Precision Asteroseismology
Proceedings IAU Symposium No. 301, 2013
J. A. Guzik, W. J. Chaplin, G. Handler & A. Pigulski, eds.

doi:10.1017/S1743921313014853

Pysca – Automated frequency extraction from photometric time series

Wiebke Herzberg and Kolja Glogowski

Kiepenheuer-Institut für Sonnenphysik, 79104 Freiburg, Germany
email: herzberg@kis.uni-freiburg.de, glogowski@kis.uni-freiburg.de

Abstract. Pysca, a Python software package for automated extraction of frequencies, amplitudes and phases from non-equally sampled photometric time series, is presented.

Keywords. methods: data analysis, stars: oscillations

1. Overview

Pysca is a software package that allows automated extraction of frequencies, amplitudes and phases from non-equally sampled photometric time series. Suited mainly for heat-driven pulsators, it can be regarded as an automated and more focused alternative to the feature rich and interactive Period04 (Lenz & Breger 2005). The use of Pysca is especially convenient when dealing with long time series of high signal-to-noise photometry, (e.g. as provided by the *Kepler* space telescope) where many frequencies are present in the data and the manual extraction can become very time consuming.

Pysca extracts frequencies by identifying the highest peaks in the Lomb-Scargle periodogram and fitting the time series with a superposition of harmonic functions of the corresponding frequencies. It is implemented as an iterative algorithm where the time series is progressively prewhitened up to a user defined termination condition. The signal-to-noise ratio is calculated for every frequency as a statistical measure of significance.

2. Algorithm

A schematic overview of the different steps involved in the frequency extraction process is shown in Fig. 1. The first step is to find the frequency of the highest peak in the Lomb-Scargle periodogram (LSP), which is calculated using the method described in Press & Rybicki (1989). The accuracy of the peak frequency is improved by applying a parabolic interpolation using 3 points. Next, the amplitudes A_i and phases ϕ_i of the new frequency and of all extracted frequencies ν_i from previous iterations are (re-)determined by fitting a model function of superimposed harmonic functions,

$$f(A_i, \phi_i; \nu_i) = \sum_i A_i \sin\left[2\pi(\nu_i t + \phi_i)\right],$$

to the original time series with fixed values of ν_i. The original time series is then prewhitened using $f(A_i, \phi_i; \nu_i)$ and a new LSP is computed where, due to the prewhitening, the peaks and sidelobes of all already handled modes are removed. The noise level at the peak positions is determined using the median of the data in neighborhood of ν_i from the “cleaned” periodogram and the signal-to-noise ratio is calculated using the A_i. After that the next iteration starts by finding the highest peak in the new LSP from which all previous peaks were removed. An example of the prewhitening performance is shown in Fig. 2.

https://doi.org/10.1017/S1743921313014853 Published online by Cambridge University Press

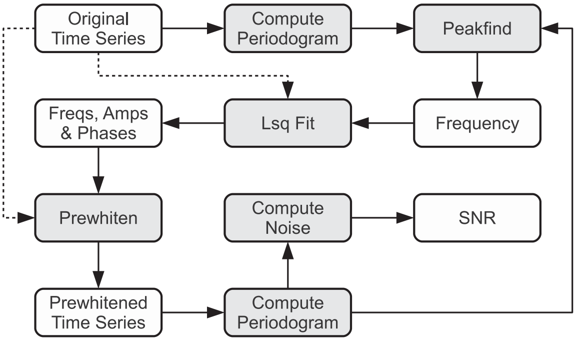

Figure 1. Steps of the frequency extraction procedure.

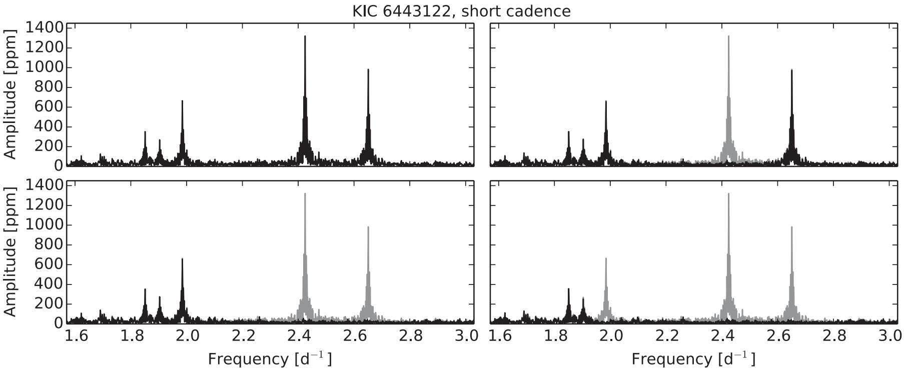

Figure 2. Peak extraction example for data obtained with the *Kepler* space telescope. The upper left panel shows a selected part of the original periodogram. The remaining three panels show the prewhitened periodograms (black) after peak extraction. The original peaks that were removed in this process are shown in gray.

3. Software package

Pysca is available via the Python Package Index (Herzberg & Glogowski 2013). The software package consists of a Python library and a ready-to-use command-line executable. The command-line program can be used to read in photometric time series from a file (supported are FITS files and plain text files) and to extract peak frequencies as well as the corresponding amplitudes and phases. The number of automatically extracted frequencies can be controlled by the various parameters. For developers it is also possible to use the library directly in order to create customized scripts for frequency extraction or incorporate Pysca into other Python programs.

Acknowledgements

WH acknowledges financial support by the Deutsche Forschungsgemeinschaft (DFG) in the framework of project UY 52/1-1.

References

Herzberg, W. & Glogowski, K. 2013, *PyPI*, http://pypi.python.org/pypi/pysca/
Lenz, P. & Breger, M. 2005, *CoAst*, 146, 53
Press, W. H. & Rybicki, G. B. 1989, *ApJ*, 338, 277

https://doi.org/10.1017/S1743921313014853 Published online by Cambridge University Press