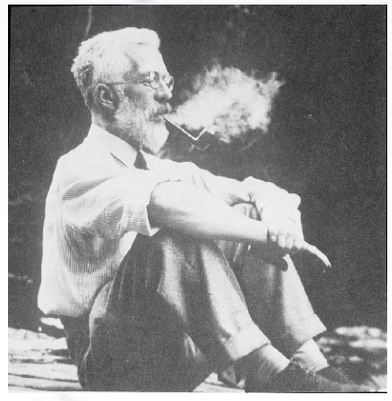
Applied Statistics

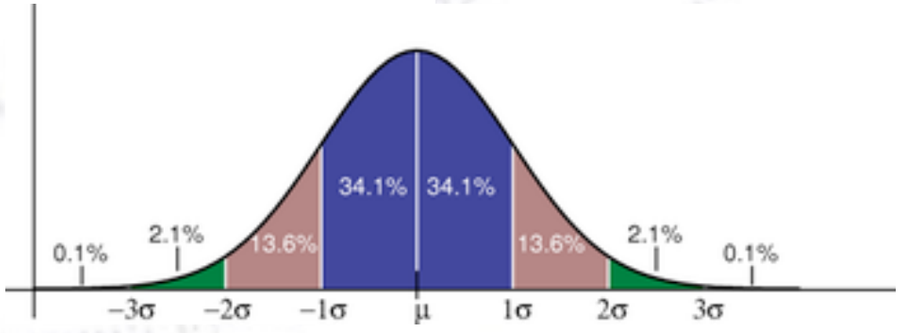
Time series analysis








Mathias Spliid Heltberg (NBI)



"Statistics is merely a quantisation of common sense"

Overview

By the end of the lecture, I hope you will know:

1) Why errors are defined by: $\min(\chi^2)+1$

2) How to detrend data with polynomial fitting

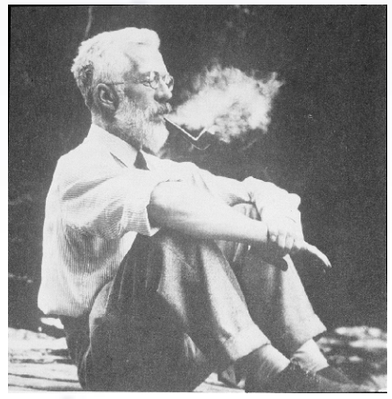
3) What comes out of the (Fast) Fourier Transform and how to apply it

4) The existence of Autocorrelation, Time Warping and Time Embedding.

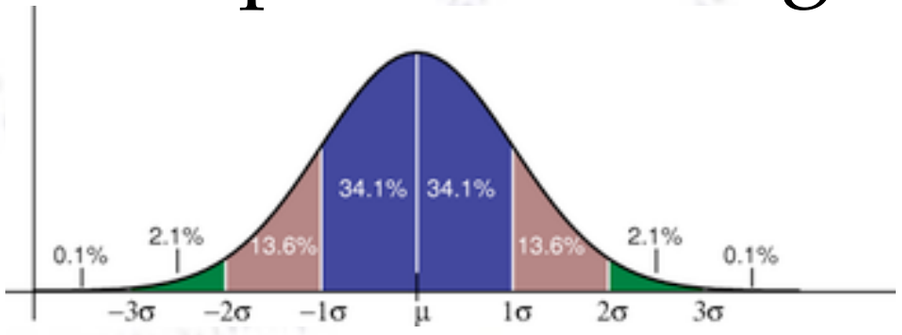
Applied Statistics

Where do fit errors come from?



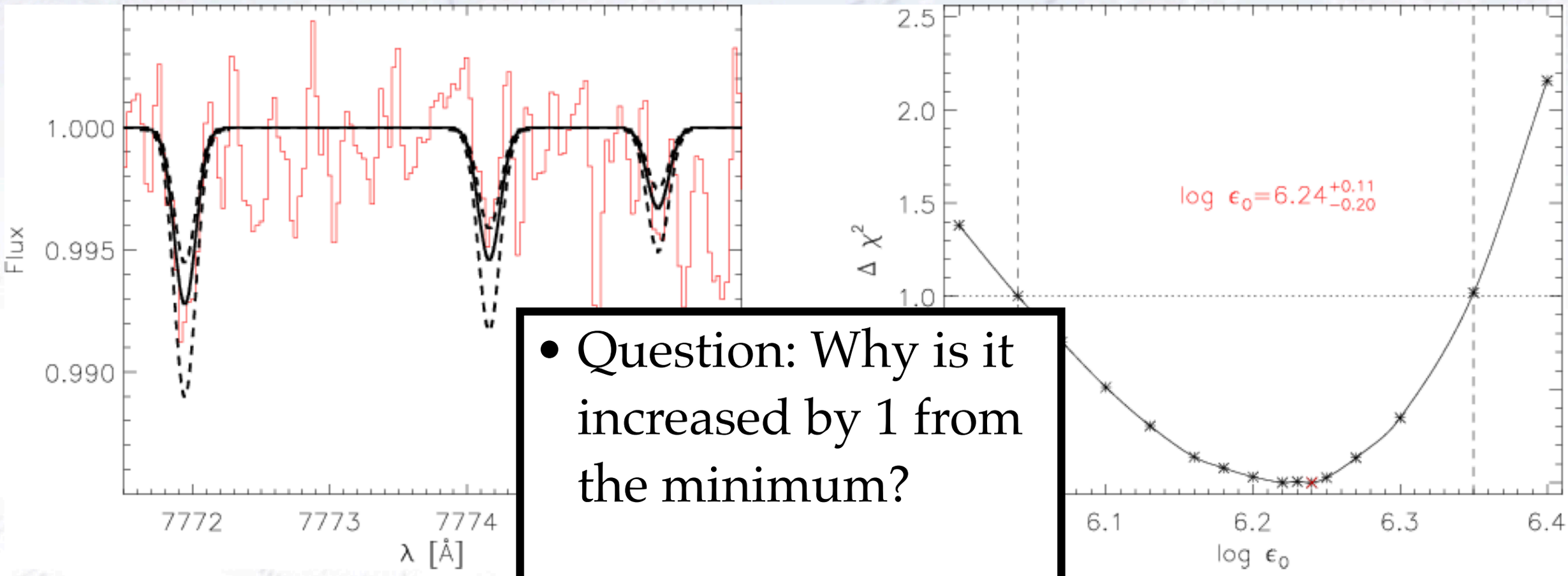

Mathias Spliid Heltberg (NBI)



"Statistics is merely a quantisation of common sense"

Example of Chi-Square

Uncertainties need not always be symmetric (though that is usually better!)

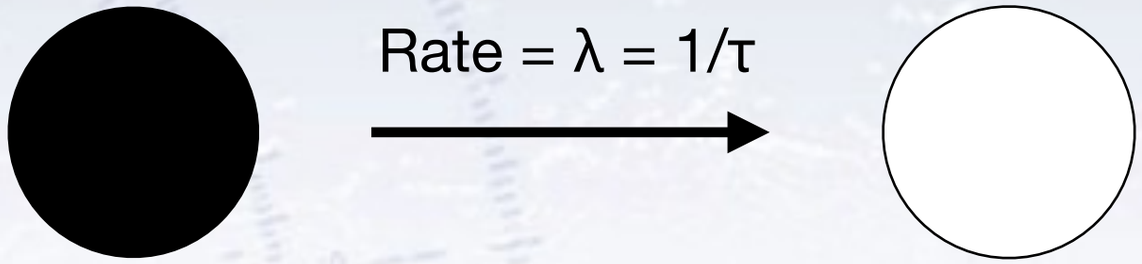


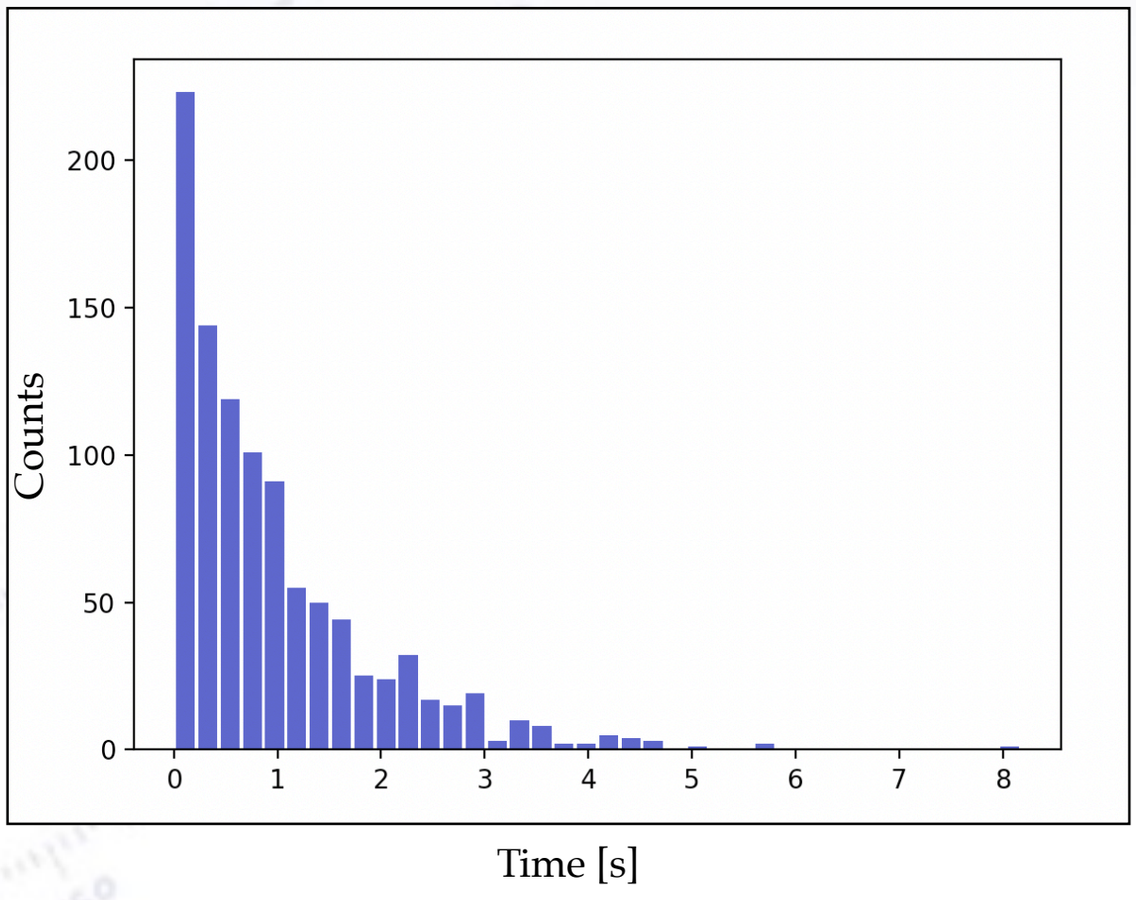
- Question: Why is it increased by 1 from the minimum?

The uncertainty on a parameter is found where the Chi2 has increased by 1 from the minimum.

The data

- Assume we are measuring the times for decay of a particle
- We measure 1000 decays, and plot them as follows
- We want to estimate the best parameters and its uncertainty

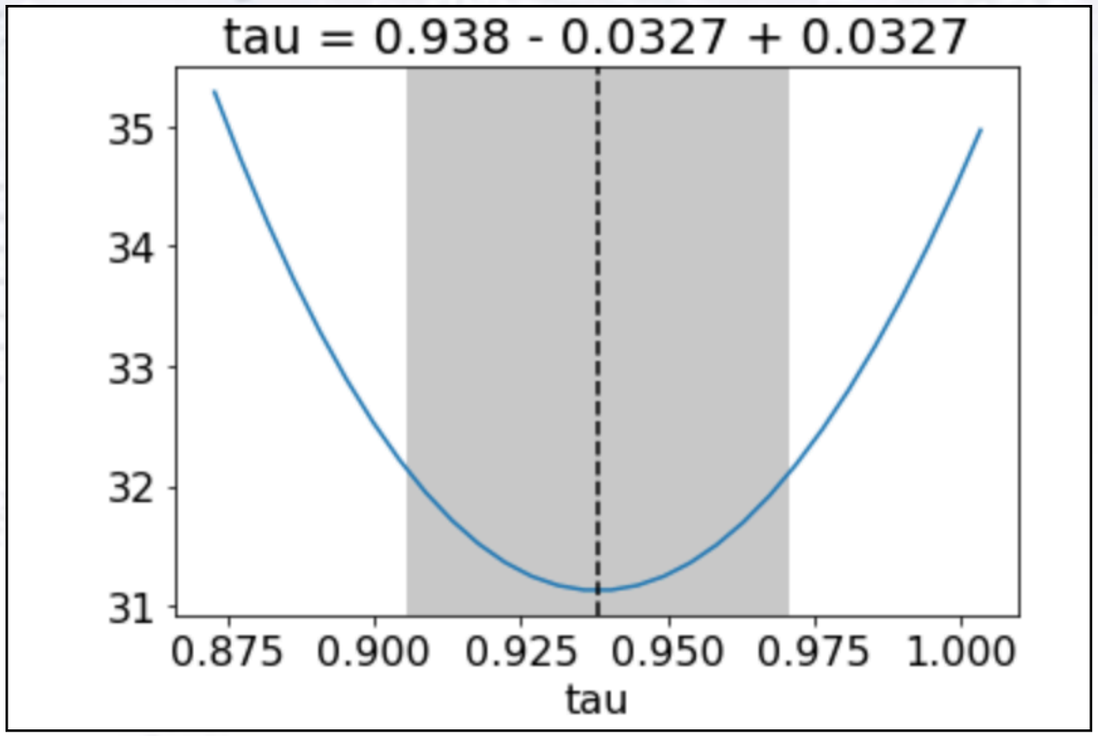




Uncertainty on parameters

- We have learned: errors are found at the y-value for min(χ^2)+1
- This is if we do it “by hand” or when it comes from Minuit.
- Why is this?

Data fitted using Minuit



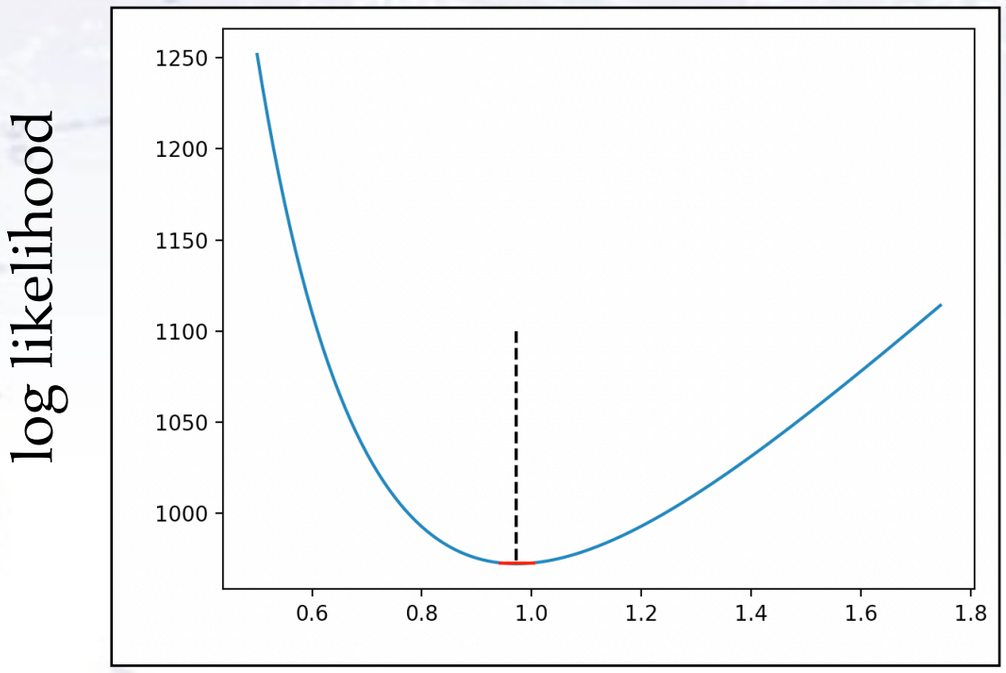
Scanning through the likelihood space

To outline this derivation, we have go through the likelihood function. Remember this takes the shape:

$$L(a) = P(X_1|a) \cdot ... \cdot P(X_n|a) = \prod_{i=1}^{n} P(X_i|a)$$

And this we typically calculated in log space, giving the log likelihood.

$$l(a) = -ln(L(a)) = -ln(P(X_1|a)) - ... - ln(P(X_n|a)) = -\sum_{i=1}^{n} ln(P(X_i|a))$$



Scanning through the likelihood space

If we zoom in on the most likely value, we see that this is to a very good approximation a parabola.

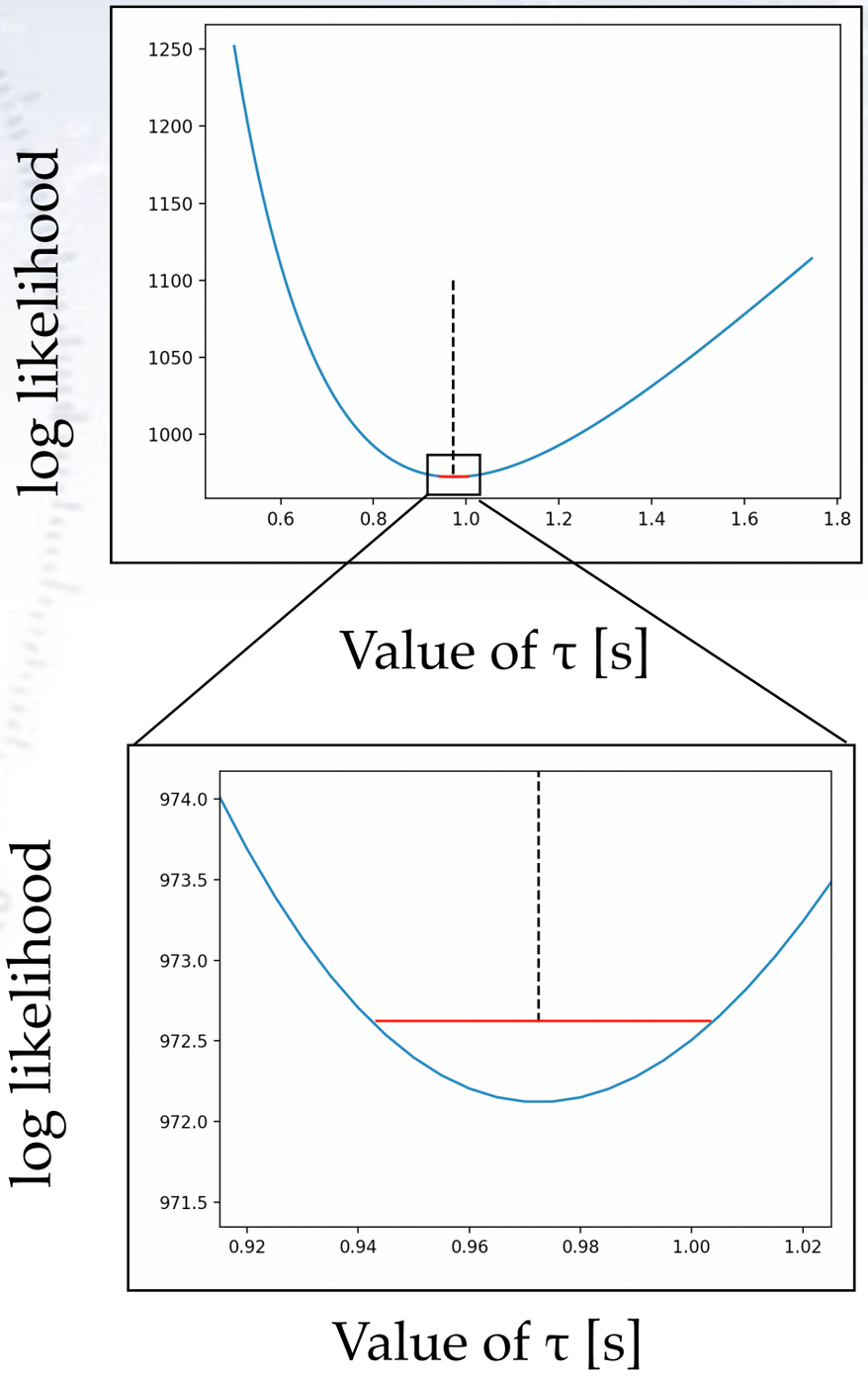
If we expand our expression for the log likelihood:

$$l(a) = l(\hat{a}) + (a - \hat{a})\frac{dl(a)}{da}|_{a=\hat{a}} + \frac{1}{2}(a - \hat{a})^2\frac{d^2l(a)}{da^2}|_{a=\hat{a}} + \ldots$$

To 2nd order, we obtain the following expression:

$$l(a) \approx l(\hat{a}) + \frac{1}{2}(a - \hat{a})^2\frac{d^2l(a)}{da^2}|_{a=\hat{a}}$$

Note that "**a**" is the parameter we are varying, whereas "**a_hat**" denotes the most likely value.



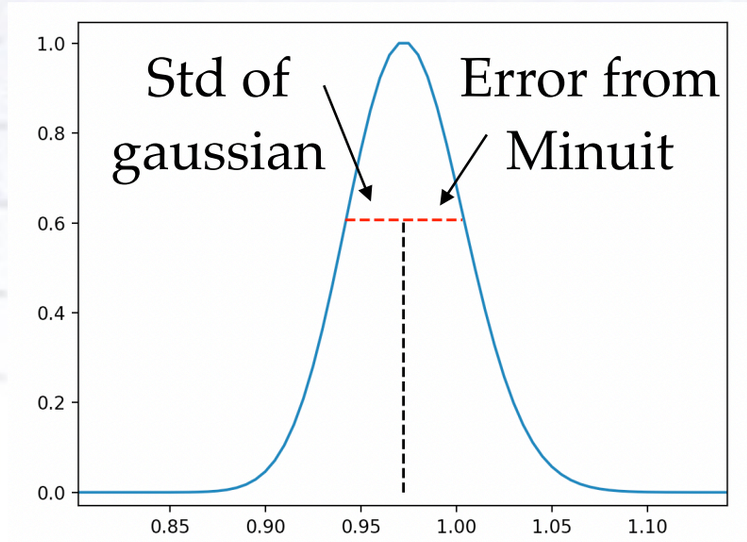
The likelihood is gaussian

So our log likelihood is a parabola, that means that our likelihood must be gaussian:

$$L(a) = e^{-l(a)} \approx e^{-l(\hat{a})} e^{-\frac{1}{2}(a-\hat{a})^2 \frac{d^2 l(a)}{da^2}|_{a=\hat{a}}} = c \cdot e^{-\frac{1}{2}(a-\hat{a})^2 \frac{d^2 l(a)}{da^2}|_{a=\hat{a}}}$$

We define the error as the std of the likelihood (gaussian) function - and we note that is the second derivative of the log likelihood!

$$\sigma^2 = \left(\frac{d^2 l(a)}{da^2}|_{a=\hat{a}} \right)^{-1}$$



Finding the uncertainty on the likelihood

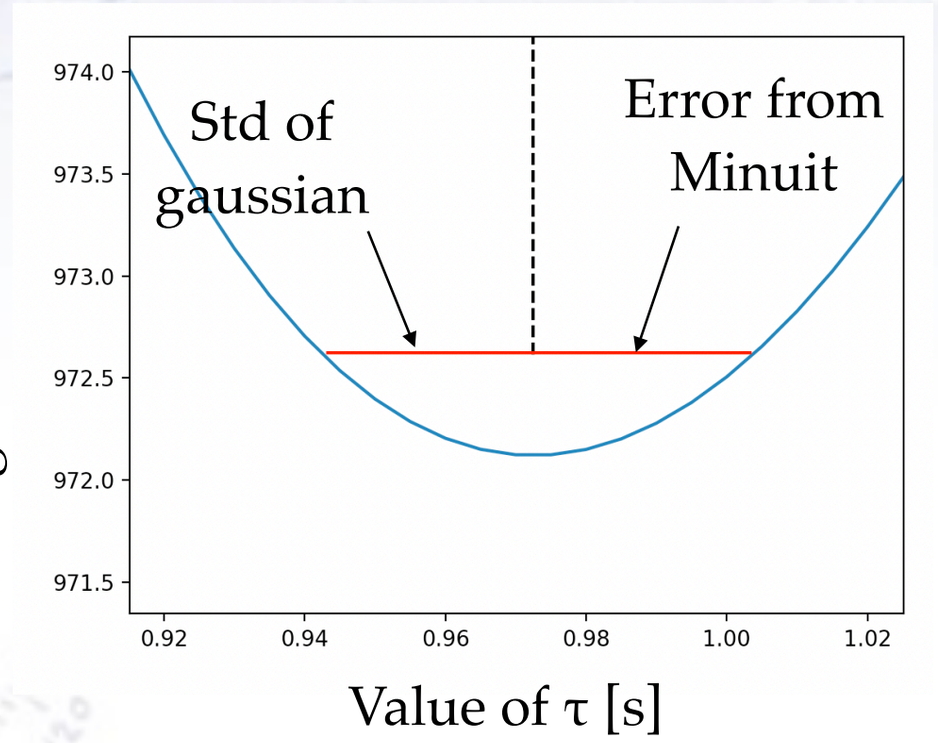
We happily jump back into the log likelihood. With our new definition that sigma is the second derivative of the log likelihood, our expansion takes the form:

$$l(a) \approx l(\hat{a}) + \frac{1}{2}(a-\hat{a})^2 \frac{d^2 l(a)}{da^2}|_{a=\hat{a}} \longrightarrow l(a) \approx l(\hat{a}) + \frac{1}{2}\frac{(a-\hat{a})^2}{\sigma^2}$$

But hey! If we insert the value a_hat+σ we see directly:

$$l(\hat{a} \pm \sigma) = l(\hat{a}) + \frac{1}{2}\frac{(\hat{a} \pm \sigma - \hat{a})^2}{\sigma^2} = l(\hat{a}) + \frac{1}{2}$$

This means that the uncertainty falls at the minimum of log likelihood + 1/2.



Relating this to χ^2

Lets assume we have enough measurements so all datapoints become gaussian. Then the likelihood can be written as:

$$L(\tau) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{1}{2}\frac{(X_i - f(\tau))^2}{\sigma_i^2}}$$

All probabilities follow a gaussian distribution

And taking the log likelihood.

$$-ln(L(\tau)) = \frac{1}{2}\sum_{i=1}^{n}\left(\frac{(X_i - f(\tau))^2}{\sigma_i^2} + ln(2\pi) + ln(\sigma_i^2)\right)$$

Where we quickly identify:

$$\sum_{i=1}^{n}\frac{(X_i - f(\tau))^2}{\sigma_i^2} = \chi^2$$

We can drop the constant terms, since we are gonna use the derivatives of likelihood with respect to the parameter, and we see that:

$$-ln(L(\tau)) = \frac{1}{2}\chi^2$$

And remember:

$$l(\hat{a} \pm \sigma) = l(\hat{a}) + \frac{1}{2}\frac{(\hat{a} \pm \sigma - \hat{a})^2}{\sigma^2} = l(\hat{a}) + \frac{1}{2}$$

So increasing the log likelihood by 1/2 give the error that has to mean that increasing χ^2+1 also give the uncertainty.

This shows that the uncertainty on a parameter is found at min(χ^2)+1!

Overview

In this introductory lecture we will go through a series of fundamental tools when working with time series. The topics will include:

- Stationarity, de-trending and de-noising
- Signal decomposition and Fourier coefficients
- Phase space analysis

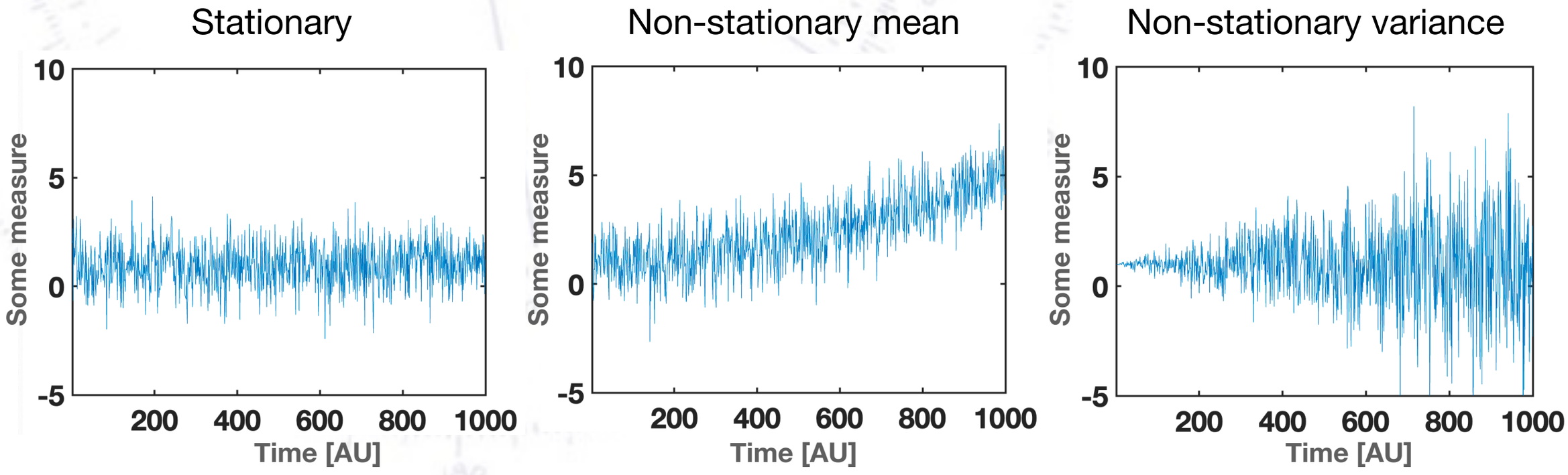
It is highly encouraged that you work on your own with the tools and try to apply them on some data you have at hand.

Stationarity

A time series is defined to be strongly stationary if all points on average are independent of the time of measurement.

Typically we will use the definition of weakly stationary processes. Here the first moment of a signal should be constant and the variance non-infinite.

Stationarity is of fundamental importance to many of the methods we are going to apply.

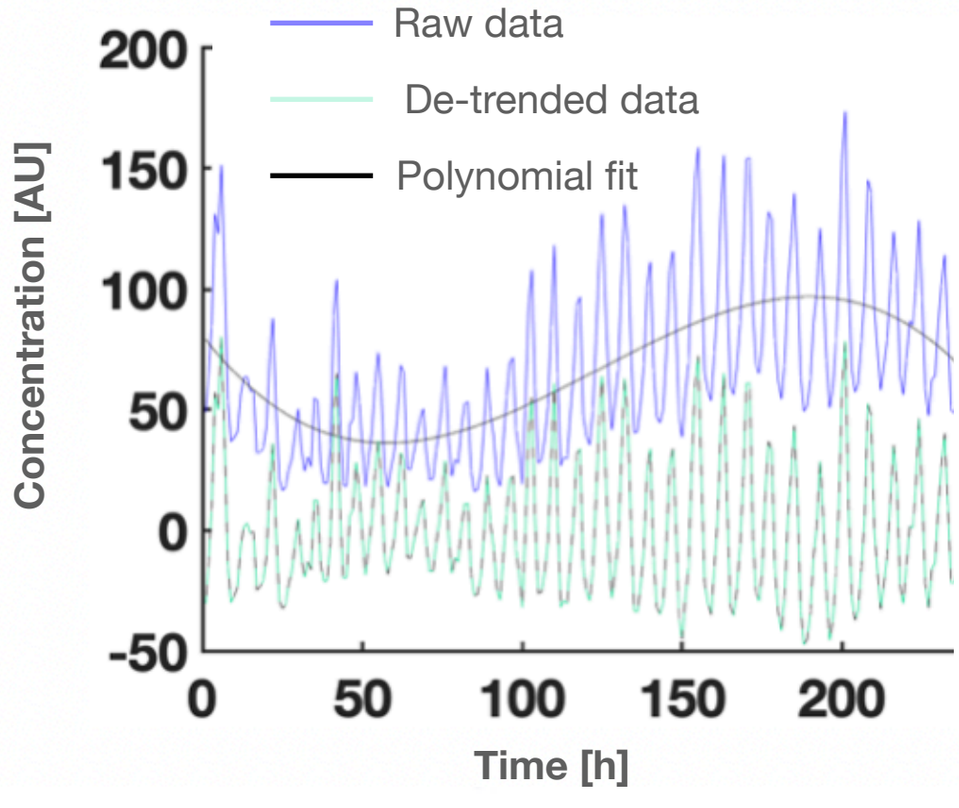


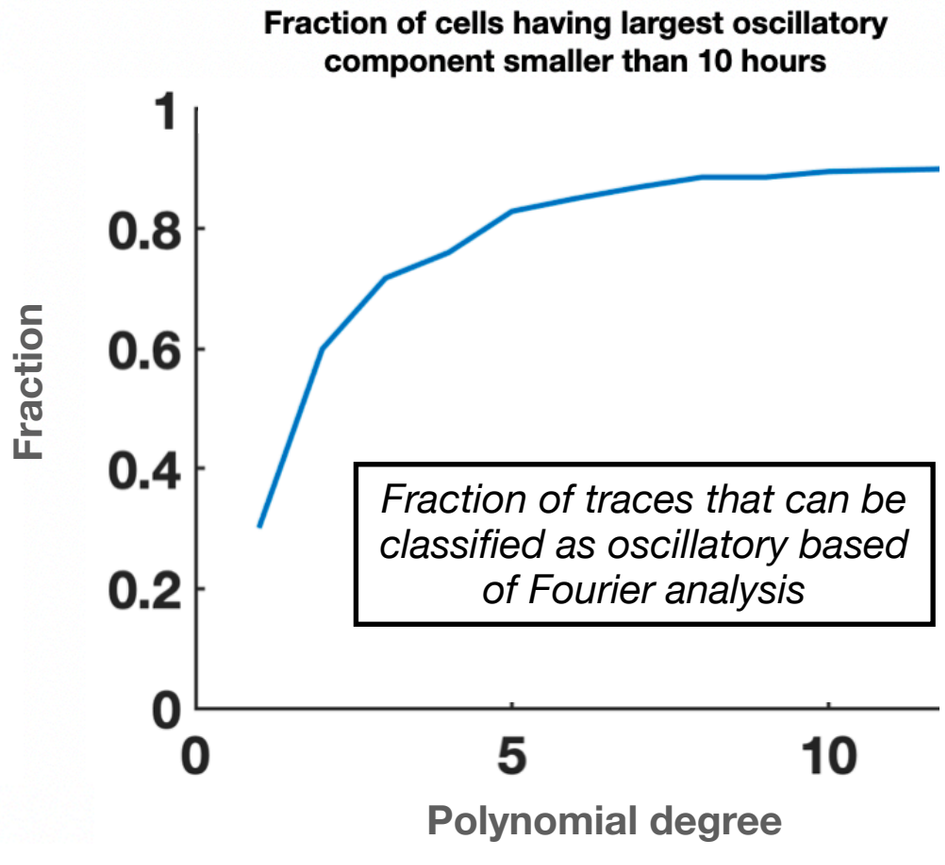
De-trending algorithms

Typically we would like a process to be stationary. This can be done by applying different kinds of filters.

Of particular importance is the Polynomial filter. This you know from the command “np.polyfit” in the numpy library.

This might not give you much information about what causes the non-stationarity, but it allows you to extract the main features.



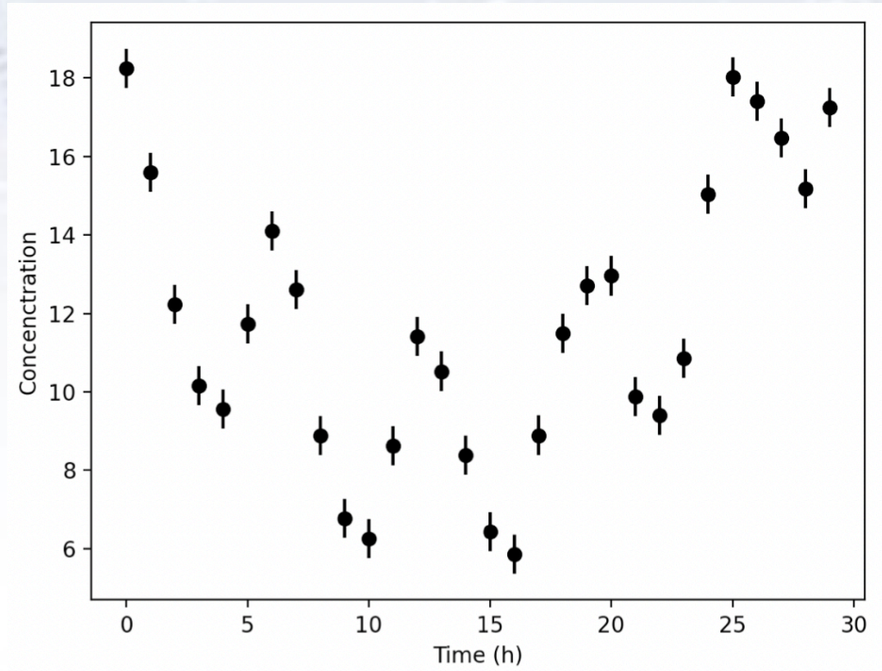


De-trending algorithms

Many of you might think: With large computers why can't I just plug everything into a fitting tool and obtain parameters?

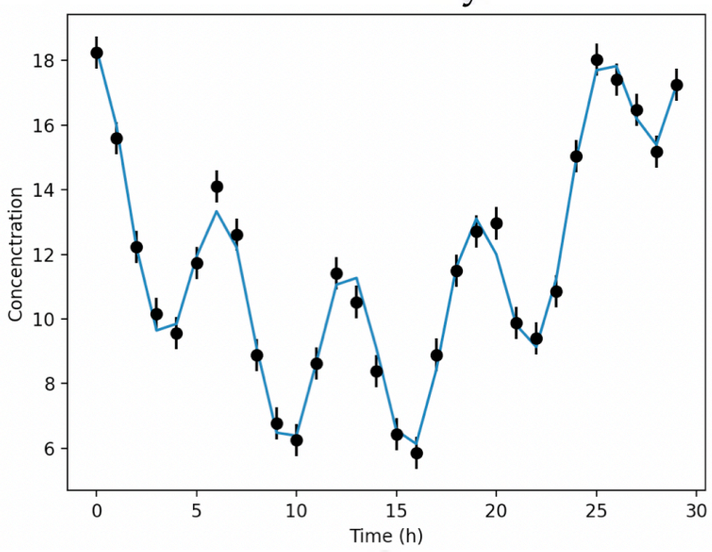
Assume we have a data series and we want to estimate the oscillations:

$$f(x) = c_1 + c_2 x + c_3 x^2 + c_4 cos(c_5 x)$$

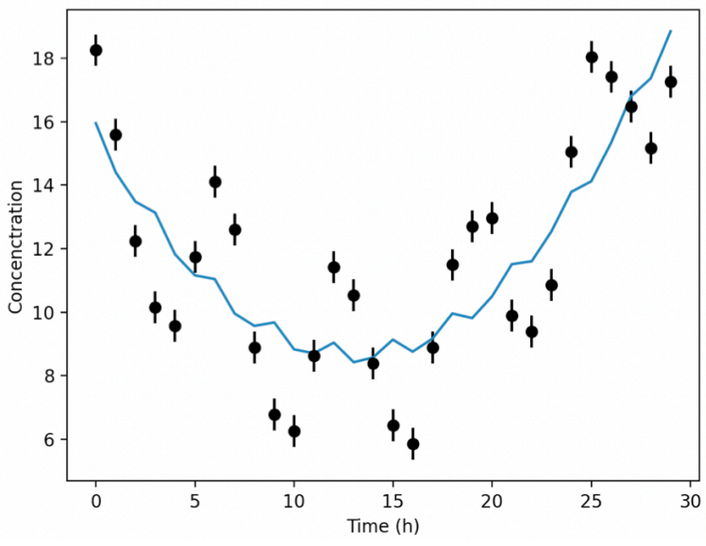


Lets now assume that we know almost all parameters, except one with a factor 10 uncertainty:

If the uncertainty is in c1



If the uncertainty is in c5



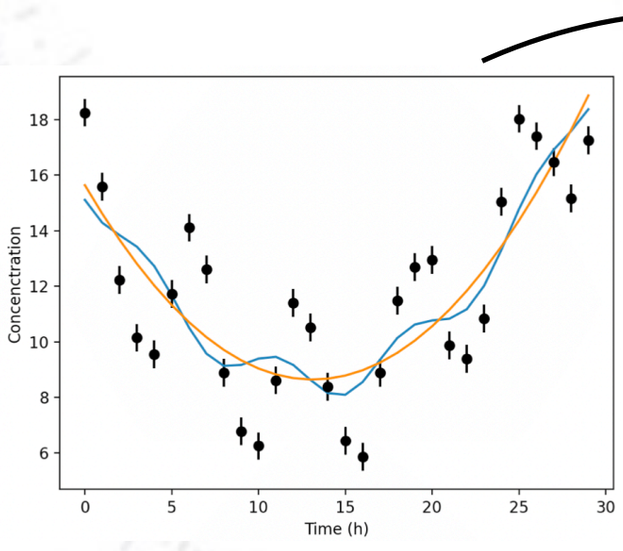
De-trending algorithms

Well, if we are interested in the oscillations, why not just apply the Fourier transform in the first place?

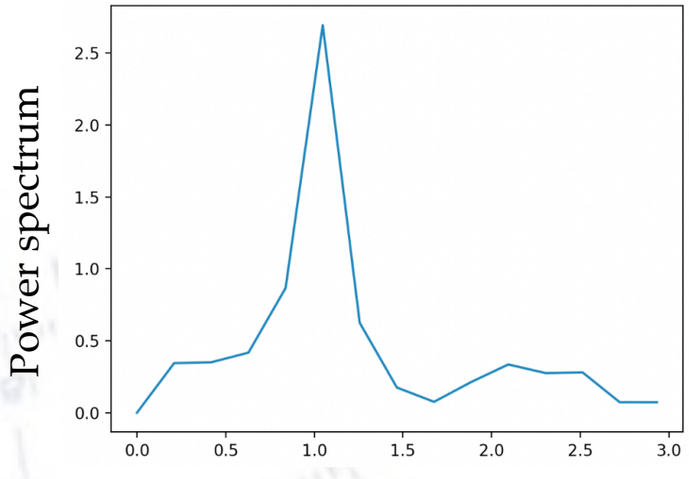
Since the data has a mean that is non-constant, and thereby not stationary, the underlying dynamics will dominate the Fourier spectrum.

Therefore we need to subtract this first - and this is done using a polynomial fit.

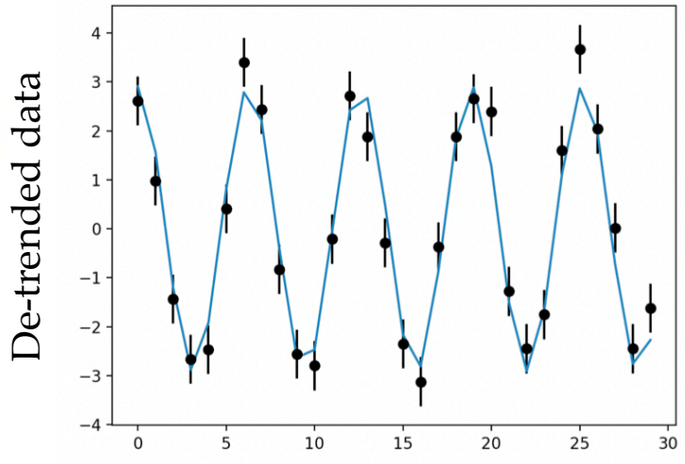
Finally we always test if the peak of the Fourier spectrum corresponds to the oscillations we are observing.



Time [h]



Frequency [1/h]



Time [h]

The optimal fit for any linear function

The polynomial fit is special since it is the analytically best solution - no numerical fitting - this is smart when efficiently detrending!

Any linear function has an analytical optimal fit. Lets take the function:

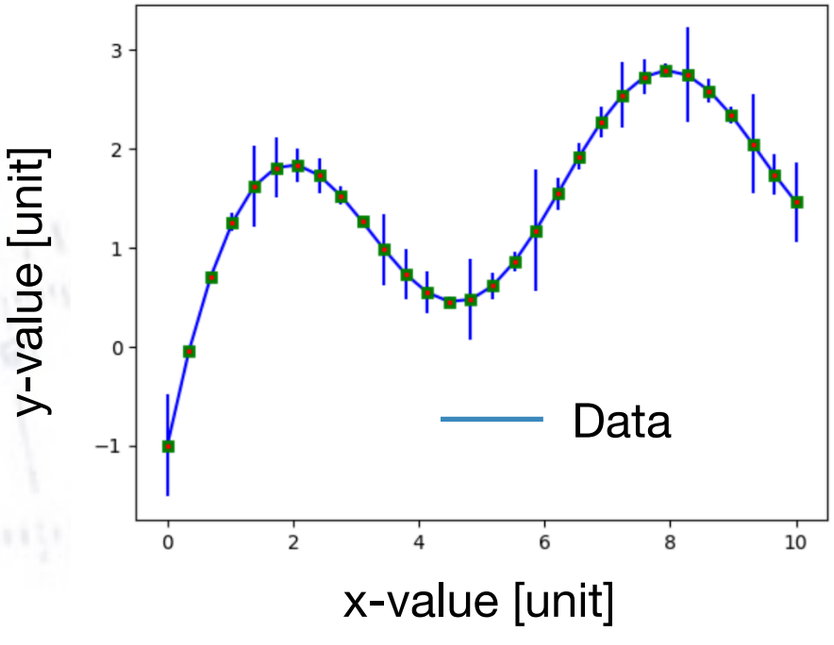
$$f(x|\vec{a}) = a_1 + a_2 x + a_3 \sin(x) + a_4 e^{-x}$$

With linear algebra we can write this in a compact form as:

$$\vec{f}(\vec{x}|\vec{a}) = C\vec{a}$$

With

$$C = \begin{pmatrix} 1 & x_1 & \sin(x_1) & e^{-x_1} \\ 1 & x_2 & \sin(x_2) & e^{-x_2} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & \sin(x_n) & e^{-x_n} \end{pmatrix} \qquad \vec{a} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix}$$



The optimal fit for any linear function

With notation we can also compactly write the χ^2 with this notation:

$$\chi^2 = (\vec{y} - C\vec{a})^T V^{-1} (\vec{y} - C\vec{a})$$
$$= (y_1 - f(x_1|\vec{a}))\frac{1}{\sigma_1^2}(y_1 - f(x_1|\vec{a})) + .. + (y_n - f(x_n|\vec{a}))\frac{1}{\sigma_n^2}(y_n - f(x_n|\vec{a}))$$

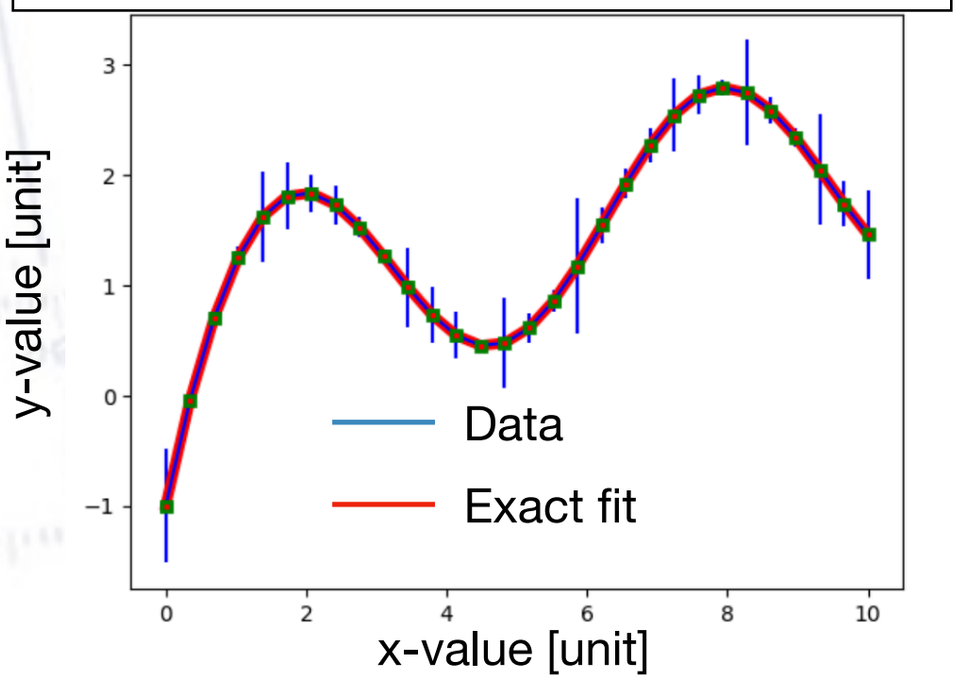
The optimal fit is for a=a_hat and this must mean that the derivative of χ^2 is zero:

$$\frac{d\chi^2}{d\vec{a}}|_{\vec{a}=\hat{\vec{a}}} = 0$$

$$\Rightarrow -C^T V^{-1}(\vec{y} - C\hat{\vec{a}}) + (\vec{y} - C\hat{\vec{a}})^T V^{-1}(-C) = 0$$

Isolating a_hat give directly the optimal fit given the data and the covariance matrix

$$\Rightarrow \quad \hat{\vec{a}} = (C^T V^{-1} C)^{-1} C^T V^{-1} \vec{y}$$



De-noising algorithms

Another aspect of the time series analysis is the subtraction of stochastic noise.

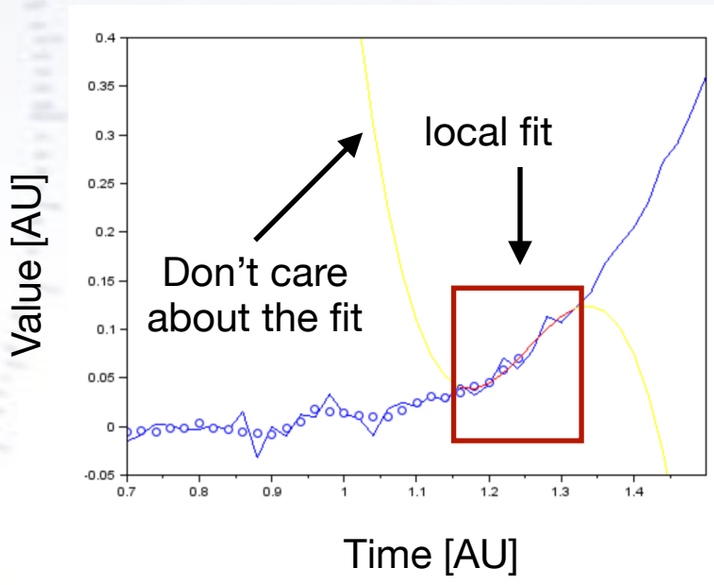
Many tools exist and for simplicity we will introduce one of them: the Salitzky-Golay filter.

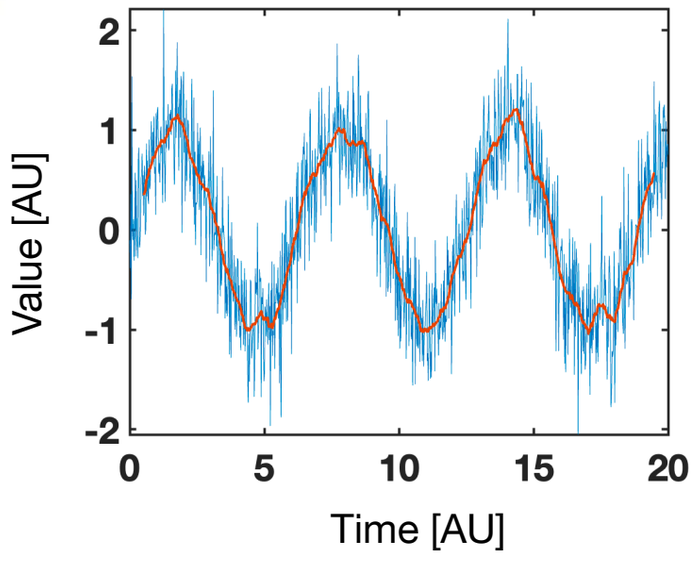
"it can be argued that the dawn of the computer-controlled analytical instrument can be traced to this article"

Here we simply take a given point, and then perform a polynomial filtering in +/- *m* data points.

This is done directly through our matrix notation

And as the implemented poly fit, where we typically choose a polynomium of order 3.





Fourier analysis

The majority of people have applied the Fourier analysis, but what is it actually?

To give direct insight, we assume to have a time series that will be periodic, we call this f(x). We try to fit it with a periodic function S(x).

$$S_n(x) = A_0 + A_1 cos(x) + ... + A_n cos(nx) + B_1 sin(x) + ... + B_n sin(nx)$$

To find the best fit, we define an error function (ε(x)) and calculate the least square fit (analytically)

$$f(x) = S_n(x) + \epsilon(x)$$

$$M = \frac{1}{2\pi} \int_{-\pi}^{\pi} \epsilon(x)^2 dx$$

Taking the derivative of M, with respect to each coefficient, we find:

$$\frac{\partial M}{\partial A_k} = 0 \Rightarrow A_k^* = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) cos(kx) dx$$

$$\frac{\partial M}{\partial B_k} = 0 \Rightarrow B_k^* = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) sin(kx) dx$$

Which are actually the Fourier coefficients. So it's simply an analytical way to derive the best oscillatory fit.

Fourier analysis

Just to finish the c

Plugging the valu

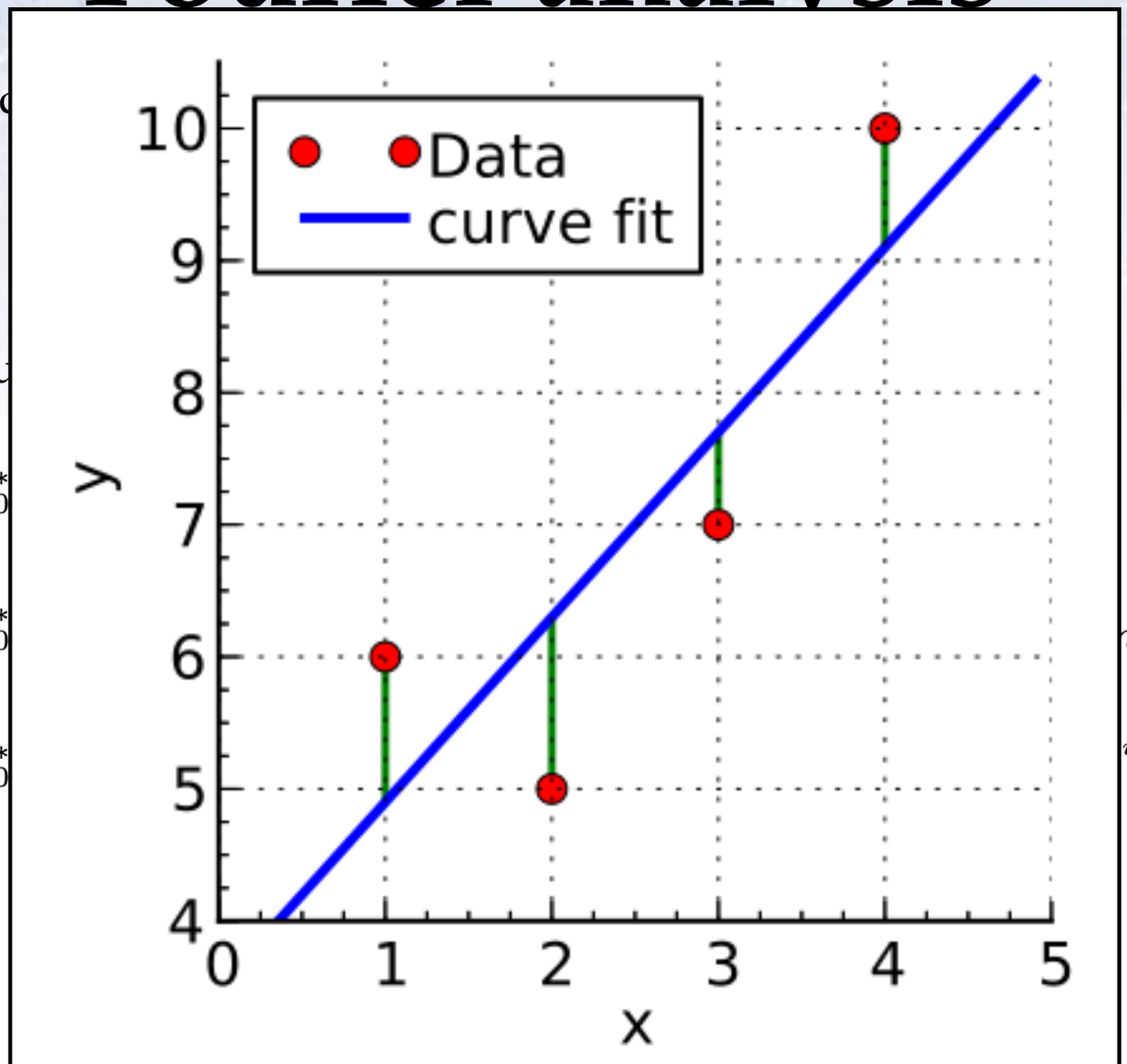
$f(x) = A_0^*$ $\cdot sin(kx)$

$= A_0^*$ $n(x)sin(y) = cos(x - y)$

$= A_0^*$ $^{ix} + e^{-ix})$

And we end with:

Which



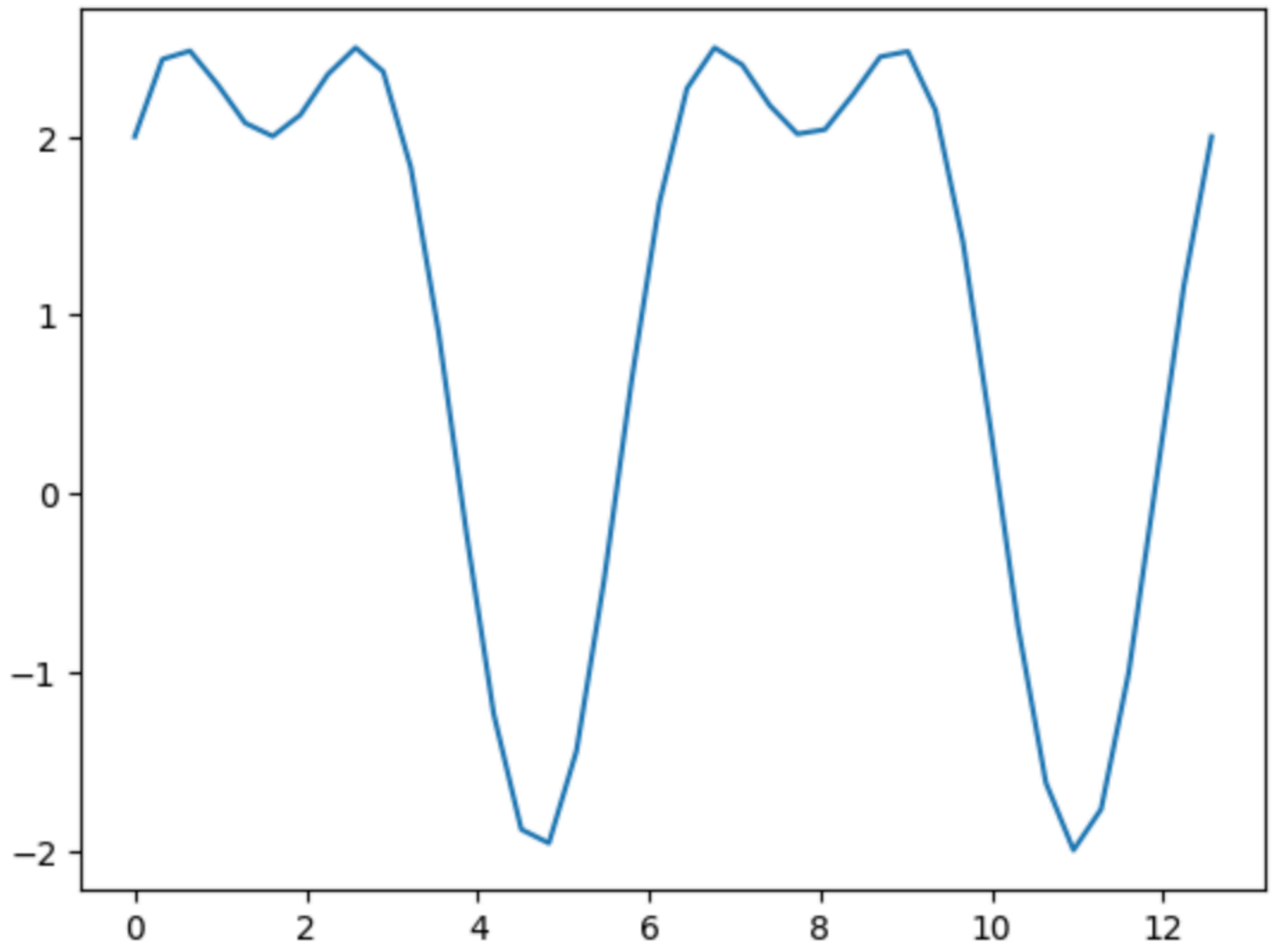
Simple an analytical Least-squares result!

Since $e^{ikx} = \frac{1}{2}(\cos(kx) + i\sin(kx))$ we must have: $C_k = \frac{1}{2}(A_k - iB_K)$

Output of FFT

We generate a signal

```
x = np.linspace(0,4*np.pi,40)
y = 1 + 1*np.cos(2*x)+2*np.sin(1*x)
plt.plot(x,y); plt.show()
```

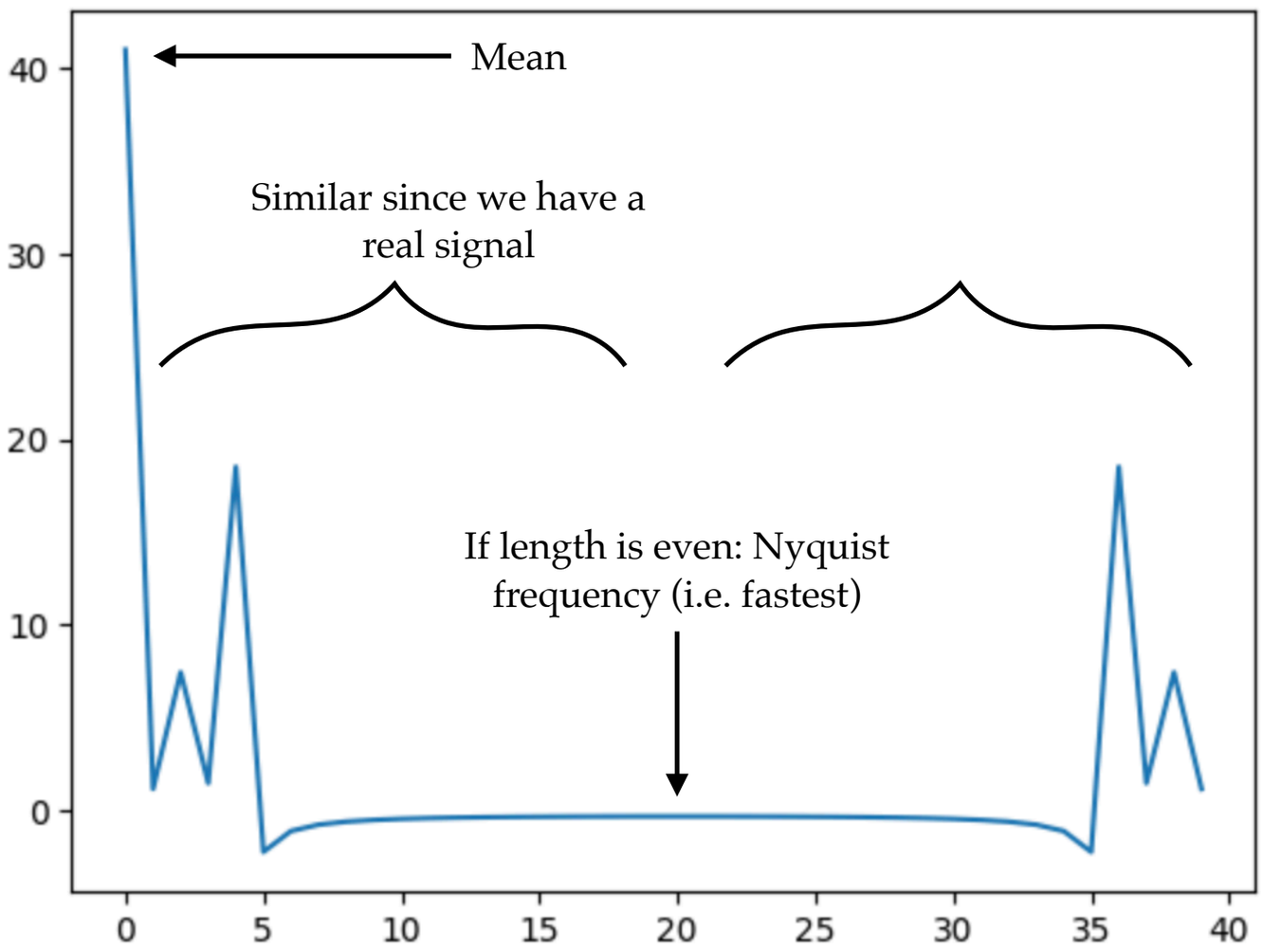


Output of FFT

Lets run the (Fast Fourier Transform:

```
X = np.fft.fft(y)
print('length',len(X))
plt.plot(X.real)
plt.show();
```

```
length 40
```



Interpret the output of the FFT:

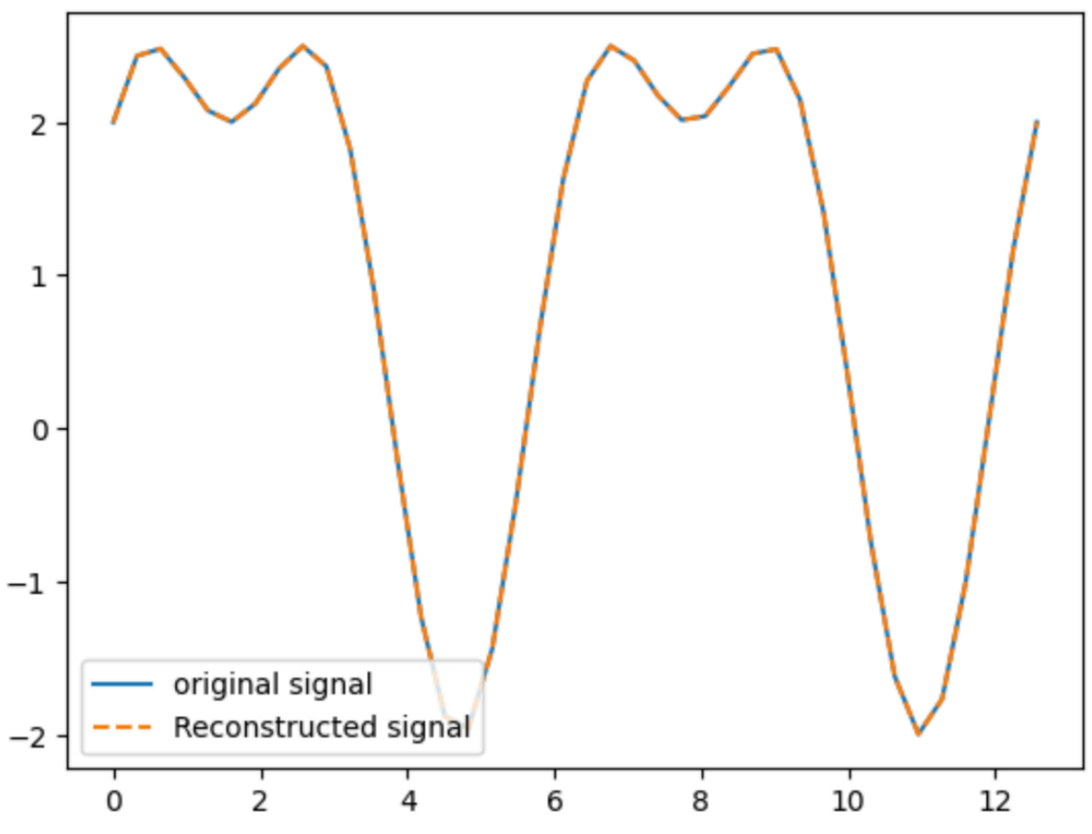
```
# Define the length of the signal
N = len(x); T = N

# The mean value is:
a_0 = X[0] / N
print('Mean value',a_0)

# Compute a_n and b_n
a_n = 2 * np.real(X[1:N//2]) / N
b_n = -2 * np.imag(X[1:N//2]) / N
```

```
Mean value (1.0250000000000001+0j)
```

Output of FFT

Reconstruct the signal

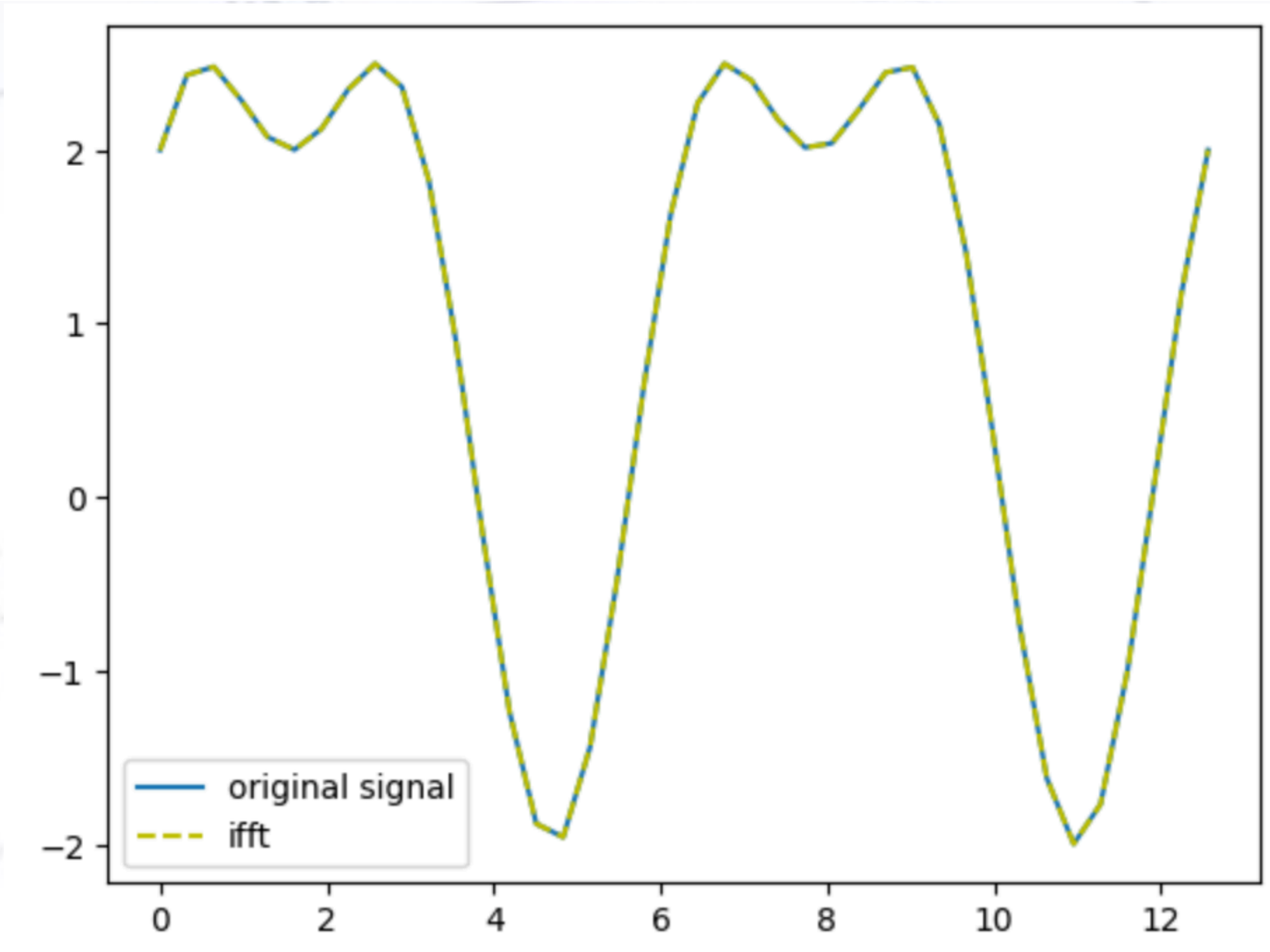
```
# Now we make the reconstricted signal
t = np.linspace(0, T, N, endpoint=False)
# Start with the mean
reconstructed_signal = a_0*np.ones(len(t))

# Add the a_n and b_n terms
for n in range(1, N//2):
    reconstructed_signal += a_n[n-1]*np.cos(2*np.pi*n*t/T) + b_n[n-1]*np.sin(2*np.pi*n*t/T)

# For even N, there is a Nyquist term (not critical to fit in many cases)
if N % 2 == 0:
    a_N2 = np.real(X[N//2]) / N
    reconstructed_signal += a_N2 * np.cos(np.pi * t)
```



Output of FFT

Of course this is similar to:

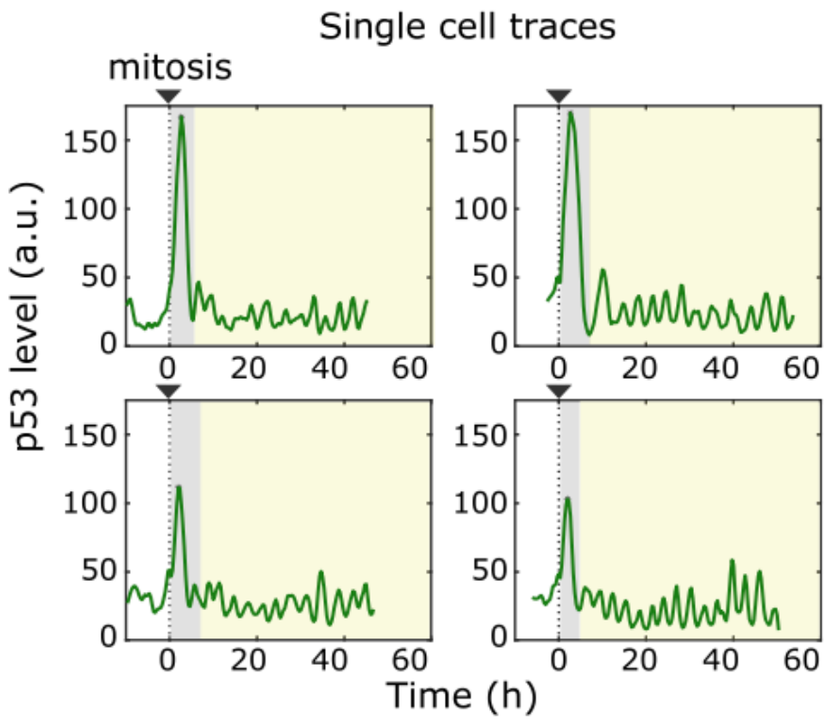
```
Y_ifft = np.fft.ifft(X)

plt.plot(x,y,label='original signal')
plt.plot(x,Y_ifft,'--y',label='ifft')
plt.legend(loc='lower left')
plt.show()
```

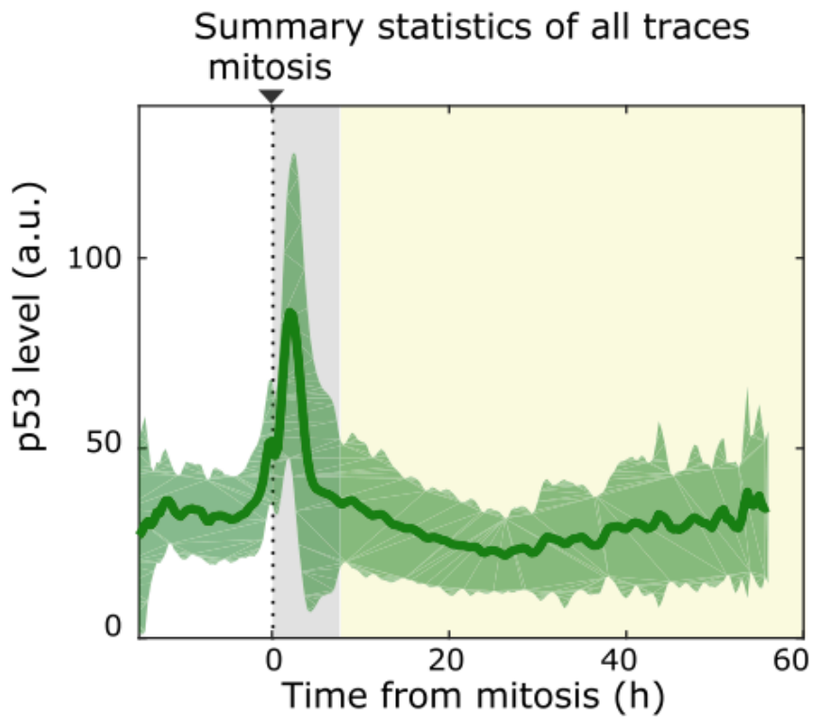


Fourier analysis - example

From this inspiration, we can directly apply the Fourier analysis - using FFT and iFFT.

Is we have a dataset of say 100 cells, we can perform an FFT analysis on each separate dataset - NOT their means.

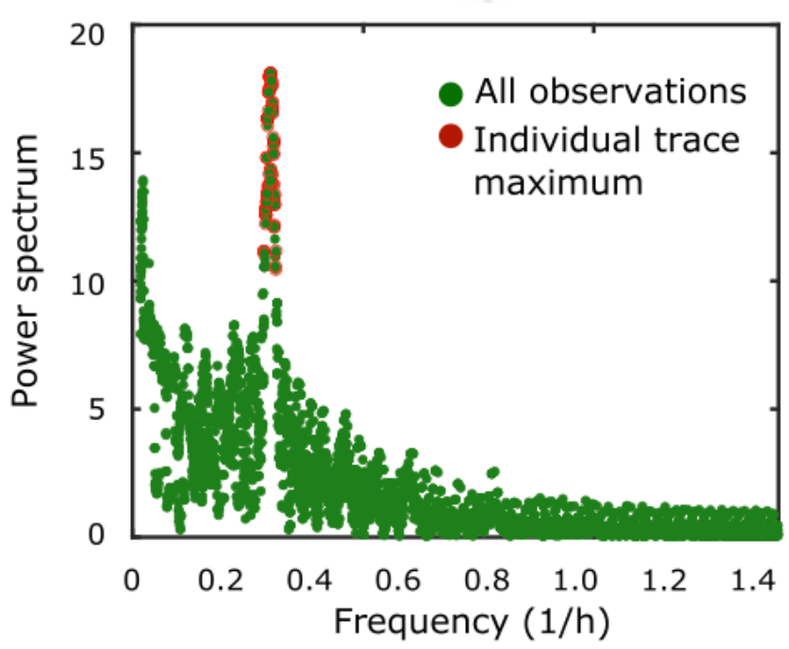


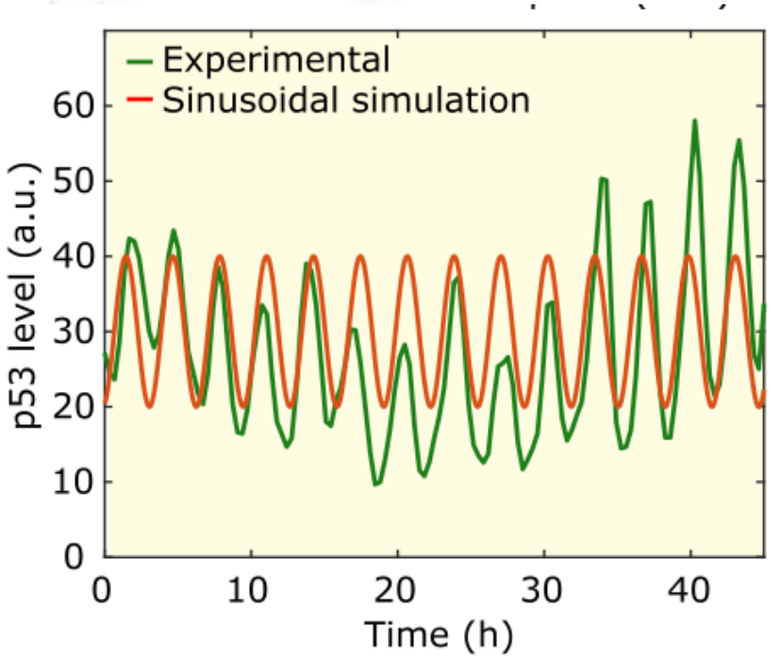


Fourier analysis - example

From this inspiration, we can directly apply the Fourier analysis - using FFT and iFFT.

Is we have a dataset of say 100 cells, we can perform an FFT analysis on each separate dataset - NOT their means.



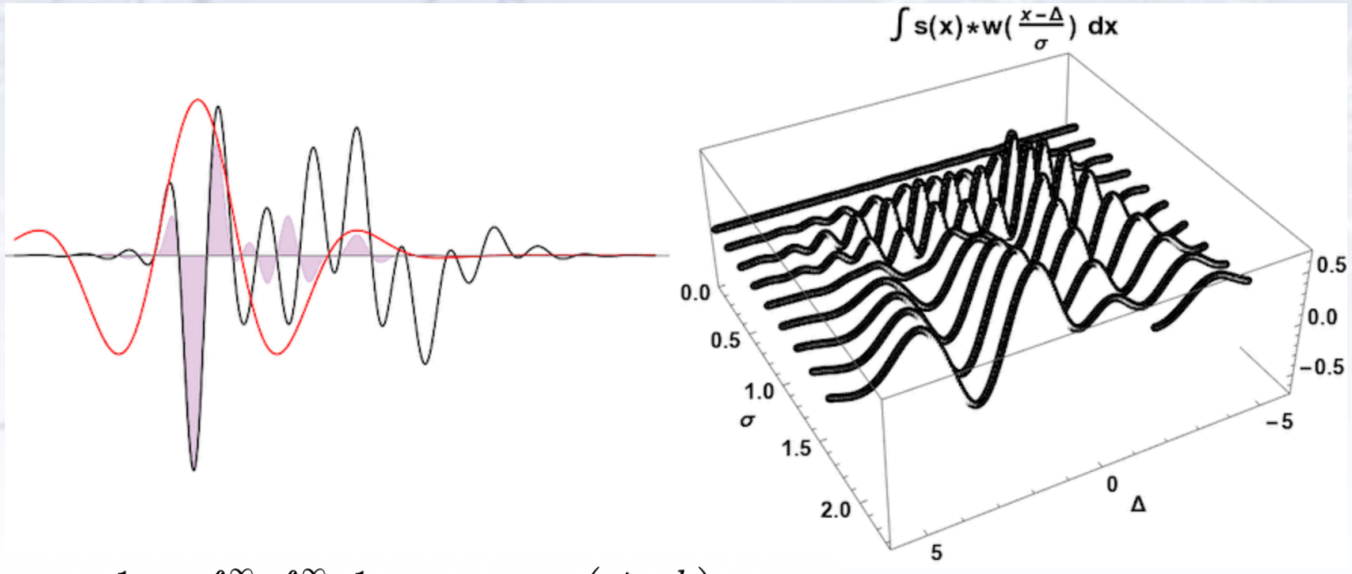


Continous Wavelet transform

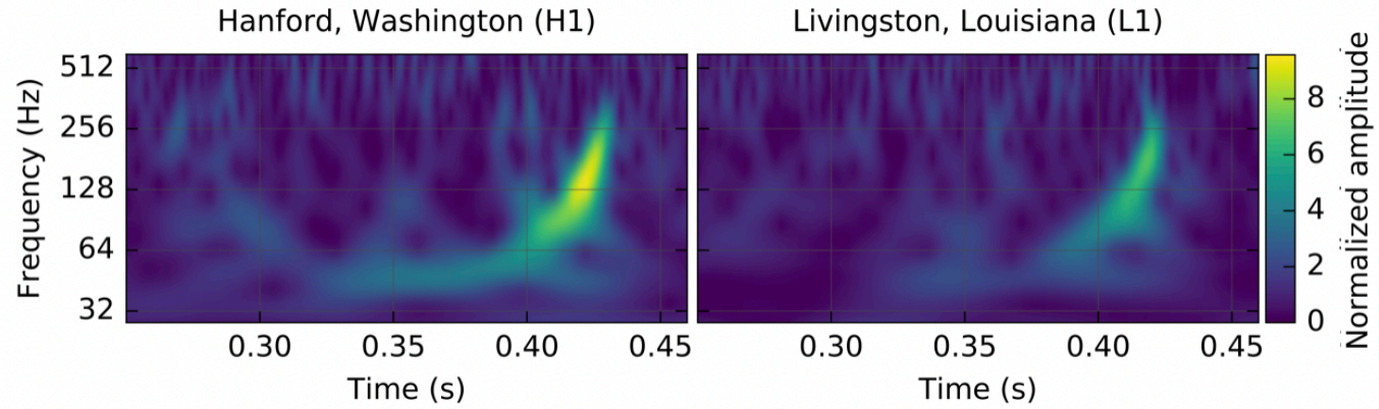
Sometimes we are not interested in a long-term oscillation - but rather small transient signals.

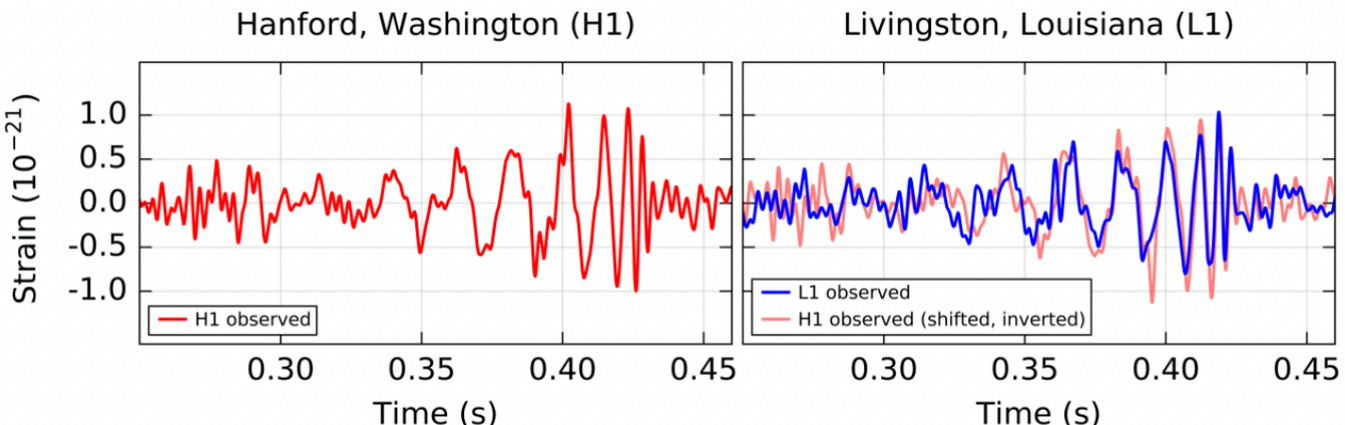
Here the continuous wavelet transform (CWT) is a natural development of the Fourier analysis, that basically estimates the coefficients of small wave packets.

This was of great importance, when the LIGO discovered gravitational waves!



$$x(t) = \frac{1}{2\pi\bar{\hat{\psi}}(1)} \int_0^\infty \int_{-\infty}^\infty \frac{1}{a^2} X_w(a,b) \exp\left(i\frac{t-b}{a}\right) db\, da$$





Some advanced tools

Autocorrelation

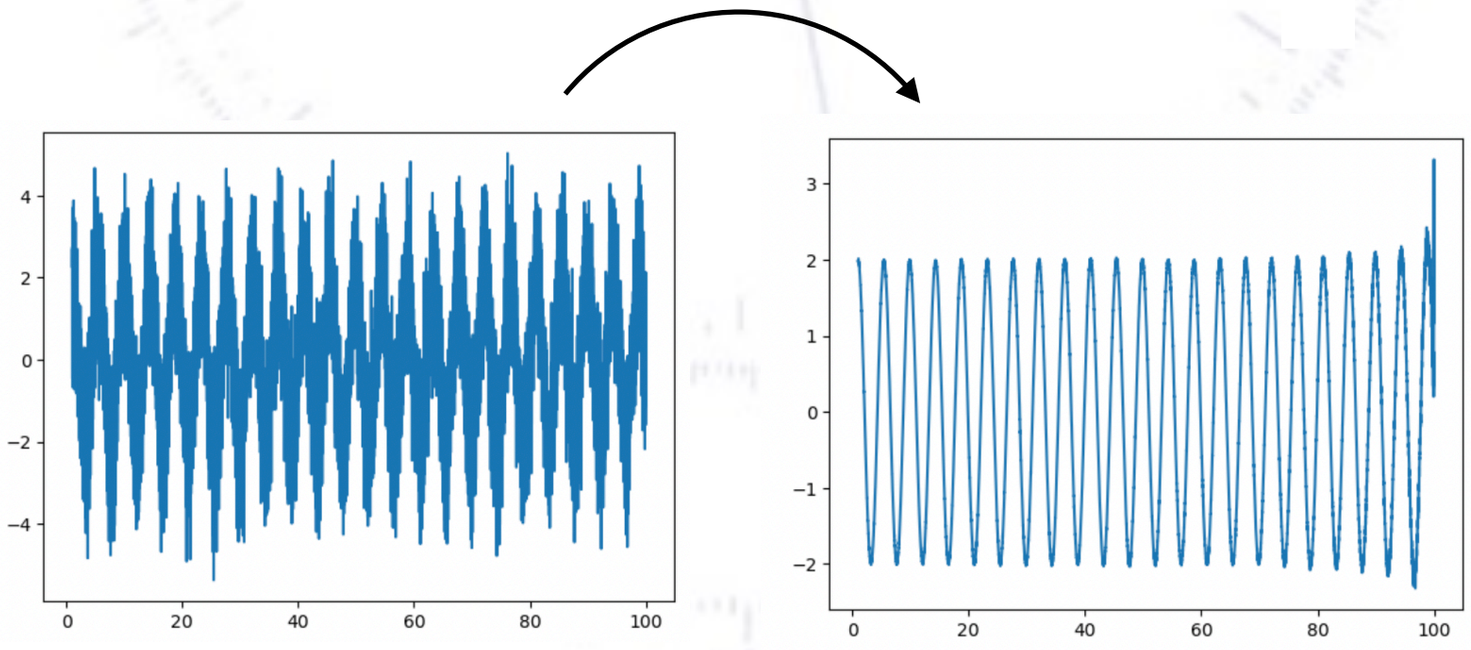
Lets say we want to compute the periodicity of some signal.

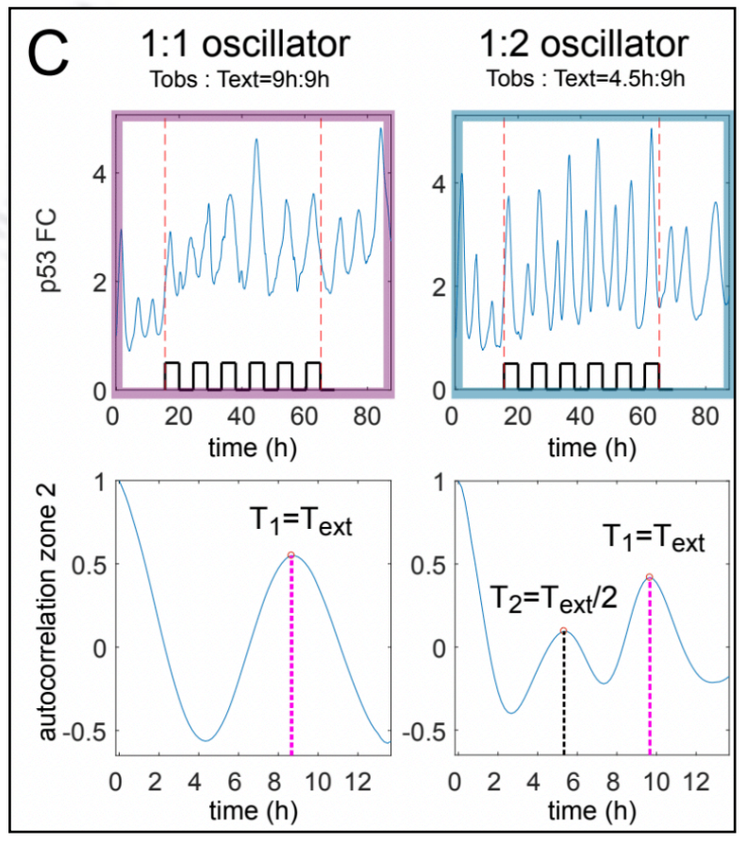
We can directly see at which points data is correlated, by calculating the correlation coefficient at future time points.

$$\hat{R}(k) = \frac{1}{(n-k)\sigma^2} \sum_{t=1}^{n-k} (X_t - \mu)(X_{t+k} - \mu)$$

By doing this and increasing the time addition we evaluate correlation in, we obtain an autocorrelation.

We can do this across different conditions and compare the resulting periodicity.





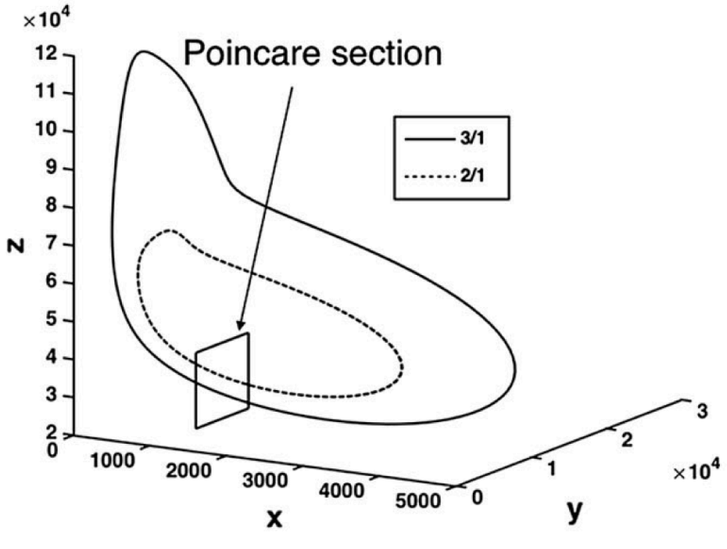
Poincare section

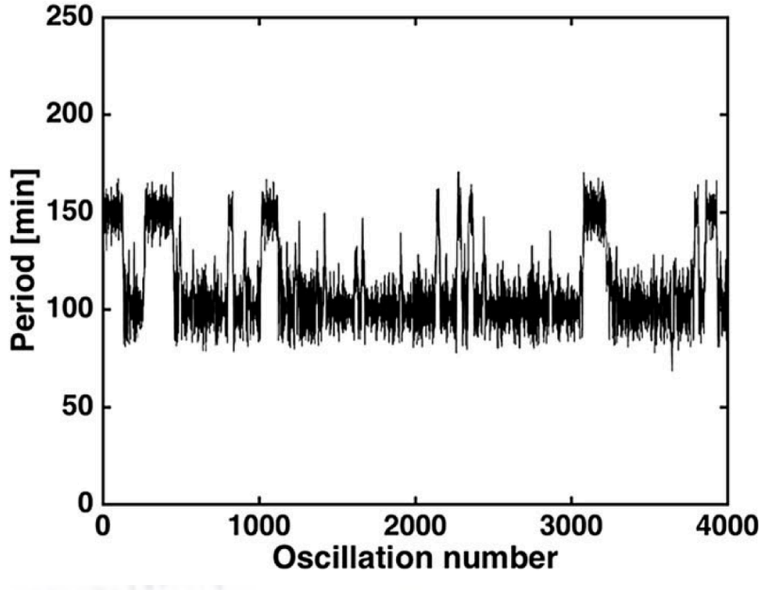
A Poincare section is has dimension n-1, where n is the dimensionality of the data.

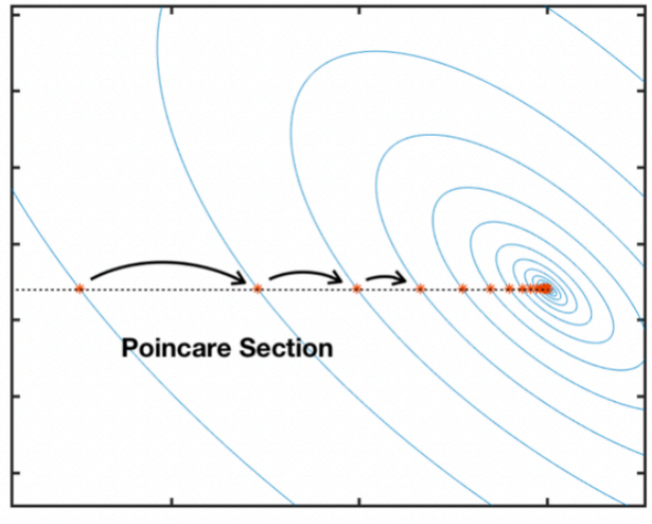
If for instance the data is in a 3 dimensional phase space, the Poincare section is a plane that is situated at some location (after your choice) where the trajectory passes through.

In this way, one can study recurrent dynamics, and investigate this in a discrete time series.

This can for instance be applied when studying the motion between two limit cycles, where the transitions might occur in the middle of a rotation.





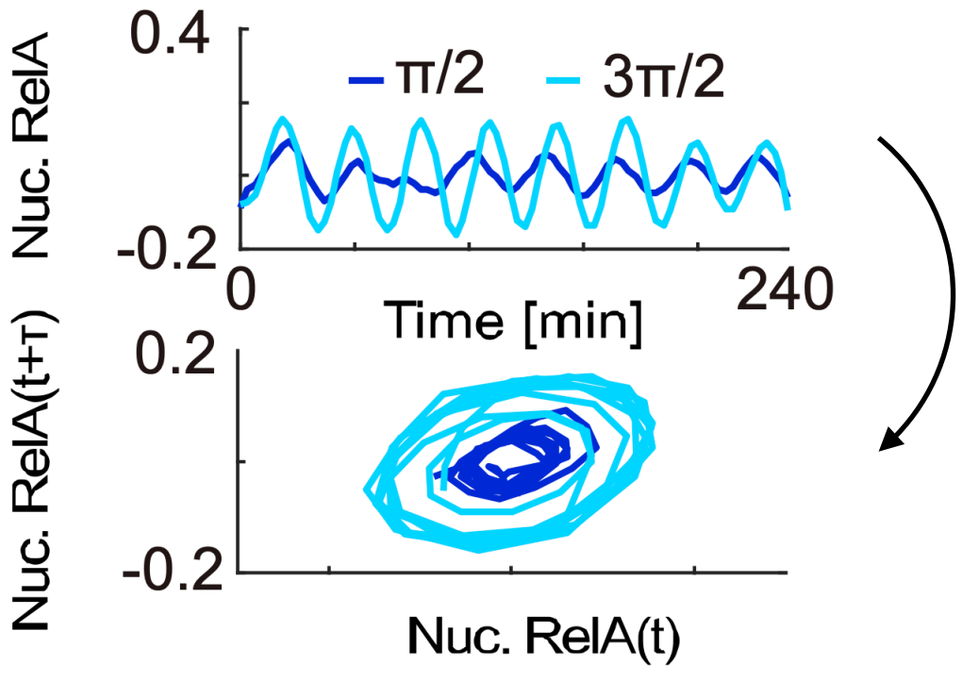


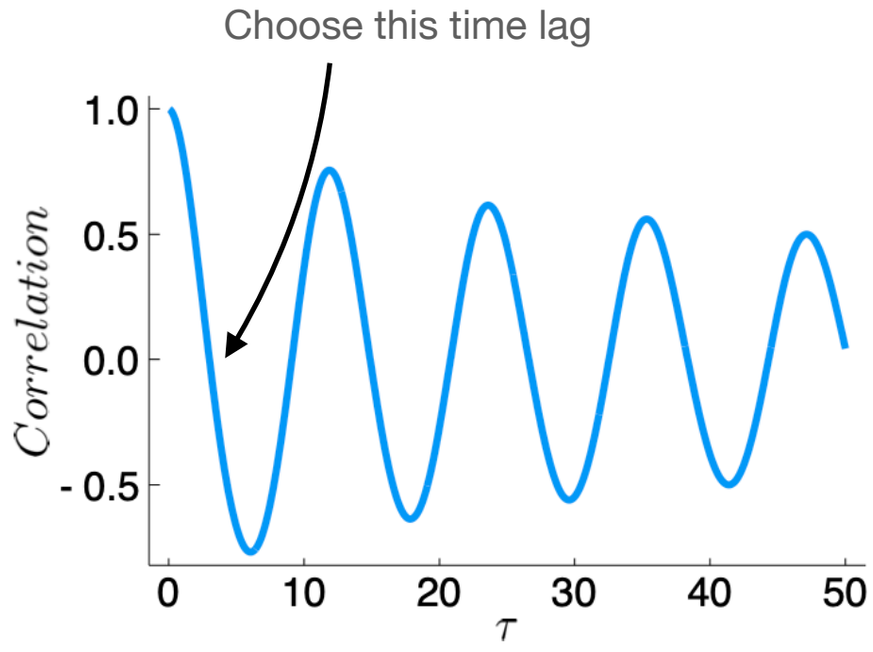
Time embedding

Suppose we have a time series, but we would like to obtain information about the geometry of the phase space.

Then we can apply a technique known as time embedding. Here we basically take a time series from x(1:end-L+1) and x(L:end) - and this can be extended to any dimension.

Building on a result known as “Taken’s theorem”, the embedded attractor will have the same properties as the underlying network in many dimensions.

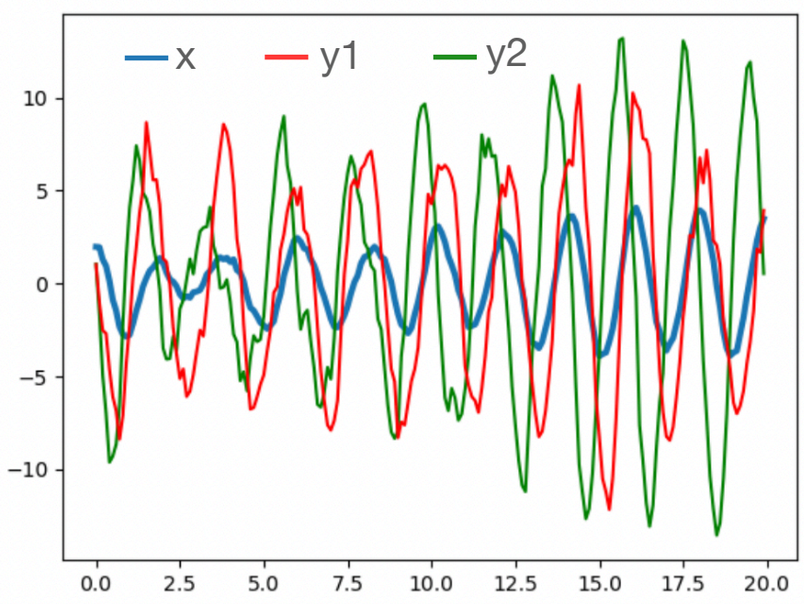


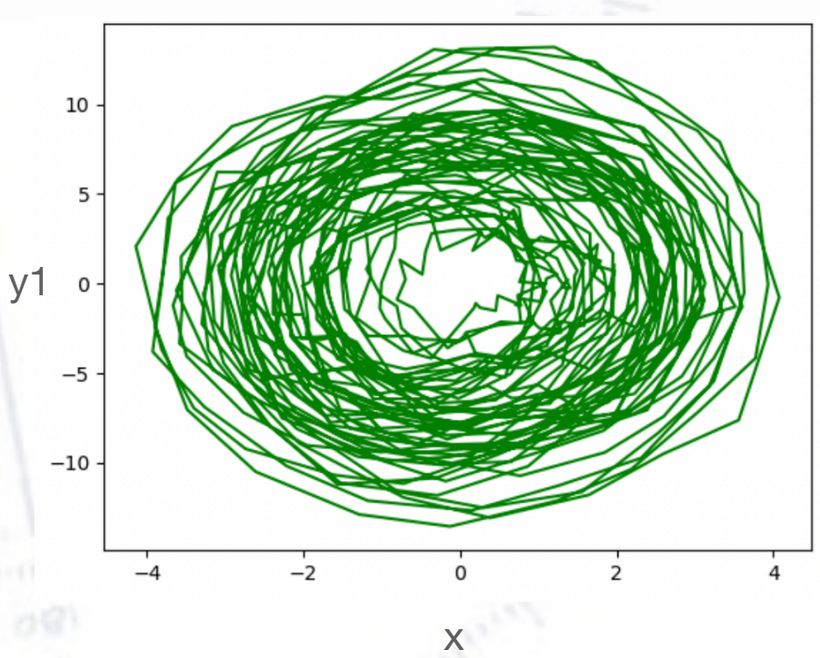


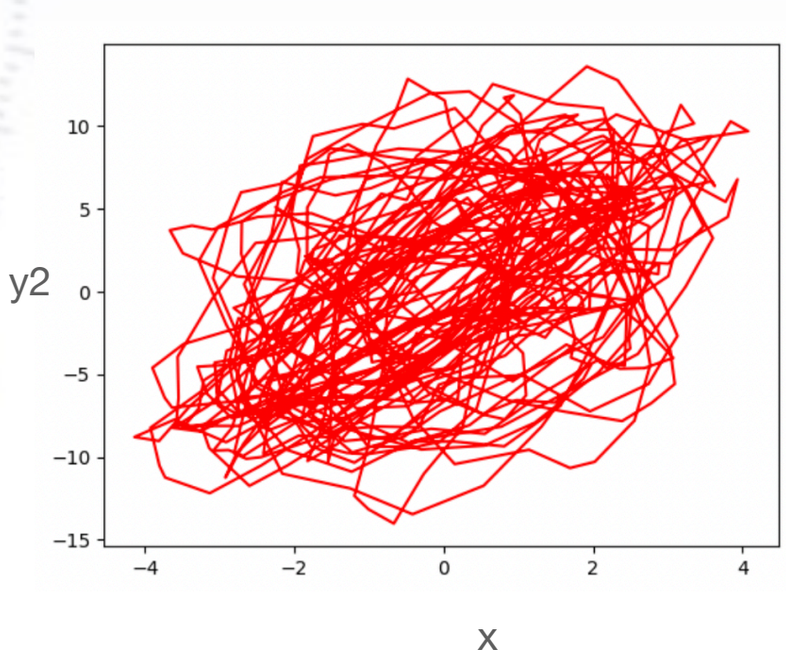
Phase space analysis

Sometimes a lot can be gained of insight by considering the actual phase space instead of the times series by itself

The phase space reveals the structure of the interplay of the studies variables. For instance a natural question would be to ask: Do these create a closed cycle?

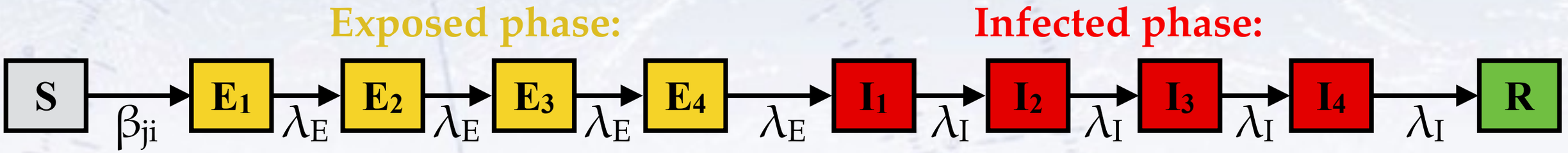






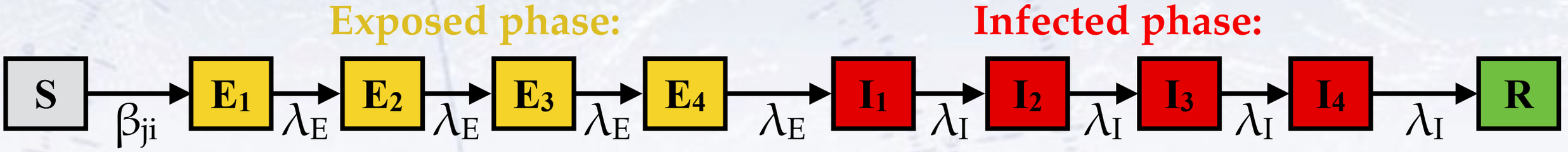
Fitting with a model

Occasionally, the model is not a function, but a more advanced model. One example could be the SEIR-model for epidemics, here with 4 E and I stages:



Fitting with a model

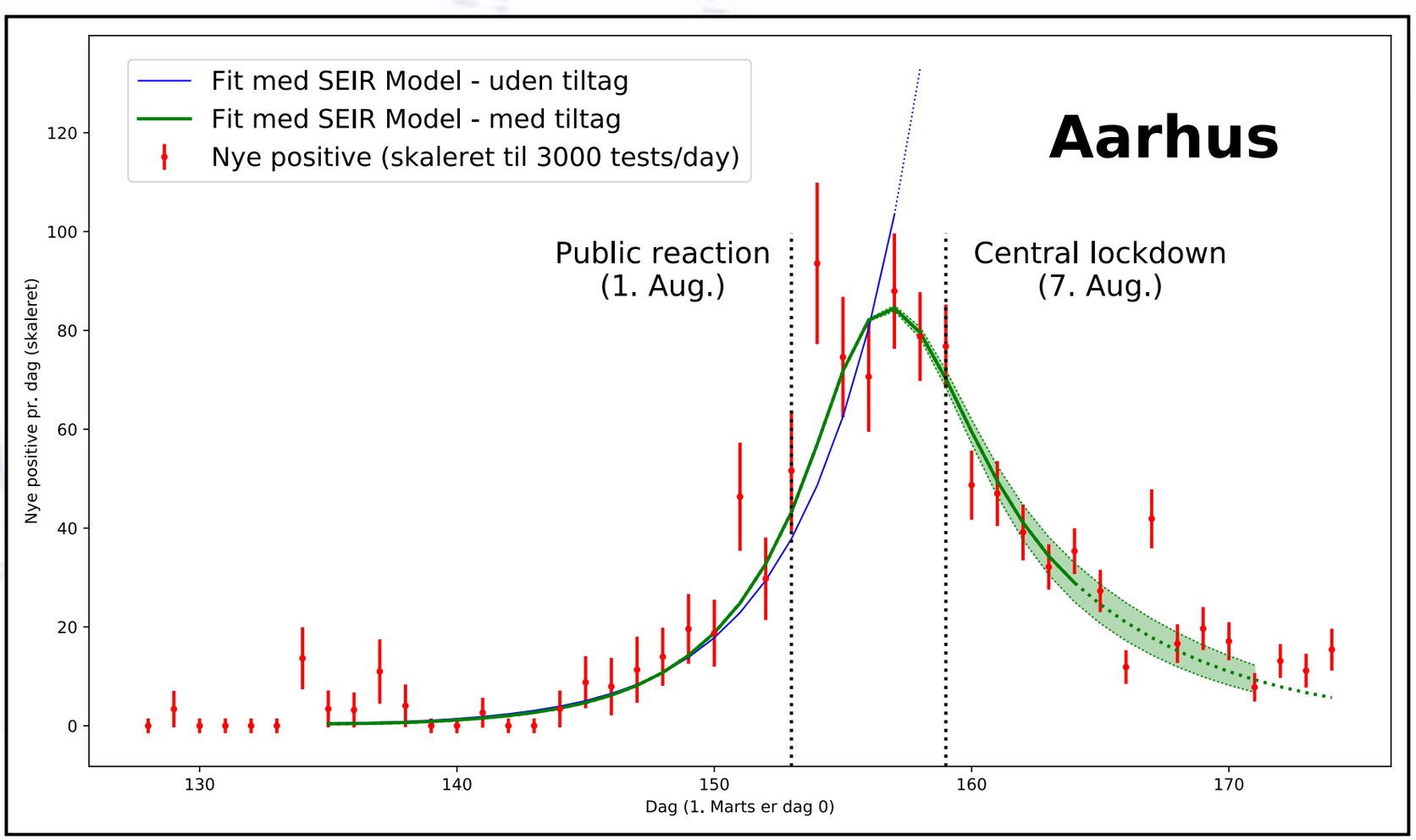
Occasionally, the model is not a function, but a more advanced model. One example could be the SEIR-model for epidemics, here with 4 E and I stages:



An example outbreak of Covid-19 in Denmark happened in Aarhus early August 2020:

The number of tests and positives are given, from which a "scaled" number of infected can be calculated and fitted.

The fit is exactly with the SEIR model above, fixing most parameters.



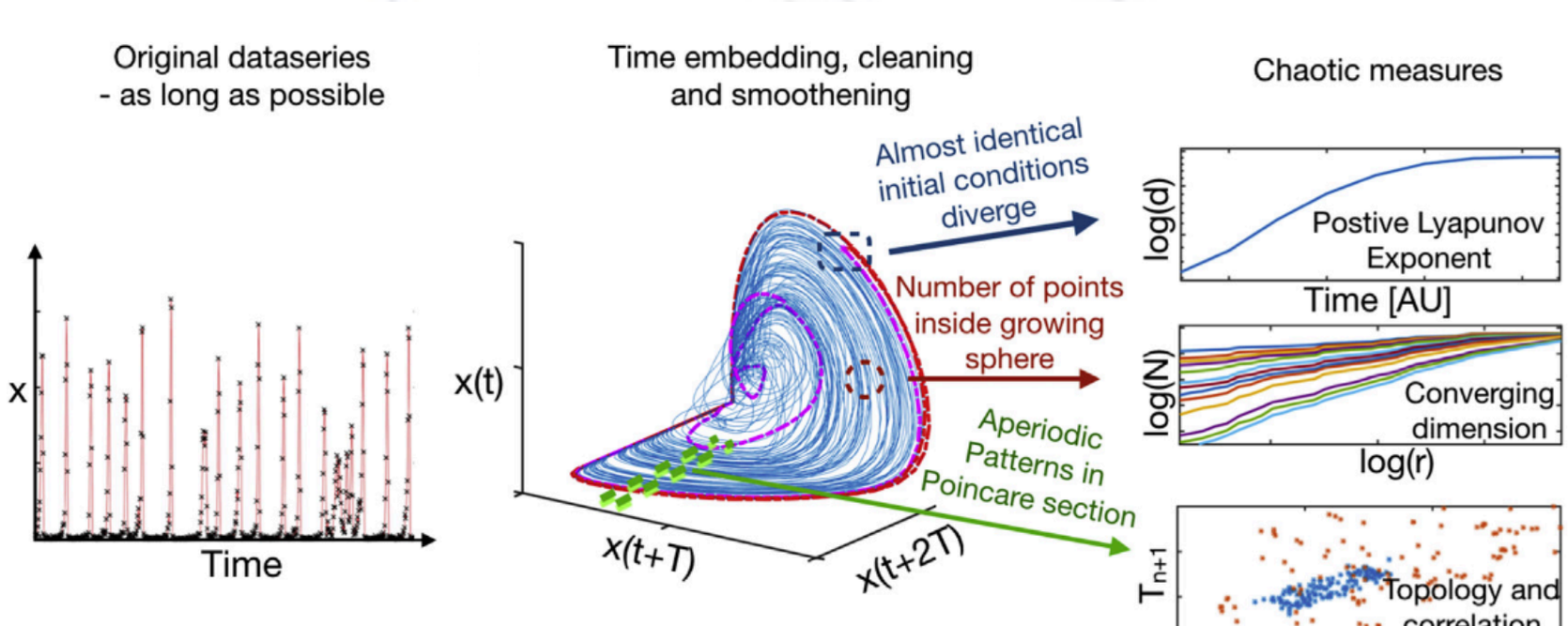
Challenge: Chaotic dynamics

When we look for dynamics, we typically investigate the possibility of oscillations.

However more complex dynamics exist that is related to the features of oscillations.

In particular we want to mention the possibility of chaotic dynamics. This is defined by the fact that two trajectories, starting infinitely close to each other diverge in time.

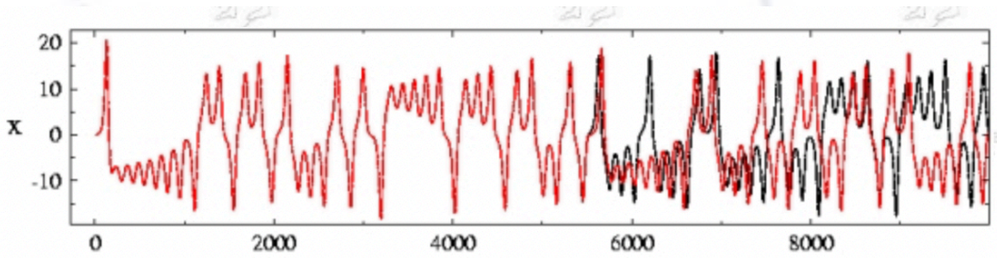
This is difficult to access, when we cannot perform two experiments starting from exactly same conditions - however methods exist to estimate this.

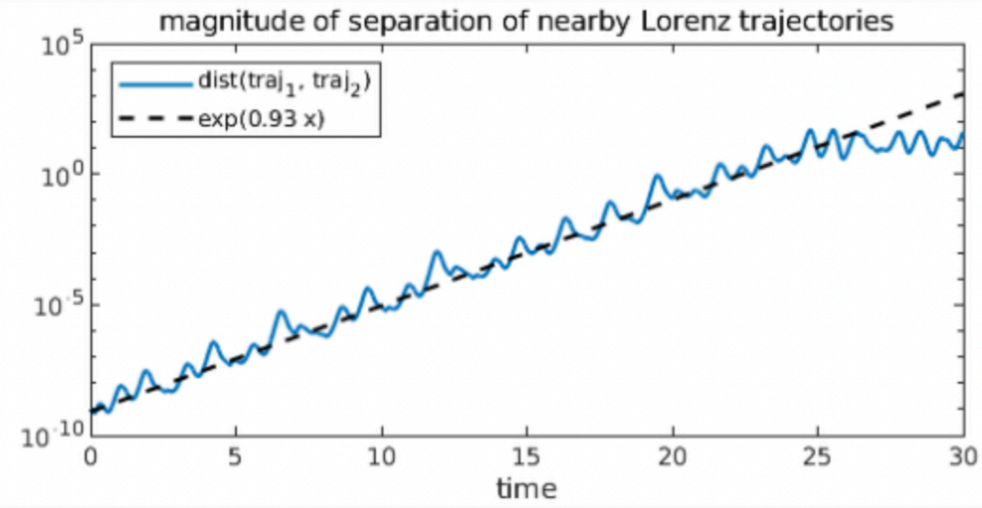


Estimating the Lyapunov exponent and dimension

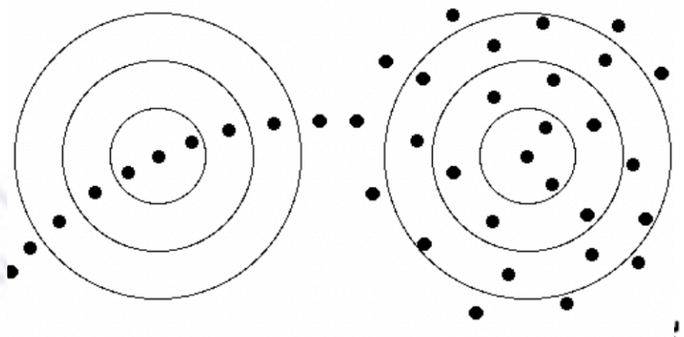
In the field of non-linear mathematics and physics of complex systems, the study of dynamics has been a major field in the past century

First in *mathematical* experiments and later in real experiments, numbers such as Lyapunov exponent and attractor dimension is estimated based on fitting tools - here always in log-space.



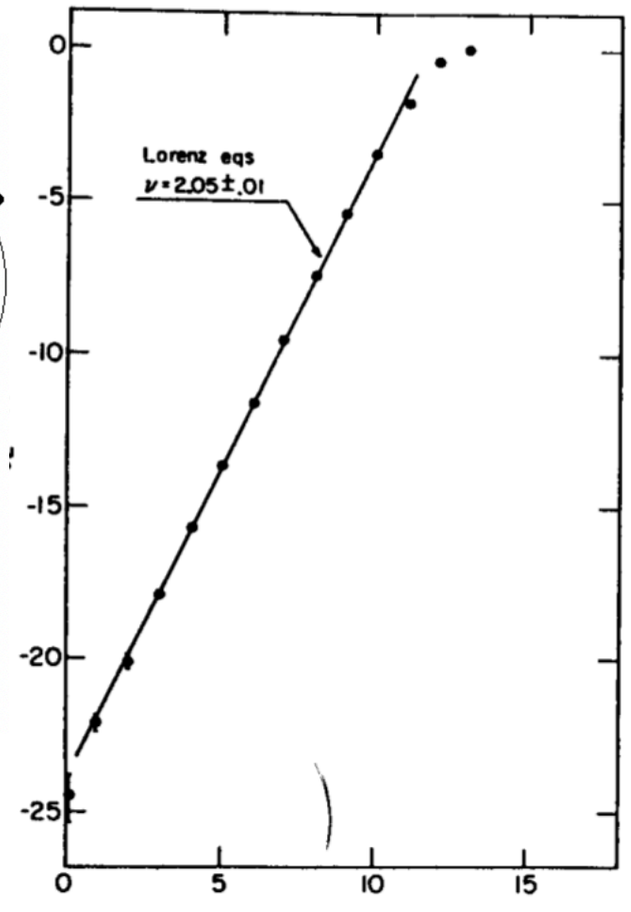


Correlation Dimension



$$C(r) = \frac{1}{N^2} \sum_{i,j}^{N} \theta(r - \Delta r)$$

where $\theta(x) = \begin{cases} 1 \text{ for } x > 0 \\ 0 \text{ for } x < 0 \end{cases}$

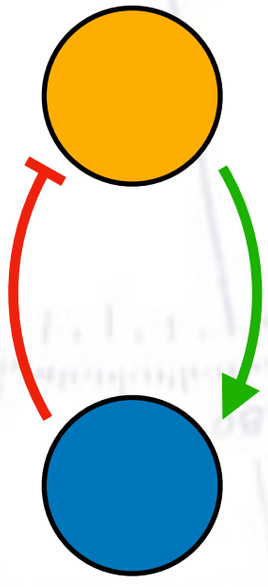


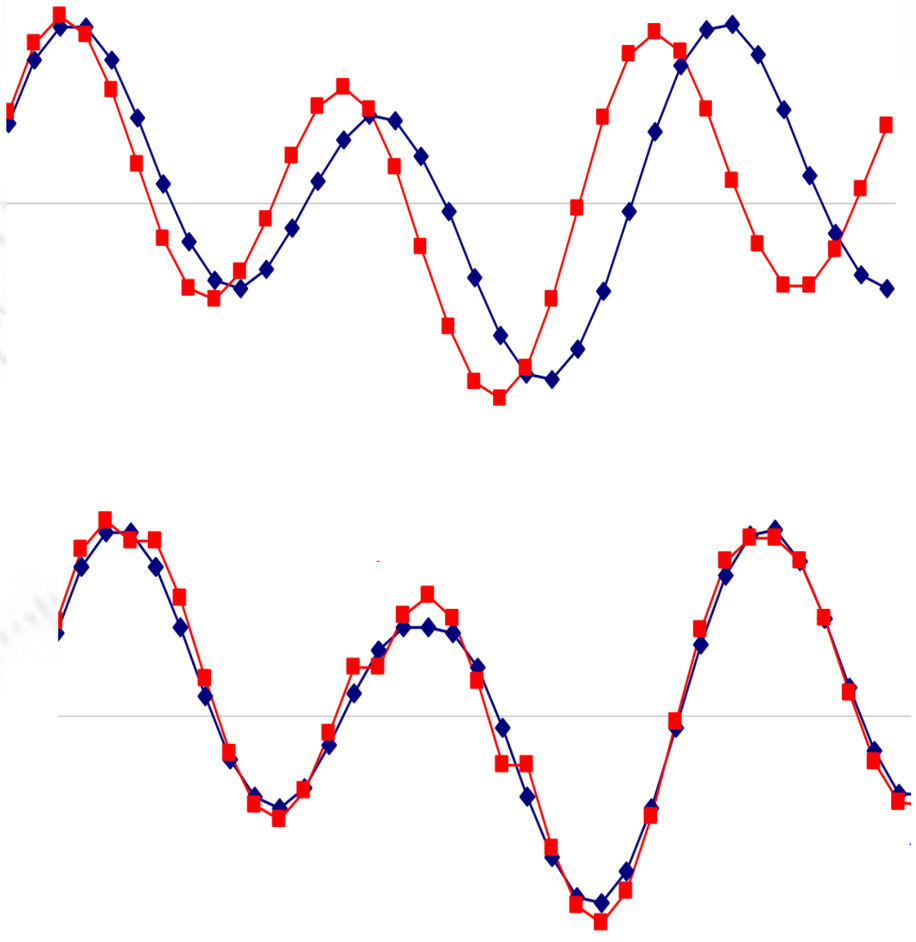
Dynamics Time Warping (DTW)

A very typical problem is the comparing of two data series that definitely seem similar

However comparing their similarities directly will not give any good result.

Instead we compare them using Dynamics Time Warping, where we consider the distances between the value of the points.





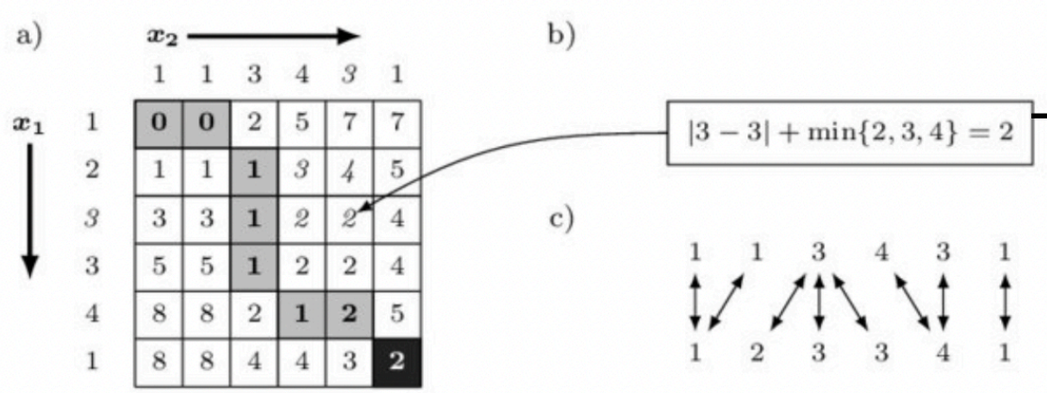
Dynamics Time Warping (DTW)

Based on the landscape of distances, we now create the so called “accumulated cost” landscape.

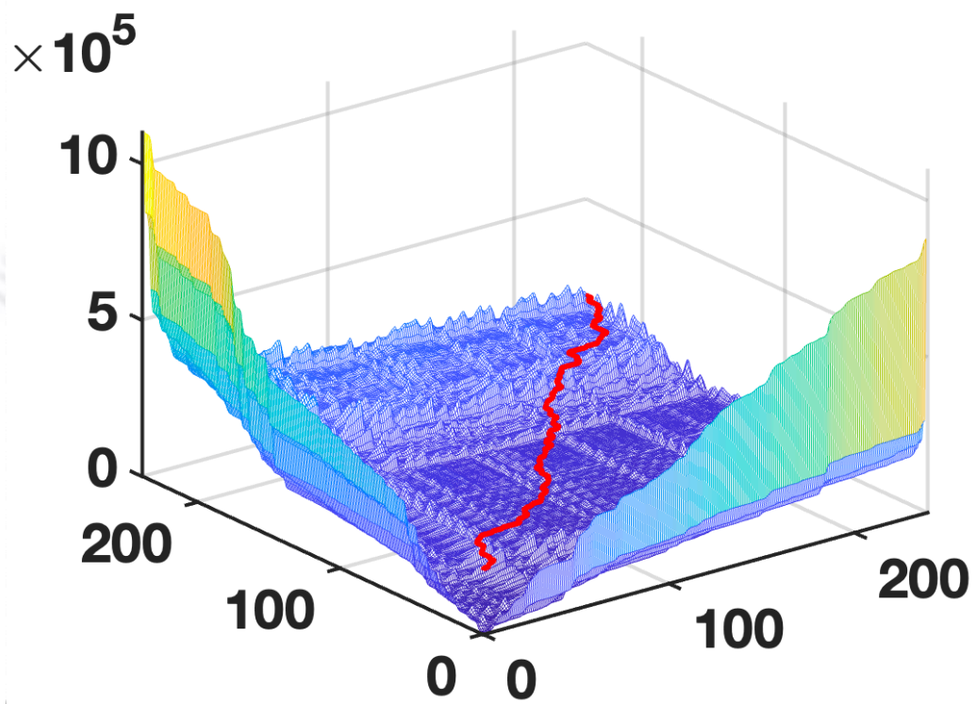
Here, for each new step, the value is the minimal of the three “previous” values plus the actual cost.

When we now move *backwards* - we find the path of minimal cost!

```
for i := 1 to n
    for j := 1 to m
        cost := d(s[i], t[j])
        DTW[i, j] := cost + minimum(DTW[i-1, j  ],    // insertion
                                    DTW[i  , j-1],    // deletion
                                    DTW[i-1, j-1])    // match
```



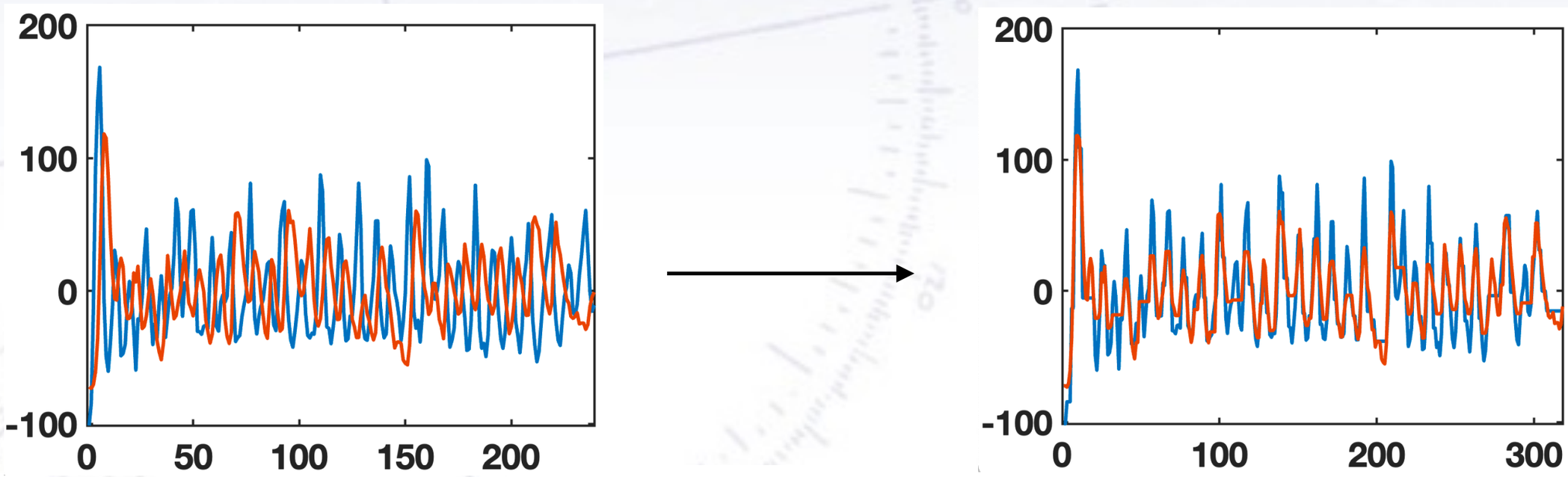
Cost landscape

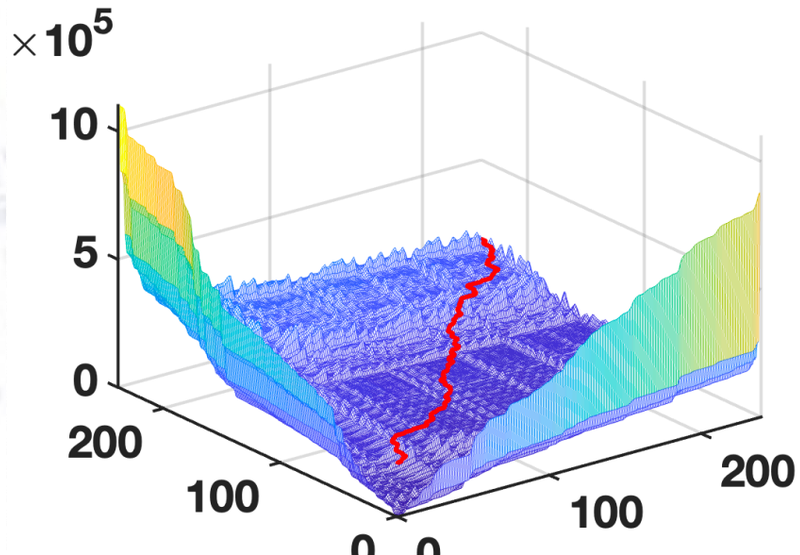


Dynamics Time Warping (DTW)

By application of the DTW, we can thereby construct aligned sequences in the time variable.

This could potentially also lead to the application of more standard fitting tools.





Summing up

In this brief lecture we have covered some fundamental elements in data analysis of time series:

1) Investigate stationarity - possibly append polynomial fitting

2) Investigate oscillatory components - apply Fourier analysis

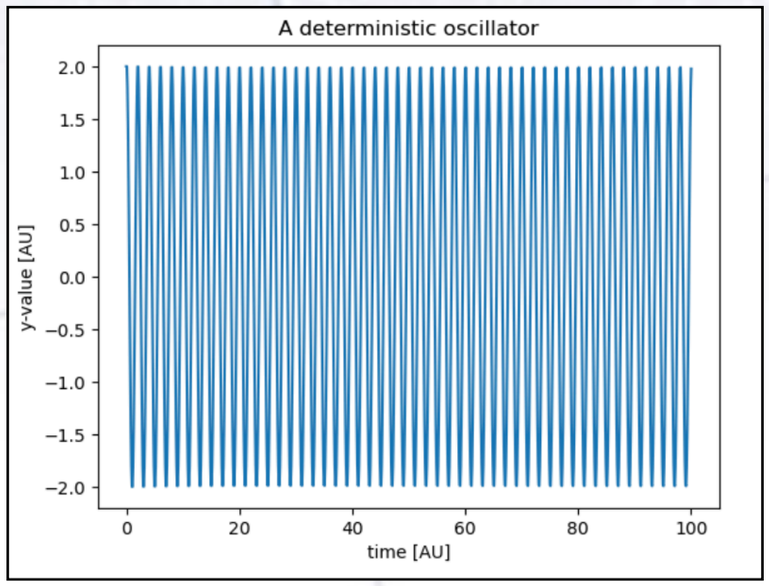
3) Consider if de-noising of data could strengthen the analysis

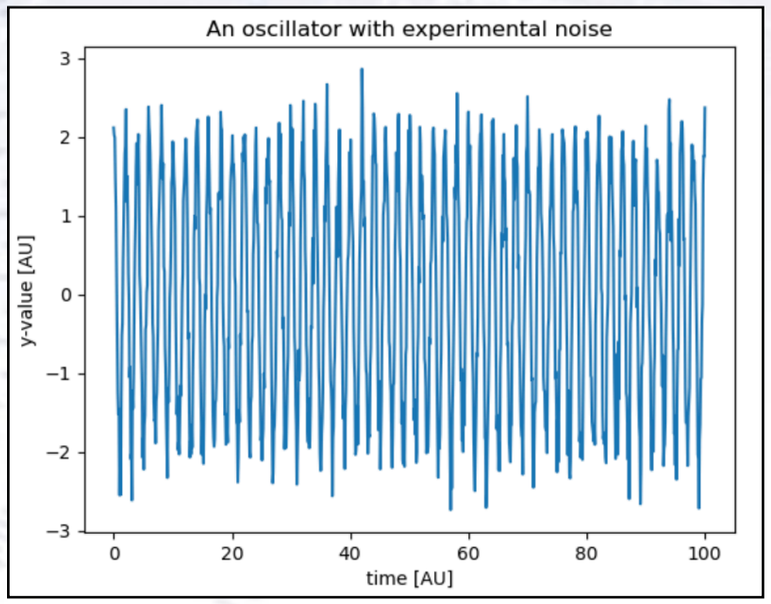
4) When comparing oscillatory signals - align the signals using dynamic time warping

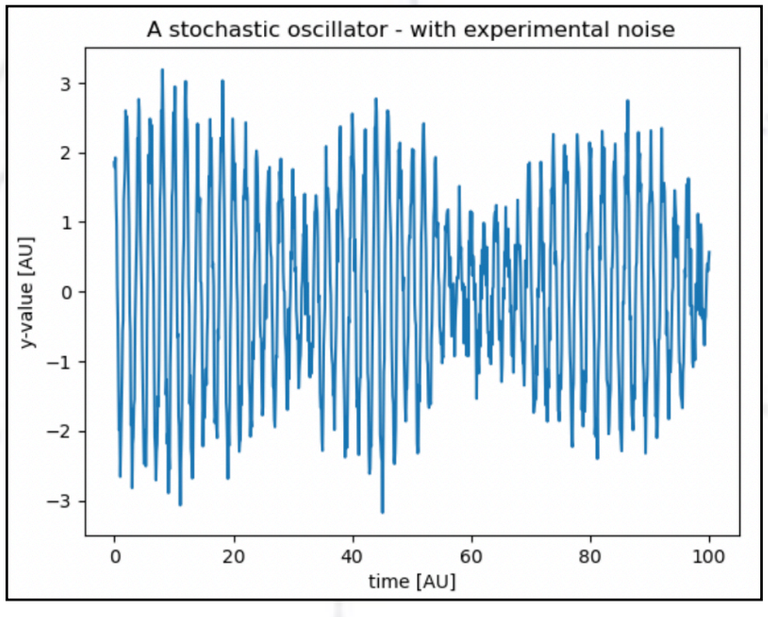
5) Investigate the phase space of the signal using time embedding

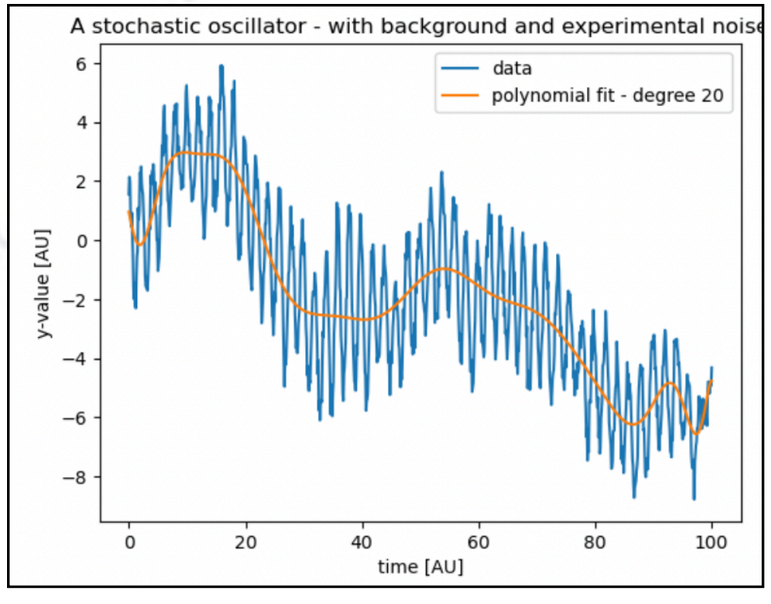
Today's exercise

Extract the frequency and amplitude of these oscillators…

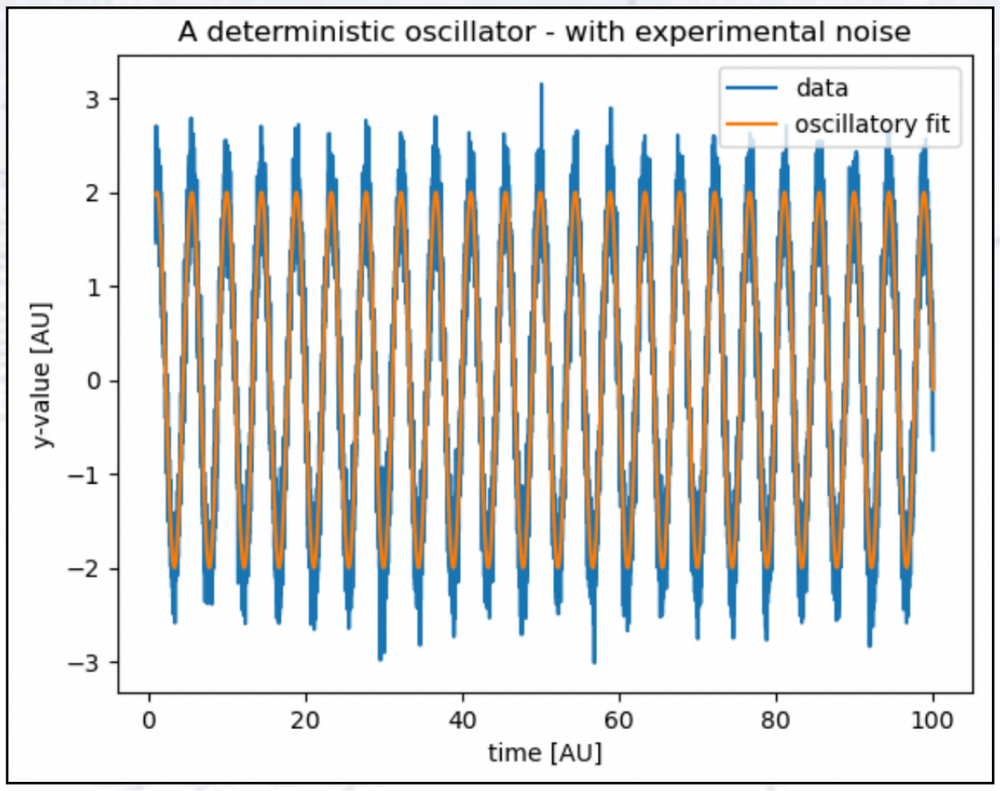


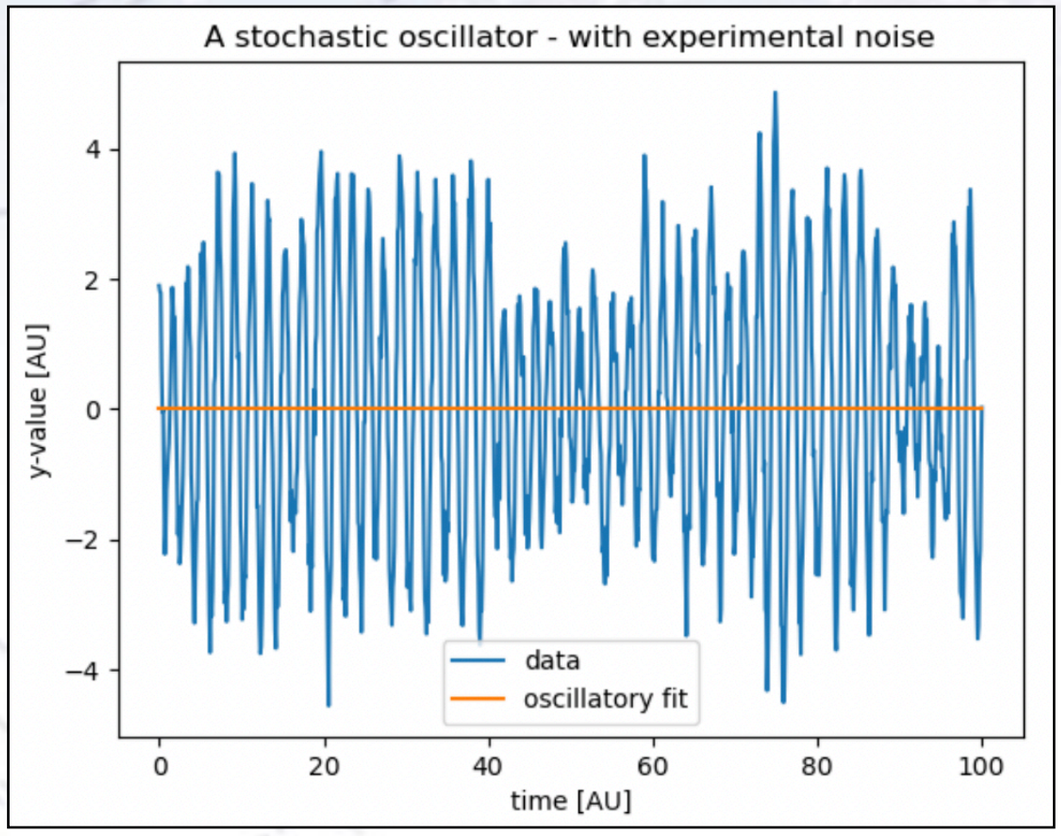






Today's exercise

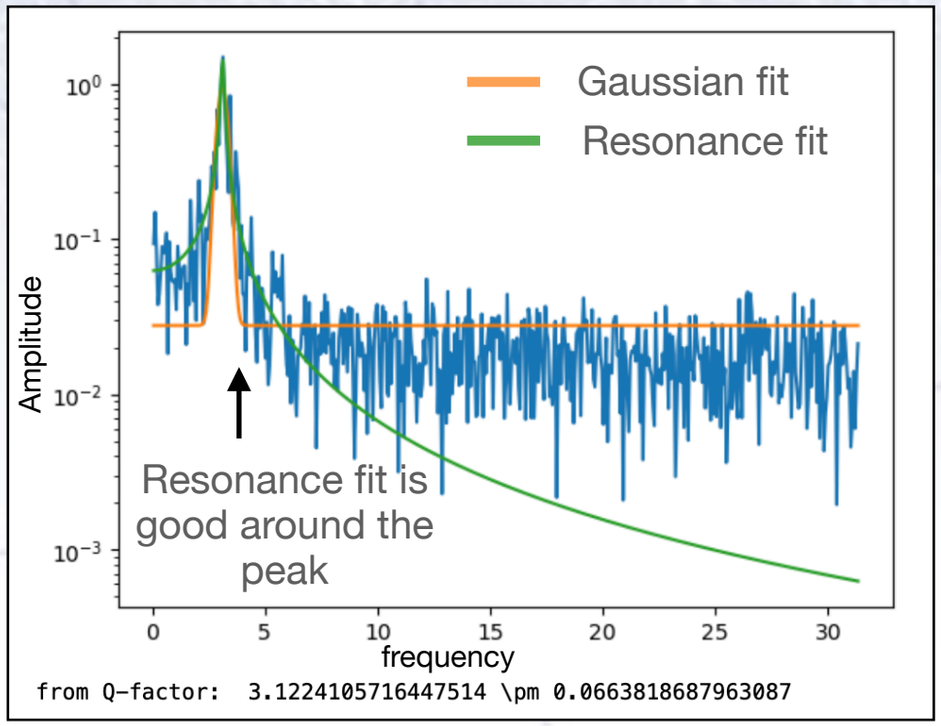




Note: See how much you can enhance sig_trace before the fit stops working!!

Today's exercise

Using Fourier analysis



from Q-factor: 3.1224105716447514 \pm 0.0663818687963087

Potential candidate for fitting:

$$A = \frac{F_{\max}/m}{\sqrt{\left(\omega_0^2 - \omega^2\right)^2 + \frac{\omega_0^2}{Q^2}\,\omega^2}}$$