

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Earth Scientist's Guide to Discrete-Time Power Spectrum Analysis

Bjørn-Gustaf J. Brooks

*Center for Climatic Research, University of Wisconsin-Madison
U.S.A.*

1. Introduction

Fourier methods are commonplace in the Earth Sciences and have greatly enhanced our understanding and forecast capabilities for cyclical phenomena that recur on interannual (e.g. El Niño, Pacific Decadal Oscillation) to millennial scales (e.g. Milankovitch cycles). Nowadays most low level programming languages (C, Fortran) have math libraries that include the fast Fourier transform algorithm (FFT) and nearly all abstract programs (Python, Octave/Matlab, IDL, R) provide an array of Fourier functions for scripting sophisticated signal processing routines. Whether your interest as a practicing Earth scientist is in Fourier transformation for efficient data manipulation, or for problems where the Fourier transform or its power spectrum is needed for direct analysis, you have probably found no shortage of relevant literature. Nonetheless, you may also have found some difficulty in making sense of *which* Fourier methods to implement for your particular analysis idea, and *how* to appropriately apply them. This chapter will serve you as a basic guide for unraveling some of the complicated implementations of discrete-time power spectrum analysis using direct language and supplementary Matlab/Octave routines using both observed and modeled data.

This chapter assumes that you have a certain task to accomplish, and therefore it is designed to teach you how to set up an approach appropriate for Fourier analysis, and also to advise you of potential pitfalls and limitations in Fourier analysis. The reader need not have prior exposure to signal processing methodologies, but should have a solid base in mathematics, probability theory and more importantly the issues related to your analysis data so that the significances of cycles within your data can be rationally interpreted. If you find yourself lost by the terminology I recommend you familiarize yourself with basic treatments of discrete-time systems, for example Press et al. (1992) or Cadzow (1973).

The body of this chapter is split into three sections, *Preprocessing data*, *Single Series Spectrum Analysis*, and *Multiseries Spectral Analysis*. Step-by-step examples are given on the analysis of a variety of freely accessible earth science datasets covering atmospheric science, biosphere-atmosphere carbon cycling, climate modeling, and paleodiversity as well as some example implementations of Markov chain Monte Carlo routines for computing statistical significances. Each section contains direct explanations with ready to deploy example code that you are free to use for your own investigations. Supplementary code can be accessed online from <ftp://ftp.climatemodeling.org/pub/esg/>.

2. Preprocessing data: Identifying the pertinent information

An often overlooked aspect in the early stages of data analysis and exploration is the preliminary processing of time series. No matter what your particular topic is it is often desirable to filter measurements that confound your analysis or do not contribute useful information. A thorough exploration of the many methods for reducing noise and removing trends prior to computing the Fourier power spectrum may seem like a lot of work, but it will be effort well-rewarded because it will increase confidence in your results. Even data that has already been heavily processed, for example satellite products like leaf area index (LAI), may still require treatment beyond, or in stead of, monthly averaging, seasonalizing, or annualizing. This section discusses filtering, subsetting, and detrending topics and provides example code for deploying this in your own data.

Within any dataset there are data that contribute useful information as well as data that do not, or even confound your analysis. One example comes from the study of atmospheric tracers of photosynthesis such as carbon dioxide or carbonyl sulfide. Atmospheric CO₂ and OCS have diurnal and annual cycles that are strongly driven by biotic processes, primarily reflecting the uptake of atmospheric CO₂ by plant photosynthesis and CO₂ release through heterotrophic and autotrophic respiration. Measurements of these tracers should show stronger annual or diurnal cycles during years when precipitation is substantially influenced by El Niño. But strong cycles may be difficult to observe unless the observations are preprocessed or filtered. This is because a series of tower measurements of atmospheric trace gasses will include both locally representative observations made when turbulent mixing is low and winds are calm, and regionally representative observations made when the boundary layer is well-mixed. If your goal were to observe smaller influences caused by interannual ENSO cycles among much larger biotic influences you may have to detrend from a fitted polynomial (section 2.1) filter the noise (section 2.2), or subset the data to remove locally representative measurements (section 2.3), which would result in non-uniform series that can be analyzed using Lomb-Scargle (section 3.2).

2.1 Detrending

Within geological time series, from ice cores to cyclostratigraphy, it is not uncommon to find long term and systematic trends. If your objective is to understand something other than these trends it will be necessary to detrend the data. An example of the importance of detrending was observed by Cornette (2007) who showed that the prominence of a 62-million-year cycle in extinctions recorded in the geologic rock record was strongly affected by detrending the original diversity time series. This is because the most significant variability was caused by a large non-linear biodiversification trend (Figure 1A) representing three major phases of biodiversification. Extinction cycles were only apparent in the Fourier power spectrum after detrending the time series. Likewise if a very long lived and significant harmonic or trend exists in your data detrending should be considered. Pseudo-code showing how this fossil diversity time series was detrended appears in Procedure 1 (and also in supplementary code `dftps.m`).

2.2 Noise and smoothing filters

Noise adds considerable clutter that can confound Fourier power spectrum (PS) analysis and reduce the strength of important cycles. Noise is given a broad definition here to refer to variability in any series that cannot be resolved or directly tied to a predictable physical cause.

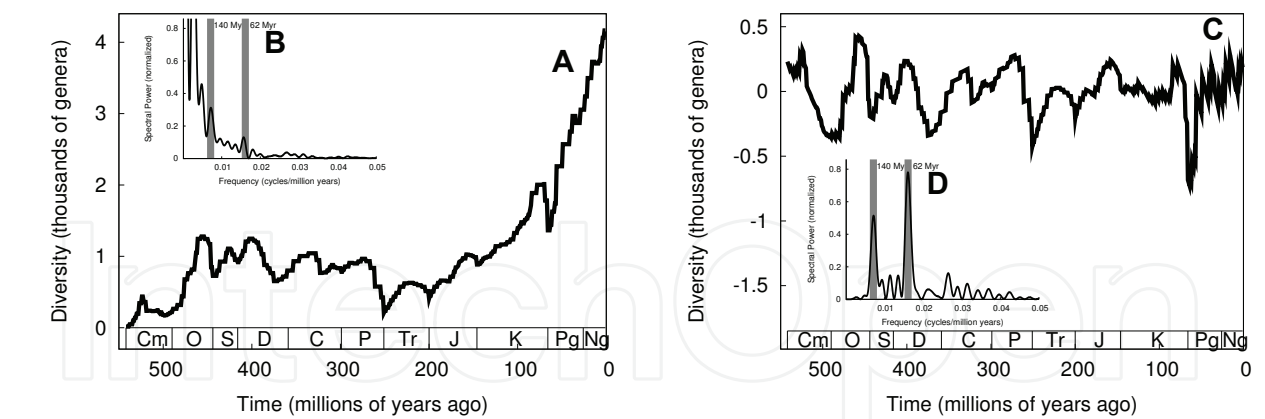


Fig. 1. Detrending effects. Subplots A and B show diversity without detrending– C and D are detrended. A) Time series of Sepkoski’s marine fossil diversity covering the past 542 million years, as reproduced from Rohde & Muller (2005). B) The Fourier power spectrum (PS) of that series. C) Time series of diversity residuals obtained from A by detrending by a cubic polynomial (*cf.* Cornette, 2007). D) The PS of residual diversity showing significant cycles corresponding to the 62 and 140 million year cycles. These show that in non-detrended fossil diversity nearly all the power of the 62 and 140 million year cycles is subsumed by the larger trend in A.

Procedure 1 General procedure for detrending a series by a polynomial curve of order N . The diversity data in Figure 1A was detrended by a polynomial of $N = 3$ to produce Figure 1C. Series x is comprised of $nterms$ measured at increments of t .

```
input: data series,  $\{(t_k, x_k)\}_{k=1}^{nterms}$ 
output: data series of residuals,  $\{r_k\}_{k=1}^{nterms}$ 

p = poly_fit(t, x, N)
pv = poly_val(p, t)
r = x - pv
```

It is also important to keep in mind that noise may simply reflect events that are not well represented by the particular measurement system or point of observation being used. Noise generally has a limited systematic effect and should be filtered if at all possible. A consideration when filtering noise is to choose a filter appropriate to the type of noise within your data (*i.e.* white, red, blue, gray). White noise can give the Fourier power spectrum plots (like those in Figure 1B and 1D) broad spectrum noise, which appears as peaks (all of similar magnitude) dispersed across a range of frequencies in the power spectrum. Least squares filters including Savitzky-Golay are often used to reduce this kind of broad spectrum noise. In Matlab/Octave such filtering is achieved by calling the `sgolayfilt` function. In IDL calling `savgol` in conjunction with `convol` achieves the same purpose. Confounding variability can also come from red noise, which is not uncommon in paleoclimate datasets and some long term atmospheric and ecosystem records. Power spectrum plots of series containing red noise have characteristic slopes that diminish with increasing frequency (toward the right of the spectral plot), resulting in power spectrum plots that appear cluttered on left near longer cycles (*i.e.* where red wavelengths of the

visible spectrum would appear). Red noise can be caused by quasi-stationary processes, for example a precipitation record with repeated and prolonged multi-year droughts that cause precipitation to fall below the mean. Red noise can be smoothed by filters that use lower order polynomials, for example in IDL using `savgol` with a degree of 2.

Blue noise causes clutter around shorter cycles toward the right of the power spectrum, and can be filtered using higher order polynomials. Grey noise is identified by its bimodal noise peaks at both the 'red' and the 'blue' end of the power spectrum. Several test cases are provided in the supplementary Matlab/Octave code `noise_filt.m`, and you should use them to test the effectiveness of the Savitzky-Golay filter using various filter configurations.

If however you find that it is not possible to effectively filter noise without reducing the information content of your time series, you may instead evaluate the power spectrum of the unfiltered data against a stringent statistical test such as a Markov chain Monte Carlo test (see the random walk test for red noise in section 3.3).

2.3 Subsetting and subsetting filters

Where noise filtering is not tenable or where it is desirable to search for the sources of cycles you should consider partitioning your time series based on the relative contributions of various groups (*subsetting*) or applying a *subsetting filter* to reject entire observations. Here I use the term subsetting to indicate partitioning of each value, as opposed to subset filtering, which I use to indicate rejecting entire observations that exceed some cutoff. Subset filtering would be relevant for analyses such as in atmospheric observation where it is not appropriate to smooth a biased measurement because ties to the actual observations are required.

A common issue with observational data is that not all measurements communicate useful information about the processes you are interested in, therefore it may be necessary to partition the complete set of observations in some way and to compare the significances of cycles between subsets. One example subsetting strategy is to partition each value in the series based on the relative contributions from different groups. Given that you also possessed meta-data, *i.e.* the annotations describing each specimen, you could begin to address some very interesting questions about the nature of cycle. As it happens, the diversity values used to construct Figure 1A can be partitioned by groups such as deep water fossils vs. shallow water fossils, or hard-shelled fossils vs. all others. Now we might ask, 'Is any one group of organisms responsible for the cycles in fossil diversity seen in Figure 1D?'

Figure 2 uses a similar but more comprehensive dataset of fossil diversity downloaded from the Paleobiology Database (PD, 2008) to subset the total diversity into multiple curves representing the contributions from each phylogenetic group, such as mammals, birds, and reptiles. Figure 2A compares the diversity curve of several dominant shelled fossil animals (gastropods, bivalves, and articulate brachiopods or GBA) against all other phyla. Figure 2B shows the Fourier power spectra of those same subsets and suggests that the 62 million year cycle in this dataset is largely driven by the GBA group.

On the other hand meta-data annotating each observation may not be available to you, and you may have to create a subsetting filter. Here you would make a decision, for example using a statistical basis to reject observations from a subset. For example, given a dataset of wind speed measurements from a meteorological station we might use a subset filter to reject all measurements of wind speeds less than 4 meters per second, leaving us with a new subset of unevenly spaced measurements. Non-uniformly sampled data cannot be analyzed using

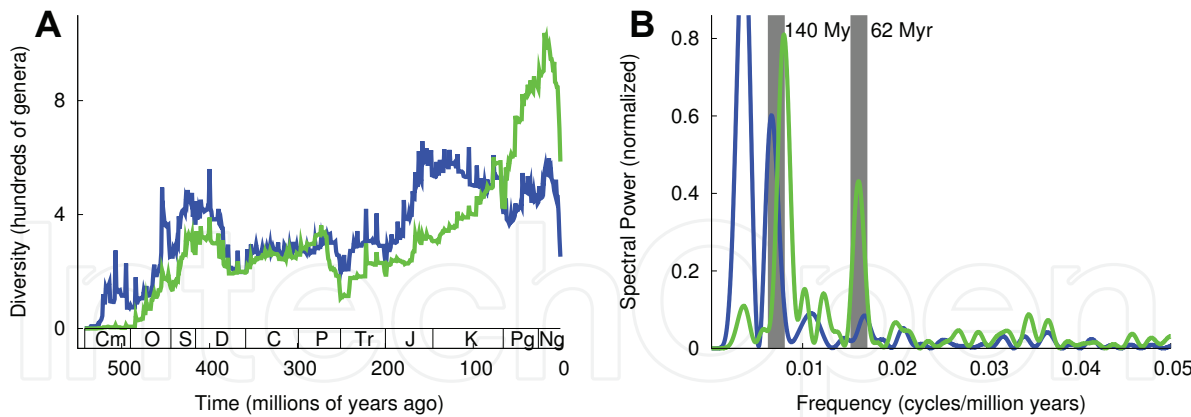


Fig. 2. Using metadata to subset by groups. A) shows two time series of fossil abundances derived from the Paleobiology Database (paleodb.org) and ranging from 542 million years ago to present. The first subset in green is comprised of gastropod, bivalve, and articulate brachiopod fossils (GBA). Their complement (non-members of that group) appear in black. B) shows the Fourier power spectrum of those same curves when detrended. Notice that a rather significant peak appears in the GBA group near the 62-million-year cycle frequency, which is not present in the non-GBA group. This shows that by using the annotations of your data it is possible to investigate the sources of cycles.

standard Fourier decomposition, but can be processed using the Lomb-Scargle method, which will be addressed later in Section 3.2).

Subsetting filters utilize *a priori* knowledge to reject outliers, which means you decide based on your analysis of the dataset what an acceptable cutoff should be. One way to avoid criticisms on the subjectivity of your cutoff would be to process your data using a series of cutoffs that range from very inclusive to very specific, and to do spectral analysis on each differently filtered subset, and compare.

2.4 Normalizing

A final but critical note on preprocessing is about the importance of normalizing your time series to a standard deviation of 1 prior to Fourier decomposition. Whether you apply filters to your data or not it is always necessary to normalize the variance of your original time series to make it comparable to other series. This is simple to do. Divide each observation by the standard deviation of the time series. See lines 21–22 in supplementary code `dftps.m`.

3. Single series spectrum analysis

Once your series of uniformly sampled values has been normalized (perhaps even filtered or subset) Discrete-time Fourier transform power spectrum analysis (DFTPS) can be used as a powerful tool for determining the relative strength of cycles within your time (or spatial) series. You should be aware that Fourier transformation works by decomposing a series into its complex conjugate of *real* (\Re) and *imaginary* (\Im) signal parts. The spectral power is computed as the square of the signal power from \Re , which produces a series of amplitudes across a range of frequencies that describe the strength of cycles. \Im is used to determine the phase of the cycles, which can tell you when in your time series the peaks of cycles of different frequencies should occur. These methods can be applied to any series of uniformly sampled

data. If your interest lies in the theory behind spectral analysis of discrete time systems a good place to start is Cadzow (1973) or Bloomfield (1976). More sophisticated implementations of signal processing designed for specific purposes (*e.g.* Brooks, 2009) are often useful when computational power is limited. But if you have access to a high performance machine or cluster then you will probably be able to solve big data problems easier using distributed memory parallelism.

3.1 Computing the power spectrum

As opposed to continuous analog signals, discrete time series are segmented and require a special step in treatment before transformation by the fast Fourier transform algorithm (FFT). You must pad your series of normalized values with zeros so that the transformed signal of your discretely sampled values is sufficiently long to allow for measurement of its peaks across a range of frequencies (see lines 18-19 of `dftps.m`).

Zero padding is particularly important when examining cycles with longer periods. Consider that you have 100 years of monthly means from a model and you want to know how well the model is able to reproduce interannual variability, caused for example by El Niño–Southern Oscillation (ENSO, 2-7 yr. period) or the Pacific Decadal Oscillation (PDO, 20-30 yr. period). Despite having cycle lengths that are roughly 20 years apart (ENSO: ~5 years, PDO: ~25 years), these cycles will show up very near each other in frequency on the power spectrum plot. Since frequency is the inverse of time $f_{\text{ENSO}} = \frac{1}{12} \text{ (months)} \times 5 \text{ (years)} = 0.0167$, which is near $f_{\text{PDO}} = \frac{1}{12} \times 25 = 0.0033$. In order to resolve such small frequency differences between longer cycles it is necessary to pad the signal with zeros. To better understand this you might try commenting-out line 18 of `dftps.m` and substituting it with: `lps=2^8`; (you will also have to adjust the y-axis range). The subsequent plot will be very coarsely sampled on the left toward longer frequencies because you reduced the padding. Zero padding will not change the amplitude of your peaks but it will help you better resolve the frequency and timing of cycles. Fortunately Matlab, Octave, and Python provide handy methods for padding the series and computing the Fourier transform in one step using the `fft` function (Procedure 2). The computation of uniform increments of f to plot your power spectrum peaks against is also straight-forward (see line 35 of `dftps.m`).

Procedure 2 General FFT procedure for Matlab, Octave, Python. Given the series x of discrete values x_k the function `fft` is used to compute its complex conjugate meanwhile ‘stretching’ the signal to length ne , provided that $ne > nterms$.

input: time series, $\{x_k\}_{k=1}^{nterms}$

output: x_ft (complex conjugate with \Re & \Im parts), sp (spectral power), f (frequency), p (period)

```
x_n = x/std_dev(x)           # Normalize to variance of 1
x_ft = fft(x_n,ne)           # Transform and pad to length ne
sp = x_ft .* conj(x_ft)      # Compute spectral density curve. sp is
                             # complex and contains real and imag parts
f = flt_intgen(ne)/ne        # Generate sampling frequencies for sp
p = 1/f                       # Cycle periods in original units
```

Remember that the FFT algorithm will give you complex output: a real part (\Re), which you will use to make your power spectrum plot, and an imaginary part (\Im), which you might use to determine the phase-synchronization. The imaginary part is not to be overlooked. In order to determine causality relationships for example, the phase data (\Im) of the independent and dependent cycles should coincide in a logical way. Computation of the phase is discussed in Section 3.4 and Procedure 3.

3.2 Non-uniformly sampled data

The FFT algorithm requires uniform time steps, and cannot be used to analyze non-uniform series such as in Figure 3A. Fortunately there are alternative ways for dealing with unevenly sampled data including the Lomb-Scargle (LS) method. The LS approach is a common implementation, but you should be aware that estimating the period of cycles from non-uniformly sampled data is not a trivial issue, and precision can vary depending on the frequency estimators used. An in-depth treatment of spectral analysis for unevenly sampled data appears in Press et al. (1992), which should be referred to for a detailed explanation.

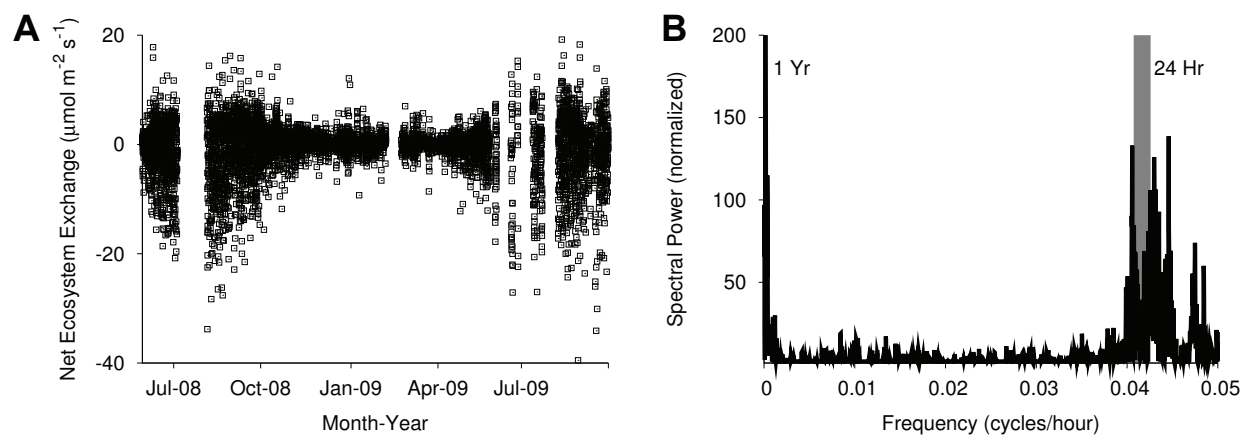


Fig. 3. Series that are not uniformly sampled can be transformed using Lomb-Scargle. A) Non-uniformly sampled time series of CO₂ exchange between the atmosphere and biosphere (Net Ecosystem Exchange) measured at the Park Falls, Wisconsin tall for June, 2008-Oct., 2009 (UW-Madison, 2010). B) The spectral power of that data when the LS method is used.

LS implementations (such as the example in supplementary code `lsps.m`) work by computing the spectral power across an increasing set of frequencies (usually controlled by a variable called `ofac`), up to a frequency limit (`hifac`). The resulting power spectrum represents an oversampling of the data (suitable results can be obtained by oversampling rates where `ofac` ≥ 4). `hifac`, which is related to the Nyquist frequency, sets the limit of frequencies to be explored. If, for example, you are not interested in frequencies larger than $f = 0.05$, then you should conserve computational time by setting `hifac` = 0.1.

Some considerations should be kept in mind when specifying your `ofac` resampling rate. If you are investigating two narrowly spaced cycles and your series has relatively few broad gaps, then you would probably be justified in using higher values for `ofac` that will allow you to resolve closely spaced cycles. On the other hand if your series is subject to substantial gaps, large values for `ofac` are probably not a good idea.

Figure 3B is a power spectrum plot, the same as in Figures 1B, 1D and 2B, except that this one was computed using the LS method. Figure 3B illustrates an important issue common to

all power spectra: the dispersion of peaks reflecting a cycle with a period that slightly varies throughout the time series. Minor variations in shorter frequencies, such as those to the right ($f > 0.03$) in Figure 3B, can result in either a broader peak or multiple peaks near the frequency of interest. The primary diurnal cycle should occur at $f = 0.0416$ in Figure 3B. Instead there are actually three peaks of nearly equal amplitude, which indicates that the timing of the diurnal NEE cycle pictured in Figure 3A varies throughout the time series, probably due to changes in daylight length throughout the year as well as synoptic frontal passages and other variability that can affect the exchange of carbon (NEE). For example, small shifts in timing of peak NEE by 1 hour in either direction (e.g. 23-hour/25-hour spacing) would result in large spreads in frequency ($f = 0.04$ to 0.045).

In spectral plots longer cycles have fewer opportunities to occur in a limited dataset and therefore the certainty about their significance is generally less than a shorter cycle of similar amplitude. The next section will discuss a method for dealing with uncertainty in the significance of longer-term cycles using Monte Carlo methods.

3.3 Estimating uncertainty using Markov chain Monte Carlo trials

By this point in the chapter you already know how to identify whether or not a cyclical pattern exists within your series. If you have already used a script to produce a power spectrum plot (e.g. `dftps.m`, `lps.m`), you probably asked yourself: 'How do I know if these cycles are significant?' There are a variety of ways to estimate the statistical significance of cycles within your data and many of them involve Monte Carlo trials. Markov chain Monte Carlo methods (MCMC) are a widely used class of algorithms that iteratively and randomly resample (permute) time series, and can be used to test the significances of cycles by comparing them to randomly derived cycles. MCMC tests allow you to compare the original time series to many randomized versions of that same data, and to examine whether or not cycles of equal or greater magnitude exist in the randomized versions. It is important to understand, however, that Monte Carlo significances do not rule-out bias in your sampling protocol. Significance tests of this kind can only inform you about how likely a cycle is to occur given a particular collection of values.

As a starting point for understanding Markov chain Monte Carlo methods I present an example called random walk trials, sometimes referred to as random step or drunkard's walk. Random walk trials can be used to determine statistical significances and are particularly useful for evaluating low frequency cycles. Many random walk implementations exist, and here we will keep things simple by using just one implementation, which appears in supplementary code `dftps_mcmc.m`.

If you have experimented with the `dftps_mcmc.m` routine you may have wondered how it computes significances. The significance calculation used in `dftps_mcmc.m` is described as follows: For a given peak of height h at a frequency f , the significance of a peak can be computed as the fraction, p , of N randomly generated sequences of the original series, from which the spectrum at f exceeds h . Here you might report the significance of a cycle as the fraction of trials in which the height of the peak in the original series exceeded the height from the randomly generated series. Since random walk trials work by randomly permuting the original series before Fourier decomposition, 10,000 MCMC trials would represent the null hypothesis that cycles within your data are merely coincidental and can be reproduced given sufficient randomization.

Uncertainty can be represented not only as a percentage but also in your power spectrum plot by a line, for example representing the mean spectral power of the MCMC iterations. Supplementary code `dftps_mcmc.m` outputs both the significance estimates and a power spectrum plot with the MCMC random walk significance curve. Note that all peaks that occur at or below the random walk significance curve should be interpreted as insignificant, whereas peaks with amplitudes greater than the curve are significant to the degree indicated by the statistics of the random walk trials.

3.4 Phase synchronization and phase shifted cycles

Two cycles (from different time series) with exactly the same period might seem to be related. However, their phases could indicate that their peaks in the original time series are asynchronous, which could indicate a lag in the cause-and-effect relationship, or they may not be related at all. Let’s say that you have two time series of daily means, one for air temperature and another for the temperature of a nearby lake. Both records will have strong annual cycles, near $f = \frac{1}{365} = 0.0027$, but because of the specific heat capacity of water, the peaks in lake temperature will lag behind air temperature. If you examined the imaginary part (\Im) of the Fourier transform at $f = 0.0027$ you would see the phase shift of the cycle in the range $-\pi : \pi$ radians. (For example find the index number of the element in `f` on line 29 of `dftps.m` that is closest to the frequency 0.0027 and then find the corresponding element by its index in the imaginary part of `xn_fft` on line 19 of `dftps.m`) A good way of visualizing this is to plot the original series along with a sine wave corresponding to the cycle of interest with the appropriate phase shift as determined from the imaginary part of the Fourier transform. This is readily done in Matlab/Octave using `angle`, and Python using `phase`, and IDL using `atan`, which compute the phase angle shift in radians ($[-\pi : \pi]$). Pseudocode describing the construction of the phase shifted sine wave appears in Procedure 3. Note also that this can be used to extrapolate the continuation of a cycle beyond the observational data, which might be useful when predicting future cycle peaks.

An example that should give us no trouble comes from modeled temperature data from the Parallel Climate Model (Washington et al., 2000). Because the complex transform corresponds perfectly to the magnitude and phase, we need only to convert the real and imaginary parts back into physical coordinates according to Procedure 3 in order to produce Figure 4.

Procedure 3 General procedure for determining the phase shift and computing the sine wave corresponding to the period and phase of the cycle of interest. This procedure uses the imaginary data at index `k` to compute the phase shift (`ps`) for constructing the sine wave.

input: $\{x_{ft}\}_{i=1}^{nterms}$ (complex conjugate); `p` (period); `t` (time increments in physical units); `a` (sine wave amplitude in physical units)

output: `ps_k`, `p_k`

```
ps = atan(x_ft[k])
sw = a * sin(2 * (pi/p) * t + ps) # the phase shifted sine wave
```

4. Multiseries spectrum analysis

Cause and effect relationships are common, but sometimes difficult to verify. One way to test whether cycles with the same period from two different time series are related is

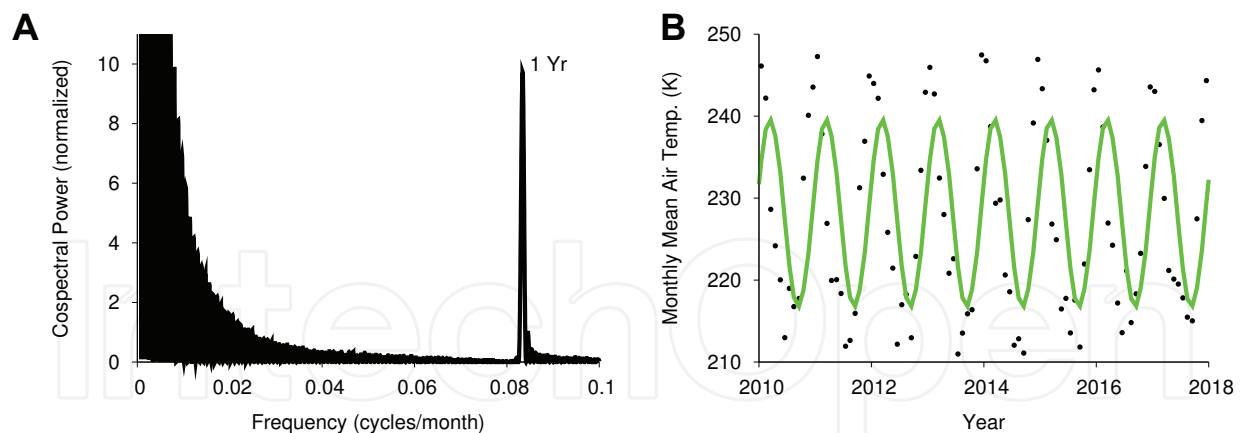


Fig. 4. Power spectrum plot (A) and time series with phase shifted wave (B). The phase shift of the annual cycle in A is applied to the green sine wave in B, which shows where the peaks and troughs of the annual cycle should occur according to their average occurrence over 100 years of model data.

through multiseries spectrum analysis. Cross-spectrum analysis is a common implementation used to compare how well cycles from two different series covary in period and phase synchronization. Think of cross-spectrum analysis as being the Fourier transform equivalent of cross correlation. For example, you would expect that net radiation measured at a meteorological station would correlate well with air temperature. Cross-spectrum analysis should reveal diurnal and annual cycles that are strong and nearly synchronized.

As opposed to the results of single series spectrum analysis the cross-spectrum density represents the covariance between two series. Cross-spectrum functions in abstract languages like Matlab typically compute spectrum density using the Fourier method or Welch's method, and their power spectra can be interpreted in the same way, except that the phase data takes on a heightened importance. Figure 5 was produced using supplementary code `csps.m` again using model data for years 2000-2099 from the Parallel Climate Model (Washington et al., 2000). Figure 5A shows the cross-spectrum density of soil moisture and precipitation for one arctic land surface grid cell, which signals a very strong correlation in both series of the annual cycle. Figure 5B represents the phase shift across all frequencies. The frequency of the annual cycle is located by the vertical gray band, which indicates that the two annual cycles are nearly π radians (180°) out of phase, meaning that precipitation is high when soil moisture is low. This seems surprising at first, but remember that in the arctic a majority of the precipitation comes as snow, which has a delayed release into soils or the soils may be frozen (permafrost) for most of the year.

5. Summary

There are many worthwhile uses for discrete-time Fourier power spectrum analysis methods, as shown in this chapter including: 1) correlating the amplitude and phase of similar cycles from different time series, 2) estimating the statistical significance of cycles, and 3) investigating the sources of cycles using subsets of the complete set of data. However, by now you have probably also realized a few of the limitations of power spectrum analysis such as: 1) very long cycles are difficult to detect given time series of limited length, 2) inappropriately preprocessing/filtering your data prior to Fourier decomposition

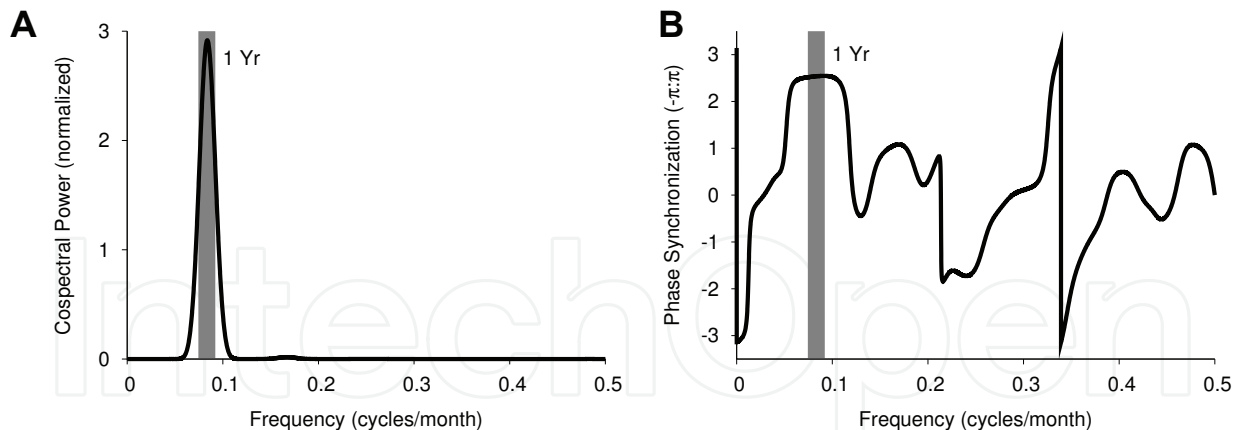


Fig. 5. Cospectrum and phase synchronization. A) shows the total power of the cospectrum representing a strong cycle in both soil moisture and precipitation. The gray band locates the annual cycle. B) shows that the phases of the annual cycles are nearly 180° (π radians) out of phase, which underscores the importance of considering the phase data in cross-spectrum analysis.

can lead to erroneous conclusions, 3) choosing a noise filter is subjective and can lead to different results, and 4) failing to specify the correct null hypothesis to test against possible cause-and-effect relationships between cycles (*i.e.* type iii error) can lead to false conclusions. Understanding the strengths and weaknesses of Fourier power spectrum analysis as you have been introduced to here will help you to place the correct emphasis on signal processing results in your study. If you do find room for such analysis in your work feel free to implement the supplementary code presented here without reservation and in any way you like.

6. References

- Bloomfield, P. (1976). *Fourier analysis of time series: An introduction*, John Wiley & Sons Inc.
- Brooks, B.-G. J. (2009). *Applying Wavelet and Fourier Transform Analysis to Large Geophysical Datasets*, Vol. 5545/2009, Springer Berlin, Heidelberg, pp. 426–434. doi: 10.1007/978-3-642-01973-9_47.
- Cadzow, J. A. (1973). *Discrete-Time Systems: An Introduction with Interdisciplinary Applications*, Prentice-Hall, Inc.
- Cornette, J. L. (2007). Gauss-vaníček and fourier transform spectral analyses of marine diversity, *Computing in Science and Engineering* 9(4): 61–63. doi: 10.1109/MCSE.2007.76.
- PD (2008). The Paleobiology Database, <http://www.paleodb.org/>. Data were downloaded on 8 August, 2008. Ichnotaxa and open nomenclature genera were excluded. The collections reflect 8,821 publications. Lithological categories can be found online at <http://paleodb.org/public/tips/lithtips.html>.
- Press, W. H., Flannery, B. P., Teukolsky, S. A. & Vetterling, W. T. (eds) (1992). *Spectral Analysis of Unevenly Sampled Data*, Cambridge University Press, 2 edition, Cambridge, p. 992.
- Rohde, R. A. & Muller, R. A. (2005). Cycles in fossil diversity, *Nature* 434: supplementary material 1. <http://www.nature.com/nature/journal/v434/n7030/extref/nature03339-s3.xls>.

- UW-Madison (2010). University of Wisconsin Park Falls tall tower NEE data, Website. Data were downloaded on 24 January, 2011 from http://flux.aos.wisc.edu/data/wlef/flux/2008/prefnee_2008.txt and http://flux.aos.wisc.edu/data/wlef/flux/2009/prefnee_2009.txt.
- Washington, W. M., Weatherly, J. W., Meehl, G. A., Semtner, Jr., A. J., Bettge, T. W., Craig, A. P., Strand, Jr., W. G., Arblaster, J., Wayland, V. B., James, R. & Zhang, Y. (2000). Parallel climate model (PCM) control and transient simulations, *Climate Dynamics* 16: 755–774. doi: 10.1007/s003820000079.



Fourier Transforms - New Analytical Approaches and FTIR Strategies

Edited by Prof. Goran Nikolic

ISBN 978-953-307-232-6

Hard cover, 520 pages

Publisher InTech

Published online 01, April, 2011

Published in print edition April, 2011

New analytical strategies and techniques are necessary to meet requirements of modern technologies and new materials. In this sense, this book provides a thorough review of current analytical approaches, industrial practices, and strategies in Fourier transform application.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Bjørn-Gustaf J. Brooks (2011). Earth Scientist's Guide to Discrete-Time Power Spectrum Analysis, Fourier Transforms - New Analytical Approaches and FTIR Strategies, Prof. Goran Nikolic (Ed.), ISBN: 978-953-307-232-6, InTech, Available from: <http://www.intechopen.com/books/fourier-transforms-new-analytical-approaches-and-ftir-strategies/earth-scientist-s-guide-to-discrete-time-power-spectrum-analysis>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2011 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen