

5 Time Series Analysis

5.1 Introduction

5.2 Generating Signals

5.3 Fourier Transforms FT, DFT and FFT

Furthermore, the data $x(t_k)$ are only available in a limited interval, that is, between t_1 and t_n . The *discrete Fourier transform* (DFT) –as opposed to the continuous Fourier transform (FT)–converts such a time series $x(t_k)$ into a discrete Fourier (or frequency) series $X(f_k)$:

$$X(f_k) = \sum_{k=0}^{n-1} x(t_k) e^{2\pi i f t_k / f_s}$$

In some fields, the *power spectral density* (PSD) is used as an alternative method for describing the power spectrum, whereby the spectral power is divided by the sampling frequency f_s and by the length of the time series n .

```
for i in range(0,T.size):
```

We can use the FFT to calculate a Fourier spectrum of the signal, which is displayed as the spectral **power** against the frequency f and the period τ (Fig. 5.6). The function `fft()` computes the Fourier transform for `nfft` frequencies, where `nfft` is the **smallest** power of two **that is greater than or equal** to the **data** length of the series x and is padded with zeros in order to make up the number of data points equal to the value of `nfft`.

```
plt.grid()
```

```
ts = np.linspace(0,14.4,10)
```

5.4 Schuster's Periodogram Method

for the Nyquist range $-1/2 f_s < f < +1/2 f_s$, where f_s is the sampling frequency and n is the length of the time series.

```
Ns = 2.5  
Fs = 2
```

```
xt = x + 0.005*t
```

5.5 The Blackman–Tukey Method and Welch's Method

```
Fs = 1  
  
plt.grid()
```

5.7 The Spectrogram (Sonogram, Sonograph) Method

```
plt.grid()
```

5.8 Thomson's Multitaper Method

```
Ns = 3
```

5.9 The Lomb–Scargle Power Spectrum

However, this number can also be replaced by a vector of several confidence levels, such as `confid=np.array([0.9,0.95,0.99])`. We can now type

We use `normalize=True` as the spectrum type in order to obtain a Lomb–Scargle normalized periodogram, which is normalized by dividing `px` by `2*var(x)`, as in our Python code above.

5.10 The Wavelet Power Spectrum

Note that it is important not to use `j` for an index in `for` loops since it is used here for the imaginary unit (Fig. 5.21):

5.11 Nonlinear Time Series Analysis (by N. Marwan)

```
S = distance.squareform(D)  
  
n = len(x)  
tau = 2  
m = 2
```

5.12 Detecting Abrupt Transitions in Time Series

We use the function `ranksums()` introduced in Chap. 3.11 to perform the Mann–Whitney test (Fig. 5.35 a):

We use the function `ansari()` introduced in Chap. 3.12 to perform the Ansari–Bradley test.

5.13 Describing Gradual Transitions in Time Series

Such a data set can be generated with the following **Python** script:

Recommended Reading

Figure Captions

Fig. 5.11 Blackman–Tukey auto-spectrum of a signal. **a** The autocorrelation series `corrxx` of a time series `x` is calculated using `autocorr()`, and **b** the power spectral density `Pxx` is calculated from the Fourier transform `XXX` of the autocorrelation series `corrxx` using `fft`.

Fig. 5.13 Interpolation methods. In this example, a 5th-degree polynomial was sampled at eight unevenly spaced points and then resampled using different interpolation methods provided in `interp1p()`. **a** Comparison of linear interpolation, nearest neighbor interpolation, and spline interpolation as method in `interp1p()`. **b** Spline interpolation often causes spurious amplitudes in the interpolated time series when there are steep gradients in the data sequence. In these cases, `PchipInterpolator` offers a good alternative that performs a shape-preserving piecewise cubic interpolation.

Fig. 5.17 Evolutionary power spectrum using `spectrogram()`, which computes the short-time Fourier transform (STFT) of overlapping segments of the time series. We use a Hamming window of 64 data points with a 50 data point overlap. The STFT is computed for `nfft=256`. Since the spacing of the interpolated time vector is 3 kyr, the sampling frequency is $1/3 \text{ kyr}^{-1}$. We use `pcolormesh()` to display the output of `spectrogram()` in a color plot that displays yellow vertical stripes, which represent major maxima at frequencies of 0.01 and 0.05 kyr^{-1} , i.e., every 100 and 20 kyr, respectively. The plot shows the onset of the 40 kyr cycle at around 450 kyr.

Fig. 5.20 Evolutionary Lomb–Scargle power spectrum. We use a window of 50 data points with a 1 data point overlap. We use `pcolormesh()` to display the spectrum in a color plot that displays yellow–green horizontal stripes, which represent major maxima at frequencies of 0.01 and 0.05 kyr^{-1} , i.e., every 100 and 20 kyr, respectively. The plot shows the onset of the 40 kyr cycle at around 450 kyr.