudcape.saao.ac.za

Question 1

1.1 Code your own discrete Fourier transform (DFT).

1.2 Create and plot a fake lightcurve containing a sinusoidal variation of 2.5 days. (You can use `create_data` function as below, or your own.)

1.3 Run your DFT on the lightcurve. Plot resulting Fourier transform as a function of frequency.

1.4 Plot $|F(\omega)|^2$ as a function of frequency.

1.4 Indicate on your plot the position of the 2.5 day input period.

1.6 Compare your result to the `numpy` implementation of the Fast Fourier Transform.

Question 2

Use the Kepler lightcurve of an eclipsing binary `09514070.00.lc.data`.

2.1 Generate and plot a Lomb-Scargle periodogram for this lightcurve. Motivate your choice of $f_{\rm min}$, $f_{\rm max}$ and $N_{\rm eff}$.

2.2 Find the peak frequency and fold the lightcurve at this frequency. Is this the orbital period of the binary? Substantiate your answer with text and more plots if needed.

2.3 Write your own bootstrapping code to determine if the measured period is significant at the 99% level.

2.4 How does the above result compare with the Horne & Baliunas false alarm probability?

2.5 A number of peaks are visible in the Lomb-Scargle periodogram. Explain what they are in terms of aliasing and/or harmonics.

2.6 Estimate the error on your period determination – explain your estimate.

Question 3

Repeat Question 2 for the lightcurve `lc100099.dat`.

Question 4

4.1 Using Jake Vanderplas's `create_data` function below, show that the peak width of the Lomb-Scargle periodogram is insensitive to the both the number of data points in the lightcurve, as well as the signal-to-noise in the lightcurve.

```
import numpy as np

def create_data(N, T=4, signal_to_noise=5, period=1.0, random_state=None):
    rng = np.random.RandomState(random_state)
    t = T * rng.rand(N)
    dy = 0.5 / signal_to_noise * np.ones_like(t)
    y = np.sin(2 * np.pi * t / period) + dy * rng.randn(N)
    return t, y, dy
```

4.2 Discuss whether the significance of the peak is sensitive to either the number of data points in the lightcurve or the signal-to-noise.