Time Series Analysis

Alex Schwarzenberg-Czerny
Copernicus Astronomical Center, Warsaw
&
Astronomical Observatory of Adam Mickiewicz University, Poznań
alexcamk.edu.pl

September 13, 2023

Contents

1 Introduction

Why yet another package of routines for time series analysis (TSA)? This is so as comparatively little effort so far was spend on tools for unevenly sampled time series. Due to day, moon and season cycles such observations are most often encountered in astronomy. Hence, our first rmotivation is to provide modern tools to handle such series. The second reason is to offer astronomers possibility to evaluate signifcance of their results with modern statistical tools. The classical Power Spectrum precludes such evaluation. Our third reason is to provide routines more fully incorporating weighting of observations than those aviaible so far. Our package is intended for observers analysing their data. However, in our opinion statistical procedures not be used as black boxes, lest serious errors may occure. For this reason we provide here an extended background on TSA to help user find its way in the statistical maze.

The present documentation is not meant to be a substitute for a textbook on TSA, however. For brief introduction to TSA we recommend the reviews by Deeming (1975) and Schwarzenberg-Czerny (1998) for a general introduction. On larger scale books by Brandt (1970) and Eadie et al. (1971) provide extremely good introduction to general statistics, the former on somewhat easier level. It would became clear through this document that for astronomers TSA is best seen as another exercise in fitting data with a model. For this reason such general statistical text are quite useful. Relevant statistical tables are listed in Brandt (1970) and Abramovitz & Stegun (1972), and a code for the computation of the probability functions is provided by Press et al. (1986), in the R package (http://www.r-project.org/) and in Mathematica, among others.

We explain here the concepts used in the description of our software and point out their statistical context (Sect. 2). Next we discuss tools to evaluate methods performance and ways to optimize it (Sect. 3). We briefly review backround of statistical procedures, including Fourier analysis in Sect. 4). Sect. 5 contains the general description of the `python` TSA package and its commands. A summary of the commands and their syntaxes is listed in Sect. 6. Examples of some typical problems and their treatment within `python` are given in Sect. 7.

The interests of galactic and extragalactic astronomers in time series analysis differ markedly. The former are particularly interested in the analysis of genuinely *periodic, i.e. deterministic* variations, while the latter are more often concerned with the *stochastic* phenomena. We have implemented tools to satisfy the basic needs of both types of users, with emphasis on the periodic variations. This chapter provides general guidelines for the user of the *python* TSA package. Detailed technical information is provided separately as help for each command of the package.

The ancestors of this `pyaov` procedures are two stand-alone interactive packages `ula` and `zuza` developed in `fortran77` and `90` consecutively at Warsaw University Observatory and Copernicus Astronomical Centre in Warsaw. Then some routines were ported into European Southern Observatory MIDAS environment as TSA contributed context.

2 Basic principles of time series analysis

The practical problems in the analysis of time series concern

1. 1. the detection of a signal against noise,
2. the estimation of the parameters characterizing the signal, and
3. the presentation of the results.

Presentation and/or evaluation of the results involves a function of the time series called statistics. Detection of a signal, e.g. a period, and evaluation of its properties, e.g. error and significance, then usually mean finding and characterizing a related signal in the associated statistics which is normally based on some model of the signal.

2.1 Signals and their models

Signals can be classified broadly into *deterministic* and *stochastic signals*. A deterministic signal, e.g. a *periodic signal*, can be predicted for arbitrary spaces of time. For a stochastic signal, no such prediction can be made beyond a certain time interval, called a *correlation length* l_{corr}. For any finite time series the classification into these two categories is ambiguous so one usually relies on some prejudice on the signal nature. Sometimes methods suitable for both stochastic and periodic signals could be applied to any time series with some success (e.g. quasiperiodic oscillations, Sect. 4.4).

Usually processes in the source of the signal (e.g. the nucleus of an active galaxy) and/or observational errors introduce a random component into the series, called noise. The analysis of such series usually aims at removing the noise and fitting a *model* to the remaining component of the series. Suitable models can be obtained by shifting a known series by some *time lag*, l, or by repeating fragments of it with some *frequency*, ν. Accordingly, we are speaking of an analysis in the *time* and *frequency domain*. In these domains the correlation length l_{corr} and oscillation frequency ν_0, respectively, have particularly simple meanings. It is transparent that the stochastic signals are analysed more comfortably in the time domain and periodic signals in the frequency domain.

One way to fit data with a model is to minimise their least squares (LSQ). Our expposition of LSQ relies on R.A. Fisher theory. Let n observations $\mathbf{x} = (x_1, \cdots, x_n)$ are fitted with the *orthogonal model* $\mathbf{x}_\| = \sum_i c_i \mathbf{p}_i(\varphi)$, such that vectors/functions $\mathbf{p}_i(\varphi) \equiv \mathbf{p}_i(2\pi\mathbf{t})$ are mutually orthogonal with respect to the scalar product defined by observed phases:

$$0 = (\mathbf{p}_i, \mathbf{p}_j) \equiv \sum_\varphi w_\phi p_i(\varphi) p_j(\varphi) \quad \text{for } i \neq j \tag{1}$$

Hereafter we assume that the average values $(\mathbf{1}, \mathbf{x}) = (\mathbf{1}, \mathbf{x}_\|) = 0$. By virtue of Fisher lemma the model $\mathbf{x}_\|$ and residuals from fit $\mathbf{x}_\perp \equiv \mathbf{x} - \mathbf{x}_\|$ are orthogonal $(\mathbf{x}_\|, \mathbf{x}_\perp) = 0$. In consequence an n-dimensional analogue of Pythagoras theorem holds:

$$\begin{array}{lll} \|\mathbf{x}\|^2 = & \|\mathbf{x}_\|\|^2 + & \|\mathbf{x}_\perp\|^2 \quad \text{where} \\ n = & n_\| + & n_\perp \\ \text{observation} = & \text{model} + & \text{residuals} \end{array} \tag{2}$$

$\|\mathbf{x}\|^2 \equiv (\mathbf{x}, \mathbf{x})$ and n, $n_\|$ and $n_\perp$ denote number of observations, number of model parameters and number of degrees of freedom of residuals. Because of the relation $\|\mathbf{x} - \mathbf{x}_\|\|^2 = \|\mathbf{x}\|^2 - 2(\mathbf{x}, \mathbf{x}_\|) + \|\mathbf{x}_\|\|^2$ where only the middle term depends on the frequency ν, our considerations for LSQ also apply for the case of cross-correlation function (CCF) periodogram.

Suitable families of orthogonal functions $\mathbf{p}_i$ are either Szegö trigonometric polynomials or the tophat functions corresponding to the phase bins (Schwarzenberg-Czerny, 1996, 1989). Nominally, Szegö polynomials follow from Gramm-Schmidt ortonormalization of Fourier harmonics, yet convenient recurrence formulae also exist. Phase folding and binning of data is equivalent to LSQ fitting of a step function built of a linear combination of tophats. The box function employed for planetary transit search corresponds to two phase bins of unequal width (Schwarzenberg-Czerny & Beaulieu, 2006), so the present considerations aplly in this area too. Quite unique orthogonal functions were employed by MACHO (Akerlof et al., 1994).

2.2 Signal detection

A *statistics* $\Theta(\nu, \mathbf{x})$ is the merit figure indicating quality of the fit. A *periodogram* is the plot of $\Theta(\nu, \mathbf{x})$ against ν. Patterns in the periodogram may relate to the presence in data of the oscillations with the corresponding frequency. Classically signal detection proceeds by assumption that data consist of pure noise (so called null hypothesis H_0) and then demonstrating that in such case the observed value of Θ is unlikely. This procedure is not unlike mathematical proofs by *reductio ad absurdum*. Hence significance of detection depends on the probability distribution

Table 1: Basic Classes of Period Statistics

Statistics	Definition	Distribution	Name	Analogues
$\Theta_{AOV} =$	$\frac{\lVert\mathbf{x}_\parallel\rVert}{\lVert\mathbf{x}_\perp\rVert}$	$F(n_\parallel, n\perp; \Theta_{AOV})$	Fisher-Snedecor	`aovw`$^{(1)}$,`amhw`$^{(2)}$
$\Theta_\parallel =$	$\frac{\lVert\mathbf{x}_\parallel\rVert}{\lVert\mathbf{x}\rVert}$	$\beta(n_\parallel, n\perp; \Theta_\parallel)$	β distribution	Power$^{(3)}$, L-S$^{(4,5)}$
$\Theta_\perp =$	$\frac{\lVert\mathbf{x}_\perp\rVert}{\lVert\mathbf{x}\rVert}$	$\beta(n_\perp, n\parallel; \Theta_\perp)$	β distribution	χ^2,PDM$^{*\ (6,7)}$

References: (1)- Schwarzenberg-Czerny (1989), (2)- Schwarzenberg-Czerny (1996), (3)- Deeming (1975), (4)- Lomb (1976), (5)- Scargle (1982), (6)- Stellingwerf (1978) with corrected distribution in (7)- Schwarzenberg-Czerny (1997)

of Θ for the hypothetical data consisting of a pure noise. This case is called in Statistics null hypothesis H_0.

2.3 Test statistics

The test statistic used for detection is a special case of a function of random variables. Θ must be dimension-less as no statistical conclusions may depend on units. There are three ways to construct dimension-less Θ statistics from the dimensioned $\lVert\mathbf{x}\rVert$, $\lVert\mathbf{x}_\parallel\rVert$ and $\lVert\mathbf{x}_\perp\rVert$ (Table 1). The probability distributions listed in the table are discussed by e.g. Eadie et.al. (1971), Brandt (1970), and Abramovitz & Stegun (1972). The two latter references contain tables. For a computer code for the computation of the cumulative probabilities see Press et. al. (1986).

Because of Eq. (2) all these Θ's are uniquely related:

$$\Theta_\perp = 1 - \Theta_\parallel = \frac{1}{1 + \Theta_{AOV}} \tag{3}$$

hence the corresponding F and β distributions may be obtained from each other by the suitable change of variable. From this we find, that conclusions drawn from the Θ_{AOV}, $\Theta_\parallel$ and $\Theta_\perp$ periodograms must all be identical *if and only if* the model $\mathbf{x}_\parallel$ remains the same. In other words what merits is not a shape of the periodogram peak but its probability (Schwarzenberg-Czerny, 1998). Turning this argument *ad absurdum*, one may argue that to obtain a clean single peak periodogram suffices to raise any periodogram to the power of 1000 or so. As no additional information was supplied, such a nice view has spurious meaning. In practical terms it suffices to discuss periodograms in which oscillations correspond to peaks. Any results would also apply to the periodograms showing throughs at the corresponding frequencies.

For the human eye the equivalent periodograms may look deceptively different, however. For high S/N and χ^2 periodogram, an alias minimum of $\Theta_\perp$ twice as high as the true minimum would not look significant. At the same time the corresponding alias peak power $\Theta_{\|} = 1 - \Theta_{\perp}$ would almost match the true peak, pretending to be significant. From this point of view for periodograms in the frequency domain we recommend Fisher Θ_{AOV} statistics, or still better $0.5 \log \Theta_{AOV}$ which has near normal probability distribution.

Depending on different signal models, the Θ_{AOV} is implemented in the `pyaov` routines `amhw`, `aovw` and `atrw` (Sect. 5.5). `amhw` uses trigonometric series and `aovw` and `atrw` usee step function (phase binning) as models. Except for $\Theta_{\|} \rightarrow \Theta_{AOV}$ conversion Lomb-Scargle periodogram constitutes a mere special case of `amhw` for `nh2=2`. In the time domain, in `pyaov.covar` routine we rely on $\|x_{\perp}\|^2$ or on $\Theta_{\perp}$ statistics, often imprecisely called χ^2. (Sect. 5.6). `pyaov.covar` serves to compare two series of observations (possibly identical). It employs a step function (binning).

3 Sensitivity of Detection

3.1 Test Power

To evaluate sensitivity of detection we must consider two different hypothetical data sets: for a pure noise noise of standard deviation 1 and for the noise plus a periodic signal of amplitude A (same units). In Statistics these two cases are called the null and alternative hypotheses, H_0 and H_1, respectively. Accordingly, for H_0 and H_1, Θ obey different probability distributions $P_0(\Theta)$ and $P_1(\Theta)$. Ideally the two distributions are separated by a critical value Θ_c. Say, $\Theta < \Theta_c$ corresponds to a pure noise and $\Theta > \Theta_c$ to the detected of signal. However, in realistic situations the two distributions overlap over a range of Θ. Thus two kinds of errors arise: one claims detection while in reality H_0 is true (*false positives*) and reversly, one claims no detection while in reality H_1 remains true (*misses*). In the classical statistics one fixes Θ_c so that false positives occure seldom, i.e. the significance level $\alpha = P_0(\Theta < \Theta_c)$ is close to 1. Then *test power* of the criterion Θ_c is defined as $\beta = P_1(\Theta < \Theta_c)$, where probability of misses is $1 - \beta$. Thus for a fixed Θ_c, large β corresponds to good detection sensitivity. Tha analytical formulae for P_0 are listed in Table 1. No corresponding formulae are known for P_1 as they depend in a complex way on signal shape. However, for small signal-to-noise $A/1 \ll 1$ it is possible to derive approximate asymptotic formulae for P_1 (Schwarzenberg-Czerny, 1999). In this approximation P_0 and P_1 retain the same shape yet are shifted, in units of their standard deviation, by

$$\Delta\Theta/\sqrt{Var\{\Theta\}} \quad = \quad A^2 n \frac{\|s_{\|}\|^2}{\sqrt{2n_{\|}}} \quad \text{where} \tag{4}$$

$$\|s_{\|}\|^2 = \frac{(x_{\|signal}, x_{\|model})^2}{\|x_{\|signal}\|^2 \|x_{\|model}\|^2} \quad (5)$$

and $x_{\|signal}$, $x_{\|model}$ denote shapes of the real signal and of the fitted model, respectively. The bigger $\Delta\Gamma$ the more sensitive is our method / model for a given signal.

Let us first consider a sinusoidal oscillation with an amplitude not exceeding the noise. Then, all statistics based on models with 3 parameters have similar values of the test power: power spectrum, Scargle statistics, $\chi^2(3)$ fit of a sinusoid, `amhw(...,nh2=3)`, `aov(...,nh2=3)` and corrected PDM(3) statistics. (Please note that we depart here from the conventional notation by indicating in brackets the number nm of the degrees of freedom of the model e.g. the number of series terms or phase bins instead of the number of the residual degrees of freedom $n_{\perp}$.) Statistics with more than 3 parameters, e.g. `amhw(...,nh2=n)`, `aov(...,nh2=n)`, PDM(n) and $\chi^2(n)$ with $n \gg 3$ and an extended Fourier series have less power. Our final choice is guided by the availability of the analytical probability distribution of the test statistics. Summing up, we recommend tu use for the detection of sinusoidal and other smooth oscillations of small amplitude the statistics with a coarse phase resolution, e.g. `amhw(...,nh2=3)`, Scargle and `aov(...,nh2=3)`.

For a narrow gaussian pulse or eclipse of width w repeating with period P (i.e. duty cycle w/P), the most powerful statistics are these with the matching resolution: `amhw(...,nh2≈` $P/2w$`)`, `aov(...,nh2≈` $P/2w$`)` and $\chi^2(P/2w)$. Power spectrum, Scargle, `amhw(...,nh2=3)`, `aov(...,nh2=3` and $\chi^2(3)$ all have less power. Note the equivalence of the $\chi^2(3)$ and Scargle's statistic (Lomb, 1976, Scargle, 1982) and the near-equivalence of the power spectrum and Scargle's statistics in the case of nearly uniformly sampled observations. Considering both test power and computational convenience we recommend for signals with sharp features, e.g. narrow pulses or eclipses, to use the ORT and AOV with the resolution matched to the width of these features.

Among many other statistics we mention the one by Lafler & Kinman (1965), phase dispersion minimization (PDM) also known as the Whittaker & Robinson statistic (Stellingwerf, 1978), string length (Dvoretsky, 1983), and statistic introduced by Renson (1983).

3.2 Corrected Significance: Bandwidth Penalty

From Table 1 one may derive the analytic tail probability of large Θ for a single frequency: $Q_1(\Theta > \Theta_0) \equiv 1 - P_1(\Theta < \Theta_0)$. As more and more frequencies are examined in the periodogram, probability of spurious occurence of a peak due to pure noise increases, in the same way as the probability of winning a lotery increases with the number of trials. This increase of the probability, called *bandwidth penalty*, has to be accounted for any realistic statistical evaluation. Because of aliasing values of a periodogram at different frequencies may be strongly correlated, so that out of N investigated frequencies only $N_{eff} \leq N$ may be independent. If so, the postulated tail probability could be (e.g. Horne & Baliunas 1986):

$$Q_N(\Theta_0) = Q_1(\Theta_0)^{N_{eff}}. \tag{6}$$

The hitch is in the unknown value of N_{eff}. Paltani (2004) proposed a useful method to estimate N_{eff} by MC simulations, yet relying on their mean rather than extreme values.

Estimation of bandwidth penalty is less of a problem in large photometric surveys. There many light curves are obtained from the same stack of images, hence they suffer from this effect in the same way, so reliable detection criteria may be obtained from analysis of Θ. However, such criteria may not be transfered reliably from one field into another as they suffer from different sampling effects. Then full conversion to probabilities, including bandwidth penalty correction, remains the only reliable solution, in our opinion.

3.3 Corrected Significance: Correlation or Red Noise Effect

Presence of a *correlation (red noise)* in observations may ruin simplistic statistical estimates. For example LSQ fit of a sine to the solar spot Wolfer numbers spanning 100 years yields the nominal period $P \approx 11$ y with an error of order $0.002P$. However, propagation of such an ephemeris for the next 50 years demonstrates that a realistic period error was $\approx 0.1P$. This happens as consecutive residuals from the fit are correlated (keep the same sign over decades) while the standard LSQ error estimates implicitly *assume* that residuals are *(uncorrelated) white noise*. Conversly, for the simulated data consisting of white noise plus the oscilation of the same variances/amplitudes as above, the $0.002P$ error estimate proves realistic.

This remarkable effect of the correlation is seldom discussed in texts on LSQ. In fact, correlation of every N_{corr} consecutive observations decreases the effective number of observations roughly by a factor of N_{corr} and hence increases the real LSQ errors by a factor of $\sqrt{N_{corr}}$(Schwarzenberg-Czerny, 1991). A simple way to estimate N_{corr} is by counting the number of sign changes in the residualds from fit (the *post mortem* analysis). For white noise one expects $N_{obs}/2$ changes of sign (every second residual should change sign, on average). If the observed number of sign changes of residuals is $N_{sign} < N_{obs}/2$ then the number of consecutive correlated observations is $N_{corr} \approx N_{obs}/(2N_{sign})$.

4 Methods

4.1 Fourier transforms

Transformations which take functions, e.g. x, y as arguments and return functions as results are called operators. The direct and inverse Fourier transform, $\mathcal{F}^{\pm 1}$, and the convolution, Λ, are operators defined in the following way:

$$\begin{aligned} \mathcal{F}^{\pm 1}[x](\nu) &= C_{\pm} \int_{-infty}^{+\infty} e^{\pm 2\pi i t \nu} x(t) dt = \frac{1}{n_o^{\frac{1 \pm 1}{2}}} \sum_k = 1^{n_o} x_k e^{\pm 2\pi i t_k \nu} && (7) \\ \left[x * y\right](l) &= C_{\pm} \int_{-infty}^{+\infty} x(t) y(l-t) dt = \frac{1}{n_o} \sum_k = 1^{n_o} x_k y_{l-k} && (8) \end{aligned}$$

where square brackets, [], indicate the order of the operators and round brackets, (), indicate the arguments of the input and output functions. Without loss of generality we consider here functions with zero mean value. Note that because of the finite and infinite correlation length of stochastic and periodic series, respectively, no unique normalization C applies in the continuous case.

The discrete operators $\mathcal{F}^{\pm 1}$ and $*$ are well defined only for observations and frequencies which are spaced evenly by δt and $\delta\nu = 1/\Delta t$, respectively, and span ranges Δt and $\Delta\nu = 1/\delta t$. Then and only then $\mathcal{F}^{\pm 1}$ reduces to orthogonal matrices. It follows directly from Eq. (7) that we implicitly assume that the observations and their transforms are periodic with the periods Δt and $\Delta\nu$, respectively. The assumption is of consequence only for data strings which are short compared to the investigated periods or coherence lengths or for a sampling which is coarse compared to these two quantities. Such situations should be avoided also in the general case of unevenly sampled observations.

The following properties of $\mathcal{F}^{\pm 1}$ and $*$ are noteworthy:

$$\begin{aligned} \mathcal{F}[x+y] &= \mathcal{F}[x] + \mathcal{F}[y] && (9) \\ \mathcal{F}[x * y] &= \mathcal{F}[x]\mathcal{F}[y] && (10) \\ \mathcal{F} e^{\pm 2\pi i t \nu_0} &= \delta_{\nu_o}(\nu) && (11) \end{aligned}$$

where δ_x denotes the Dirac symbol: $\int \delta_x f(y) dy = f(x)$. In the discrete case, δx assumes the value n_o for x and 0 elsewhere.

4.2 The power spectrum and covariance statistics

Let us define power spectrum, covariance and autocovariance statistics P, Cov and ACF :

$$P[x](\nu) = |\mathcal{F}x|^2 \quad (12)$$

$$Cov[x,y](l) = x(t) * y(-t) \quad (13)$$
$$ACF[x](l) = Cov[x,x](l) \quad (14)$$

The power spectrum is special among the periodograms in that it is the square of a linear operator and reveals the important correspondence between frequency and time domain analyses:

$$P[x](\nu) = |\mathcal{F}[ACF[x](l)x](\nu) \quad (15)$$

by virtue of Eq. (12.5).

Let us consider which linear operators or matrices convert series of independent random variables into series of independent variables. For the discrete, evenly sampled observations the ACF is computed as the scalar product of vectors obtained by circularly permutating the data of the series. For a series of independent random variables, e.g. white noise, the vectors are orthogonal. It is known from linear algebra that only orthogonal matrices preserve orthogonality. So, only in the special case of evenly spaced discrete observations and frequencies (Sect. 4.1) are $\mathcal{F}[x]$ (and $P[x]$) independent for each frequency. In the next subsection we discuss the case of dependent and correlated values of $P[x]$.

4.3 Sampling patterns

The effect of a certain sampling pattern in the frequency analysis is particularly transparent for the power spectrum. Let s be the sampling function taking on the value 1 at the (unevenly spaced) times of the observations observation and 0 elsewhere. The power spectrum of the sampling function

$$W(\nu) = |\mathcal{F}s|^2 \quad (16)$$

is an ordinary, nonrandom function called the spectral window function. The discrete observations are the product of s and the model function $f : x = sf$ so that their transform is a convolution of transforms: $\mathcal{F}x = [\mathcal{F}s] * [\mathcal{F}f] \equiv S * F$, where $S = \mathcal{F}s$ and $F = \mathcal{F}f$. For $f = A\cos 2\pi\lambda t \equiv A\left(e^{+2\pi\lambda t} + e^{-2\pi\lambda t}\right)/2$ and $F = \mathcal{F}f = A(\delta_{+\nu} + \delta_{-\nu})/2$ we obtain the result $\mathcal{F}x = A(S(\nu-\lambda)+S(\nu+\lambda))/2$. Because of the linearity of $\mathcal{F}$ our result extends to any combination of frequencies. Taking the square modulus of the result equation, we obtain both squared and mixed terms. The mixed terms $S(\nu+\lambda_k)S(\nu+\lambda_j)$ correspond to an interference of frequencies $\nu+\lambda_k$ and $\nu+\lambda_j$, differing by either sign or absolute value. Therefore, if interference between frequencies is small, the power spectrum reduces to the sum of the window functions shifted in frequency:

$$P(\nu) \approx \sum |[\mathcal{F}s](\nu + \lambda_k)|^2 \equiv \sum W(\nu + \lambda_k) \quad (17)$$

In the opposite case of strong interference, ghost patterns may arise in the power spectrum due to interference of window function patterns belonging to positive as well as negative frequencies. The ghost patterns produced at frequencies nearby or far from the true frequency are called aliases and power leaks, respectively.

4.4 Time domain analysis

As noted before stochastic signals are best analysed in the time domain. The analysis in the time domain often involves the comparison of two different signals while in the frequency domain analyses usually concern only one signal. The expectation value of the covariance function of uncorrelated signals is zero. The expected value of the autocorrelation function ($E\{ACF\}$, Sect. 4.2) of white noise also is zero everywhere except for 1 at 0 lag. The expected ACF of a stochastic signal of correlation length l vanishes outside a range $\pm l$ about the lags. The ACF of a deterministic function does not vanish at infinity. In particular the ACF of a function with period P has the same value, P. Signals of intermediate or mixed type with an ACF which has several maxima spaced evenly by l and a correlation length $L \gg l$ is called a *quasiperiodic oscillation*. Its power is significantly above the noise in the $1/\pm 1/L$ range of frequencies and its correlation length L is called the *coherence length*.

4.5 Parameter estimation

In this context, ν (or, in the time domain, l) are no longer independent variables. They are treated like any of the other parameters: i.e. are assumed to be random variables to be estimated from the observations by fitting a model. Parameter estimation in the frequency domain is best done by fitting models using χ^2 statistics (least squares). The present package contains just one such model, namely Fourier series (`pyaov.fouw`). However, note that with its non-linear least squares fitting package, `pylab` offers very versatile, dedicated tools for model fitting.

In the time domain, the most important parameters to be estimated from the data are the correlation length of and time lag between the input signals. This measurement can be done with the command `pyaov.peak`. The correlation length can be obtained as the width of the line centered at zero lag. The time lag can be measured as the center of the corresponding line in the ACF.

4.6 Presentation and inspection of results

A simple way to graphically present the results of a TSA is to plot the test statistics S against its parameter ν or l, depending on whether the analysis was performed in the frequency or time domain. Plots in the frequency domain are called periodograms. In them, oscillations are revealed by the presence of spectral lines. However, some (often many) lines are spurious and simply arise from random fluctuations of the signal. By means of the confidence level α and the probability distribution of S one can find the critical value S crit for significant features. Examples of statistics used in the time domain are covariance and correlation functions. The correlation of a signal with itself or with another signal produces maxima in these functions at particular lags. Detection of genuine lags then consists of testing the significance of such maxima.

5 Python utilities for time series analysis

In this section we describe the functioning of the Python time series analysis (PYAOV) package. For a detailed description of syntax and usage we refer the reader to the respective help information. We precede the command description with the description of the scope of the applications and the structure of the TSA input and output files and of the keywords holding the relevant parameters.

5.1 Scope of applications

Our package is well suited to the analysis of small to modest sized data sets, with no regard to the sampling which may be even or uneven. Thus our package suits astronomers who often have to deal with unevenly sampled observations well. One of the advantages of the package is the availability of tools for a statistical evaluation of the results.

Data sets containing many observations but covering only few cycles and/or characteristic time intervals can be reduced in number by averaging or decimation, usually with little loss of information. However, the analysis of very extensive datasets, which cover many cycles, contain, say, over 10 5 observations and/or are sampled evenly, is more demanding in terms of computing efficiency than in the choice of the method. With the present package, `python` offers an excellent general purpose environment and a variety of tools for the analysis of astronomical data. Any overheads are minimal as computing-intensive routines are implemented in `f95` and pyton wrappers are used only to pass parameters. Very large data sets usually concern important problems and therefore deserve extra attention in the analysis. For such cases any extra overhead is undesirable, whereas extra efficiency can be gained from specialized algorithms implemented as purpose-built standalone codes. One important class of such specialized algorithms not covered here is based upon the fast Fourier transform technique (see e.g. Bloomfield, 1976, Press and Rybicki, 1991).

5.2 The PYAOV environment

Before any of the commands of the `pyaov` package can be used in `python`, the package must be enabled by issuing `import pyaov` command. Some parameters are optional and can be skipped by call. If on a command line a parameter is omitted, it assumes the default value indicated after = sign in command description.

5.3 Input & Output Data

To read an ascii file with numerical data in `python`, use load command.

```
from pylab import load
x = load('ascii.dat', comments='!', skiprows=2)
```

Above one also specifies a characted that is used to recognize disregarded comment lines (default character is '#') and one skips the first two lines. The rest of the file should have numerical data, same number of columns on each line. The result is returned as an array of numbers. For example, `x[:,0]` will be the first column and `x[-1,:]` the last row read from the file. A numeric array can again be written as an ascii file by using save

```
from pylab import save
save('ascii.out', x)
```

With additional arguments one can specify, for example, the formatting used in the writing of the numerical values.Python supports flexible, on the run data conversion. All data are considered float or double precision in case of times. Some command parameters are integers or strings.

5.4 Fourier analysis

Orthogonality of Fourier harmonics in the sense of Eq. (1) holds only for even spaced observations and suitable FFT frequency grid. In consequence of non-orthogonality for uneven sampled observations, results of Fourier analysis are non-optimal as they suffer more from noise interference as do the orthogonal methods discussed later. Only for pure sine model this drawback could be removed by using Lomb-Scargle (LS) periodogram (Lomb 1976Scargle 1982). The LS periodogram is provided here as a special case of more general multi-harmonic AOV periodogram. However, we provide below the Power Spectrum and Fourier series fit routines as they serve two important applications. Namely, time sampling pattern is best evaluated by means of window function calculated with `pyaov.pspw`. The next versatile routine, `pyaov.fouw`, needs preliminary value of frequency and serves for such purposes as: improvement of period accuracy, calculation of Fourier coefficients and pre-whitening data with the given period or trend removal. However, for basic tasks of periodogram analysis one should employ methods of the next section.

`pspw` – **Discrete Power Spectrum (DPS):**

```
th,fr,frmax=pyaov.pspw(time, valin, error, fstop,
fstep, fr0=0.)
```

This command computes the DPS for unevenly sampled data. This implementation has rare property of beeing able to account for weights/errors of measurements. Generally power spectrum *is not optimal* for period analysis, because lack of its model orthogonality in the sense of Eq. (1). One important application of this routine is to calculate window function for evaluation of time sampling pattern. For this particular application, set all data values in column `valin` to 1 and then in `pyaov.amhw` apply `pyaov.pspw`. The resulting power spectrum is the window function of the data set. The advantage of PS stems directly from its excess sensitivity to poor sampling patterns. Excellent discussion of PS and its relevance for different types of signals is by Deeming (1975).

`fouw` – **Fourier series fit:** `Fouw` returns data after subtraction of the best fit Fourier series of given length. The base frequency is an input parameter. For negative value, its sign is changed and the corresponding Fourier series is calculated. For positive value prior to that the frequency value is adjusted by non-liner LSQ iterations. The initial frequency value may be obtained by call of a periodogram routine and/or `peak`. The returned values are Fourier coefficients and their errors. The corresponding value ofχ^2 may in principle be recovered from the standard deviation $\sigma_0 = \sqrt{\chi^2(df)/df}$, where $df = n_o - n_{\|}$ and n_0 and $n_{\perp}$ are the number of observations and the number of Fourier coefficients (including the mean value), respectively.

This versatile command may be used not just for parameter estimation but also for data massaging, to remove a given frequency from the data by prewhitening or just for trend removal (high pass filtering). For the latter purpose choose a negative frequency of low absolute value, so that only few cycles cover the time interval spanned by the observations. After routine completion the fitted trend (low pass filtering) is obtained by subtraction of the returned values from the original ones.

LSQ are known to grossly underestimate frequency errors for correlated observations and/or red noise. To avoid that, analyse the returned vector of residuals with `pyaov.totals` routine to find number of sign changes and hence the average number of consecutive correlated observations n_{corr}

as explained in (Sect. 3.3). To obtain realistic estimates multiply the errors by $\sqrt{n_{corr}}$ (Schwarzenberg-Czerny, 1991).

Residuals may be again analysed by any method supported by the TSA package. In this way, the command can be used to perform pre-whitening and a CLEAN-like analysis by manually removing individual oscillations one by one in the time domain (see Gray & Desikachary 1973, Roberts et al. 1987).

5.5 Time series analysis in the frequency domain

The AOV periodograms already have been in use over 20 years as they found application in half thousend papers. Basing on accumulated experience and theoretical analysis of test power we recommend for smooth signals, e.g. sinusoids, use of either `pyaov.amhw` with `nh2=2` or `4`, or `pyaov.aovw` with `nh2=3` or `5` bins. The former routine is 10 times slower yet has slightly better statistical properties. The latter was already used for analysis over 10^5 stellar light curves and proved fast and reliable. The sensitivity of these statistics to sharp signals (such as strongly pulsed variations or light curves of very wide eclipsing binaries) is poor. For the detection of such signals better use `pyaov.amhw` or `pyaov.aovw` with the width of these features matched by the width of the of the top harmonics or the width of a phase bin, respectively. The command `pyaov.fouw` serves two purposes: a) least squares estimation of the parameters of a detected signal and b) filtering the data for a given frequency (so-called prewhitening). χ^2 statistic used in `pyaov.fouw` is related to that used in Lomb-Scargle (Lomb, 1976, Scargle, 1982). The trend removal (zero frequency) constitutes a special case of this filtering. For the latter purpose `pyaov.prew` is more robust and faster. For a pure sinusoid model we offer generalized Lomb–Scargle (GLS) routines implemented either as special case of the `pyaov.amhw` routine or in its original implementation by Ferraz-Mello (1981). The `pyaov.amhw` implementation of LS is improved as it involves further correction for orthogonal fit of data with an additive constant as argued by Ferraz-Mello (1981). In that sense it is a *generalized least squares* algorithm, yielding results identical to that by Ferraz-Mello (1981) and rediscovered by Zechmeister and Kuester. In some applications provision for a constant has a small effect, yet Foster (1995) (his Fig. 1.) provides an example of data when despite mean subtraction, fit of a constant has profound effect. For completness we provide also original Lomb (1976) implementation, slightly more efficient than its Scargle version. Our implementation includes provision for unequal weights.

`pyaov.amhw` **– Multiharmonic analysis of variance periodogram:** The command computes the analysis of variance (AOV) periodogram for fitting data with a (multiharmonic) Fourier series. The fit of the Fourier series is done by a new efficient algorithm, employing projection onto orthogonal trigonometric polynomials. The results of the fit are evaluated using the AOV statistics, a powerful method newly adapted for the time series analysis (Schwarzenberg-Czerny, 1996, 1989). The model used in this method is the Fourier series of n harmonics. The resolution of the method may be tuned by change of n. Hence it is the method of choice for both smooth and sharp signals. The AOV statistic is the ratio $S(\nu) = Var_{\|} = Var_{\perp}$. The distribution of S for white noise (H_0 hypothesis) and n_j order bins is the Fisher-Snedecor distribution $F(2n+1; n_o-2, n-1)$. The expected value of the AOV statistics for pure noise is 1 for uncorrelated observations and n_{corr} for observations correlated in groups of size n_{corr}.

`amhw`– **Generalized L–S Periodogram** is obtained by the call of `pyaov.amhw` with `nh2=3`, identical to original Ferraz-Mello (1981) method and its later reimplementations. The returned periodogram is of Θ_{AOV} type. In principle it may be converted to its traditional $\Theta_{\|}$ by use of Eq. (3). This should be seldom needed, as we demonstrated in Sect. 2.3 that any statistical conclusions drawn from them must be entirely equivalent. We recommend this version of `pyaov.amhw` or equivalent `pyaov.f_mw` periodogram for the detection of smooth, nearly sinusoidal signals, since then its test power is large and the statistical properties are known. In particular the expected value of Θ_{AOV} is near 1.

`aovw` – **Analysis of variance for phase bins:** This routine computes the analysis of variance (AOV) periodogram for phase folded and binned data. Depending on choice of `nh2=3` or `nh2` large the AOV method is suitable respectively either for smooth or sharp, nonsinusoidal signals (Schwarzenberg-Czerny, 1989). Its slight statistical inefficiency in comparison to `pyaov.amhw` stems from use of the step function model, i.e. phase binning. Such a model does not fit exacly observed light curves. However, in most applications the speed of the AOV routine more than offsets any drawbacks. Its operation count $\mathcal{O}(nn_\nu)$, where n and n_ν denote number of observations and frequencies, remains at least factor 10 less than that of `pyaov.amhw`. In fact in application to typical astronomical data spanning seasonal gaps the binned AOV algorithm is *the fastest available* as it beats even FFT due to avoidance of calculations in the gap.

For observations correlated in groups of size n_{corr}, divide the value of LS statistics by n_{corr} (Sect. 3.3). As indicated in Table 1 the AOV periodograms of both `pyaov.amhw`and `pyaov.aovw` have distinct advantage in obeying F analytical probability distribution with well known statistical properties, for small samples too. On large samples, AOV is not less sensitive than other statistics using phase binning, i.e. the step function model: χ^2 , Whittaker & Roberts and PDM. Therefore we recommend the `pyaov.amhw`and `pyaov.aovw` commands with matching larger `nh2` for samples of all sizes and particularly for signals with narrow sharp features (pulses, eclipses). For numerous observations and sharp light curves use phase bins of width comparable to that of the narrow features (e.g. pulses, eclipses). For peaks/eclipses lasting fraction ϵ of the period, use `nh2` of order $1/\epsilon$. For smooth light curves use low order, $2 \leq$ `nh2` ≤ 6, for optimal sensitivity.

Note that for any periodogram phase coverage and consequently quality of the statistics near 0 frequency are notoriously poor for most observations.

`atrw` **– Analysis of variance for planetary transits/eclipses:** This procedure consits a powerfull variant of `aovw` employing only two unequal size phase bins. The smaller one is selected to coincide with en eclipse/planetary transit in the light curve (Schwarzenberg-Czerny & Beaulieu, 2006). In effect this routine implements the box-like eclipse model in the very economic way. To our knowledge, current method remains the only one with known analytical properties of its probability distribution and sensitivity, while it is not surpassed in terms of sensitivity and speed by any competition. This analitical results enabled evaluation of the cost involved in commonly used box-like model profile as compared to the realistic profile. The effect is equivalent to loss of mere 5% in-transit observations.

This detection method is particularly powerfull in application to light curves pre-filtered of any variations on the scale longer than the transit duration. Then variance of the data in the large bin becomes minimal, aiding detection od departures in the small bin. For best results choose such `nh2` that `1/nh2` matches the transit duty cycle (width-to-period ratio). Remember, that the probability derived from Θ_{AOV} returned by this routine has to be multiplied by `nh2` to account for multiplicity of transit phases.

`f_mw` **– Generalized L–S Periodogram** In this particular application the generalized L-S routine may be sped up by performing calculations on real numbers. For that purpose we have implementsd the original Ferraz-Mello (1981) algorithm which preceeds that by Zehmeister and Kuerster, yet results in the same values at the same cost.

`lomw` **– Lomb Algorithm** Yields the usual L–S periodogram using the original Lomb algorithm (Eq. (1) in Lomb (1976)), slightly faster than that of Scargle. For completness reasons we do provide the veighted version of the original Lomb algorithm. For reasons discussed by Foster (1995) L–S generalized versions, `pyaov.f_mw` or `pyaov.amhw` should be preferred.

`prew` **– trig polynomials fit:** `Prew` returns data after subtraction of the best fit trigonometric polynomial of a given length. Its advantage over `pyaov.fouw` is use of a more robust algorithm, based on diagonal covariance matrix of the base polynomials. Its deficiency is lack of Fourier coefficients output. Frequencyis not not be adjusted, unlike to `pyaov.fouw`. To adjust frequency use `refine`, based on a similar algorithm. To recover values of the smooth model fitted to the data calculate `valin - valout`.

`refine` **– fine tuning of fit** `Refine` returns data after subtraction of the best fit trigonometric polynomial, like to `fouw` and `prew`. However, to get the best fit `refine` both adjusts frequency *and* order of fitted series. It terminates on success of Fisher-Snedecor test for randomness of residuals.

5.6 Analysis in the time domain

The command `covar` serves for the calculation of the covariance and autocovariance functions. Matching ACF functions may be obtained for some data after some massaging.

`covar` **– Covariance analysis:** This command computes the discrete covariance function for unevenly sampled data. Alexander (1997) method is used for the estimation of the cross correlation function (CCF) of unevenly sampled series. The binned covariance function is returned with its gaussian errors. Significant are the portions of the curve differing from 0 by more than a number of standard deviations. This command can also be used for the calculation of the autocovariance function (ACF) by simply using the same series for the two input data sets. Here one shifted series is used as a model for the other. The covariance statistic is used to evaluate the consistency of the two series. The covariance statistics is akin to the power spectrum statistics and hence to the χ^2 statistics (Sect. 4.2). The number of degrees of freedom varies among time lag bins. Thus, in order to facilitate the evaluation of the results, errors of the ACF are returned. The expected value of the ACF for pure noise is zero. The value returned for 0 lag corresponds to the correlation of nearby but not identical observations. This is so because the correlation of any observation with itself is ignored in the present algorithm, for numerical reasons. The correlation function for a lag identical to zero can be easily computed as the signal variance.

The statistical evaluation of TSA results rests on the assumption that the noise in the data is white noise. However, quite often this assumption is wrong. One way to test its justification is to compute the residuals from the model fit (e.g. by using `pyaov.fouw`) and to examine the correlation length in the residuals from the autocorrelation function (ACF; computed with `pyaov.covar`). The average number of observations per correlation length is the average number of correlated observations n_{corr} . For white noise this number should be of order 1.

5.7 Auxiliary utilities

The following commands implement auxiliary utilities for time series analysis:

`fgrid` – **Frequency grid:** The frequency grid suitable for the analysis of evenly sampled observations is well determined. However, for uneven sampling no simple rules exist in general. `pyaov.fgrid` may be used to find a reasonable guess for the frequency grid. The returned parameters of the frequency grid may be subsequently used in calls of the periodogram routines `amhw, aovw ....` `pyaov.fgrid` may err, as usual in guessing; its results must be checked for consistency.

`normalize` – **Normalize mean & variance:** Normalize mean and (optionally) variance of an input column to 0 and 1, respectively. Subtraction of the average value from the data is always recommended for numerical reasons. Certain commands will not work correctly for large mean values.

`peak` – **Find peak and return its parameters:** This simple routine fits a parabola to top 3 points of input vector and returns peak position and value as well as its half width at the continuum level. The continuum iscalculated as mediane of input.

`pldat` – **Plot time-value data:** pldat(time, value)

`plper` – **Plot periodogram and phase folded data:** plper(frmax, time, value, fre qs, th)

`post` – **Post-processing of periodogram** This auxillary routine finds periodogram peak and generates vector of frequencies.

`pre` – **Pre-processing periodogram input** This auxillary routine tests consistency of input to periodogram routines: matxhing sizes of arrays and the frequency grid.

`simul` – **Simulates data** A rather primitive routine to generate test data for checks of package integrity and demo runs.

`totals` – **Return general characteristics** of data, not depending on any fitted period.

6 Command summary

amhw `th,fr,frmax=pyaov.amhw(time,valin,error,fstop,fstep, nh2=3,fr0=0.)`
Compute multiharmonic analysis of variance periodogram (Sect. 5.5).

aovw `th,fr,frmax=pyaov.aovw(time,valin,error,fstop,fstep, nh2=3,fr0=0.,ncov=2)`
Compute analysis of variance periodogram (Sect. 5.5).

atrw `th,fr,frmax=pyaov.atrw(time,amplitude,error,fstop, fstep, nh2=30,fr0=0.,ncov=2)`
Calculate box-like function AOV periodogram for detection of planetary transits and eclipses (Sect. 5.5).

covar `lav,lmi,lmx,cc,cmi,cmx=pyaov.covar(t1,d1,v1,t2,d2,v2, nct=11,eps=0.0,iscale=0,ifunct=0)`
Compute discrete covariance function for unevenly sampled data (Sect. 5.6).

fgrid `fstop,fstep,fr0=pyaov.fgrid(time)`
Evaluate frequency band for time series analysis (Sect. 5.7).

fouw `fr,valout,cof,dcof=pyaov.fouw(time,valin,error,frin, bacgnd=0.,nh2=2)`
Fit ordinary Fourier series with adjustment of frequency,if required (Sect. 5.4).

f_mw `th,fr,frmax = pyaov.f_mw(tin, vin, er, fup, fstep, fr0=0.)`
Returns generalized L-S periodogram of data, using the original Ferraz-Mello (1981) algorithm.

normalize `y = normalize(x, error, mean=0.,var=1.)`
Normalize weighted mean and (optionally) variance to 0 and 1,respectively (Sect. 5.7).

peak `xm,fm,dx = pyaov.peak(fx)`
Evaluate spectral line width and profile (Sect. 5.7).

pldat `pldat(time,value)`
Plot time-value data (Sect. 5.7).

plper `plper(frmax,time,value, freqs,th)`
Plot periodogram and phase folded data (Sect. 5.7).

post `th, freqs, frmax = post(nam, th, nfr, fstep, fr0)`
Performs post-processing of periodograms, finds a peak.

pre `nfr = pre(nam, tin, vin, er, fup, fstep, fr0)`
Test consistency of input to periodogram routines.

prew `fr,valout=pyaov.prew(time,valin,error,frin, bacgnd=0.,nh2=2)`
Fit trig orthogonal polynomial series with adjustment of frequency, if required (Sect. 5.4).

pspw `th,fr,frmax=pyaov.pspw(time,valin,error,fstop, fstep, fr0=0.)`
Compute discrete power spectrum for uneven sampling by slow method (Sect. 5.4).

test_per `test_per(file)`
For package consistency test runs a sequence of periodogram routines and plots output.

totals `pyaov.totals(x)`
Prints some general characteristics of input vector (Sect. 5.7).

7 Examples

7.1 Period analysis

Let us assume that file `ex1.dat` contains observations of a periodic phenomenon.
Note that to move to the next step you have to close currrent plot.

```
from pylab import *
import aov as _aov
import pyaov
o=mlab.load('ex1.dat',comments='!',skiprows=0)
od=pyaov.normalize(o[:,1],var=0.)  # Subtract mean
fstop,fstep,fr0=pyaov.fgrid(o[:,0])# Find suitable frequency band
th,fr,frmax=pyaov.pspw(o[:,0],od*0.+1.,o[:,2],fstop,fstep)
                                   # Compute window
pyaov.plper(0.,o[:,0],od,fr,th) # Plot spectral window (time & value ignore
th,fr,frmax=pyaov.amhw(o[:,0],od,o[:,2],fstop,fstep,nh2=3)
                                   # Calculate periodogram
pyaov.plper(frmax,o[:,0],od,fr,th) # Inspect periodogram &
                                   # phase folded data
th,fr,frmax=pyaov.amhw(o[:,0],od,o[:,2],2.8,fstep/10.,nh2=3,fr0=2.7)
pyaov.plper(frmax,o[:,0],od,fr,th) # Inspect zoomed periodogram
#xm,fm,dx = pyaov.peak(fr)         # Find periodogram peak parameters
fr0,dfr,od1,cof,dcof=pyaov.fouw(o[:,0],od,o[:,2],frmax, nh2=2)
                                   #  Remove one oscillation from data
th,fr,frmax=pyaov.amhw(o[:,0],od1,o[:,2],fstop,fstep,nh2=3)
                                   # Periodogram of pre-whitened data
pyaov.plper(frmax,o[:,0],od1,fr,th)# Inspect this periodogram
```

7.2 Comparison of two stochastic processes

Let the two files `ex2a.dat` and `ex2b.dat` contain light curves of two images of a gravity lens, presumably similar yet shifted in time.

```
from pylab import *
import aov as _aov
import pyaov
from scipy.optimize import curve_fit
def acf(lag, *p):
   return p[0]+p[1]*np.exp(-0.5*(lag/p[2])**2)

oa=mlab.load('ex2a.dat',comments='!',skiprows=0)
ob=mlab.load('ex2b.dat',comments='!',skiprows=0)
oad=pyaov.normalize(oa[:,1],var=0.)  # Subtract mean from data A
obd=pyaov.normalize(ob[:,1],var=0.)  # Subtract mean from data B
lav,lmi,lmx,cc,cmi,cmx=pyaov.covar(oa[:,0],oad,oa[:,2], \
   oa[:,0],oad,oa[:,2],nct=20)       # Compute autocov. of A
pyaov.pldat(lav,cc)                        # Plot autocovariance of  A

p0 = np.array([0.,0.7,3.])
popt, pcov = curve_fit(acf, lav, cc, p0=p0)
print 'fitted ACF parameters:'
for i in range(p0.size):
   print popt[i],'+/-',np.sqrt(pcov[i,i])
lav,lmi,lmx,cc,cmi,cmx=pyaov.covar(ob[:,0],obd,ob[:,2], \
   ob[:,0],obd,ob[:,2],nct=20)       # Compute autocov. of B
pyaov.pldat(lav,cc)                        # Plot autocovariance of  B
lav,lmi,lmx,cc,cmi,cmx=pyaov.covar(oa[:,0],oad,oa[:,2], \
   ob[:,0],obd,ob[:,2],nct=20)       # Compute crosscov. of A and B
pyaov.pldat(lav,cc)                        # Plot cross-covariance
```

References

Abramovitz, M. & Stegun, I.A. 1972, Handbook of Mathematical Functions, Dover, New York.

Acton, F.S. 1970, Numerical Methods that Work, Harper & Row, New York.

Akerlof, C., Alcock, C., Allsman, R., et al. 1994, ApJ, 436, 787

Alexander, T., 1997, in Astronomical Time Series, Eds. D. Maoz et al., Kluwer, Dordrecht, ASSL, v218, p163

Bloomfield, P. 1976, Fourier Analysis of Time Series: An Introduction, Wiley, New York.

Brandt, S. 1970, Statistical and computational methods in data analysis, North Holland, Amsterdam.

Chatfield, C. 1985, The Analysis of Time Series: An Introduction, Chapman & Hall, London.

Deeming, T. J. 1975, Ap&SS, 36, 137

Dvoretsky, M.M. 1983, MNRAS203, 917.

Eadie, W.T., Drijard, D., James, F.E., Roos, M. & Sadoulet, B. 1971, Statistical methods in experimental physics, NorthHolland, Amsterdam.

Edelson, R.A. & Krolik J.H. 1988, ApJ333, 646.

Ferraz-Mello, S. 1981 AJ, 86, 619

Foster, G. 1995, AJ, 109, 1889

Horne, J. H., & Baliunas, S. L. 1986, ApJ, 302, 757

Gray, D.F. & Desikachary, K. 1973, ApJ181, 523.

Lafler, J. & Kinman, T.D. 1965, ApJS11, 216.

Lomb, N. R. 1976, Ap&SS, 39, 447

MIDAS Users Guide: 1992 November, European Southern Observatory, Garching.

Paltani, S. 2004, A&A, 420, 789

Press, W.H., Flannery, B.P, Teukolsky, S.A. & Vetterling, W.T. 1986, Numerical Recipes, Cambridge University Press, Cambridge.

Press, W.H. & Rybicki, G.B. 1989, ApJ338, 277.

Press, W.H. et al. 1992, ApJ385, 404.

Renson, P. 1978, A&A63, 125.

Roberts, D. H., Lehar, J., & Dreher, J. W. 1987, AJ, 93, 968

Scargle, J. D. 1982, ApJ, 263, 835

Schwarzenberg-Czerny, A. 1989, MNRAS, 241, 153

Schwarzenberg-Czerny, A. 1991, MNRAS, 253, 198

Schwarzenberg-Czerny, A. 1996, ApJ, 460, L107

Schwarzenberg-Czerny, A. 1997, ApJ, 489, 941

Schwarzenberg-Czerny, A. 1998, Baltic Astronomy, 7, 43

Schwarzenberg-Czerny, A. 1999, ApJ, 516, 315

Schwarzenberg-Czerny, A., & Beaulieu, J.-P. 2006, MNRAS, 365, 165

Stellingwerf, R. F. 1978, ApJ, 224, 953