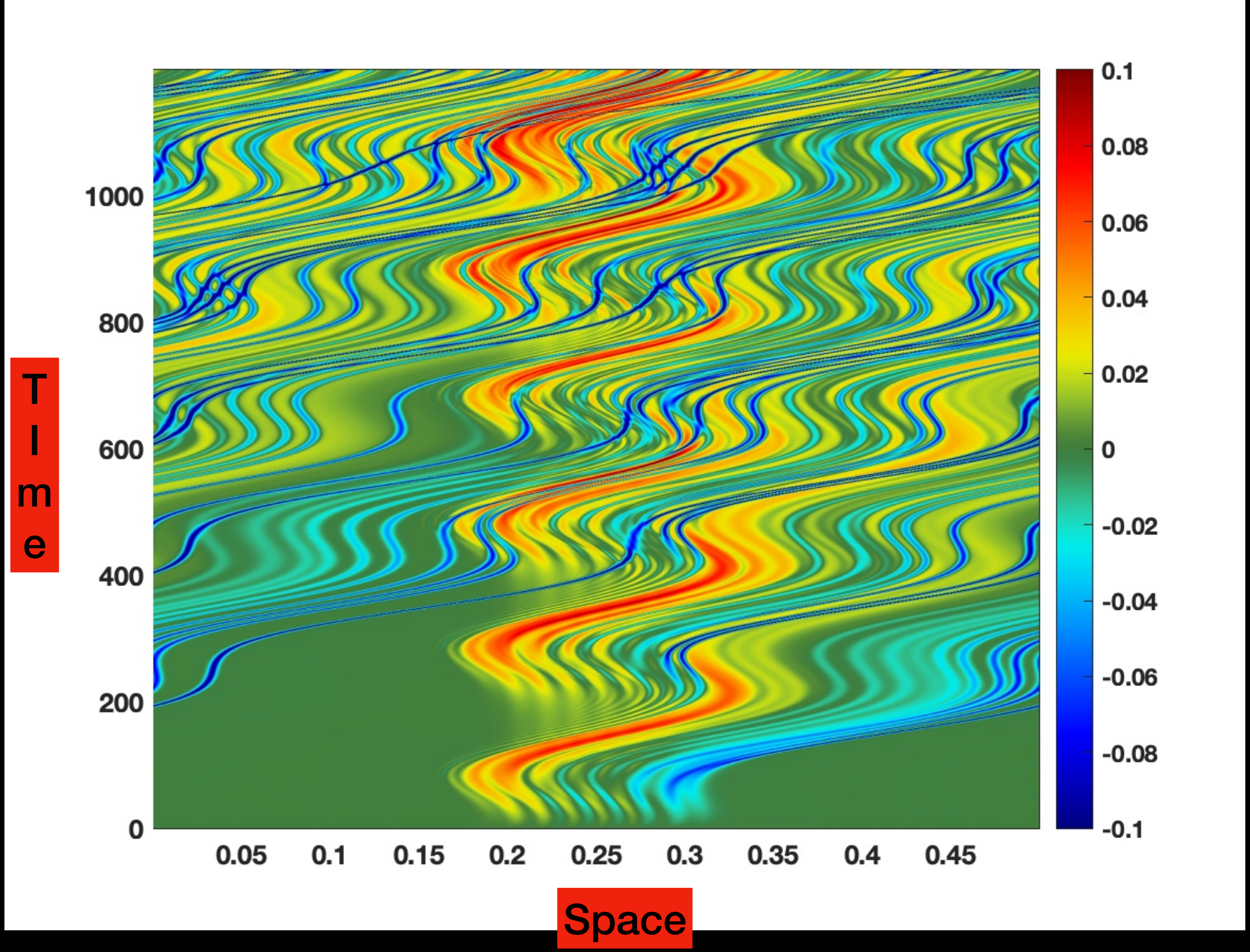
Quantitative Climate Science: Data Centric Methods, Fourier Methods

Marek Stastna Fall 2024

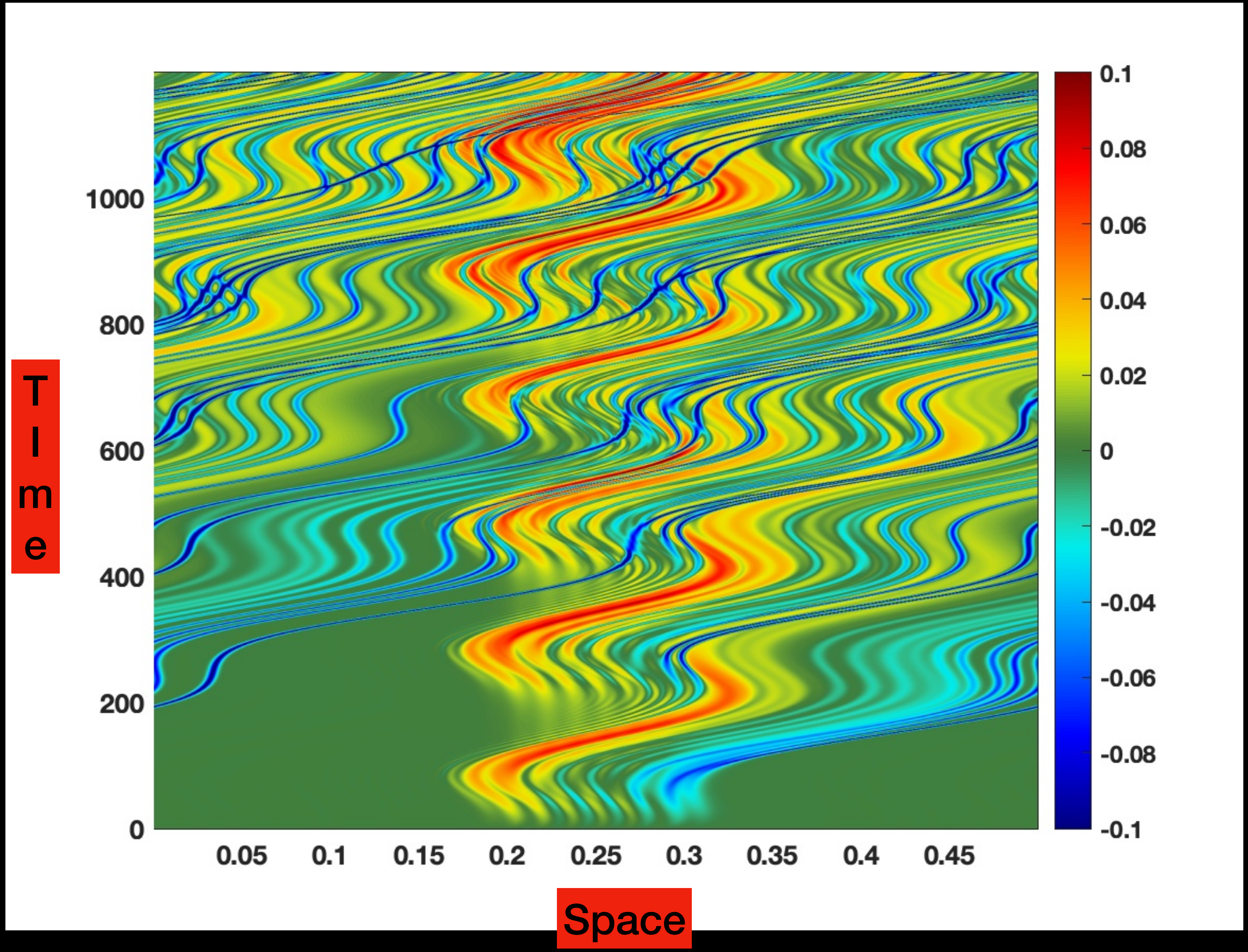
The data I

- Our “applied” focus will be on a model data set.
- The data set is generated by the pseudo spectral integration of a nonlinear wave equation (the forced BBM equation).
- This equation is a conceptual model of wave generation by tidal flow over a ridge.
- It is not meant to be realistic, but it does have several useful characteristics.



The data II

- The resulting waves are nonlinear dispersive waves and hence have non-trivial interactions (similar to, though not quite the same as, solitons).
- The picture on the right shows the solution as a space time plot
- There 2048 points in space and 1200 outputs in time.
- This is a non-trivial data set in terms of size, but still can be handled on a decent laptop.



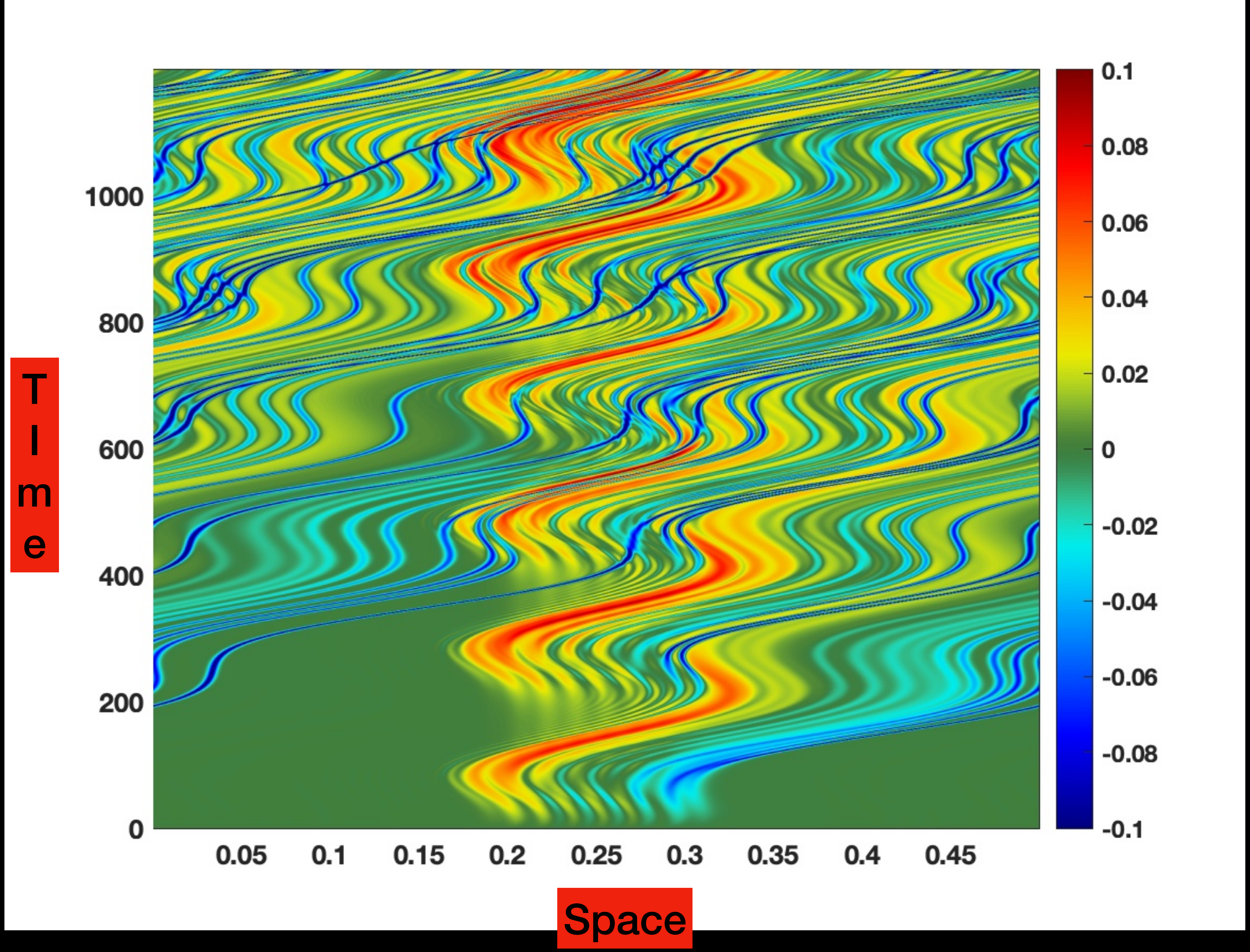
The data III

- You can see a lot visually:

- The repeated generation of waves near the centre of the domain (on the flanks of the topography).

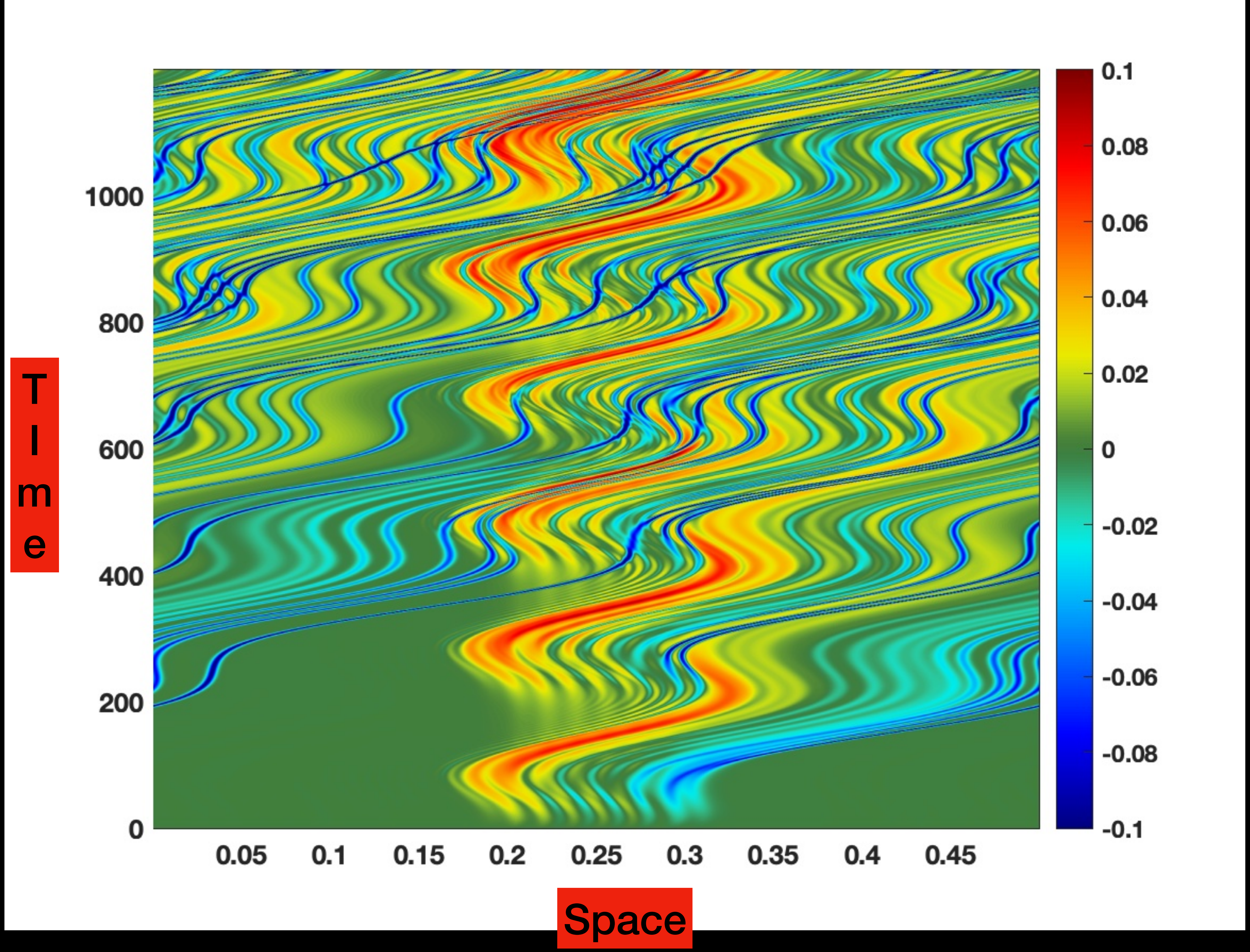
- The occasional stronger, narrower blue waves that pass through the periodic boundaries and interact with subsequently generated waves.

- What can we hope to learn about this data set without digging into the wave theory?



The data IV

- We could ask what the most efficient way to represent the data is.
- We could try to break the data down in terms of standard mathematical techniques (i.e. choose a well known basis like sines and cosines).
- We could use off-the-shelf techniques and try to visualize them.
- To do any of this, we are often borrowing from existing mathematics and so on occasion we will take mathematical detours.



Fourier, or Spectral Analysis

- Almost all students see some version of Fourier series in undergraduate studies.

- Mostly this is presented using a different “language” than how such analysis is used in practice.

- We assume some exposure, and aim to present key ideas in how to use such analysis to examine data like that shown on the last few slides.

Fourier Series: Textbook

$$F_n(x) = \sum_{k=0}^{n} c_k \exp(ik\pi x) \quad c_k = C\int_{-\pi}^{\pi} f(x)\exp(ik\pi x)dx$$

- Consider a periodic function f(x) on $[-1,1]$ and choose a basis of sines and cosines.
- Project f(x) onto the basis (C is a normalization constant)
- Have very rapid convergence (exponential, so faster than any power!).
- Also have a sense of preservation of size, or energy, in Parseval's theorem.

Fourier Series: Time series

$$F_n(t) = \sum_{k=0}^{n} c_k \exp(i\omega_k t) \qquad c_k = C\int_0^T f(t)\exp(i\omega_k t)dt$$

- Consider a periodic function f(t) on [0,T] and choose a basis of sines and cosines.
- Project f(t) onto the basis (C is a normalization constant)
- The smallest frequency is $\omega_{nyq} = \frac{2\pi}{T}$ so that $\omega_k = k\omega_{nyq}$

For functions of time we generally use $[0,T]$. For space we often take $[-L,L]$

Parseval's Theorem

- https://en.wikipedia.org/wiki/Parseval%27s_theorem
- Parseval's theorem tells us about the relation between the root mean square of a function and its Fourier series.
- In numerical applications the term "Discrete Fourier Transform" or DFT is often used instead of Fourier Series.
- Parseval's theorem's main implication is that you can think of a function "one frequency at a time".

Matrix Algebra: Projections

$$\hat{e}_1 = (1,0,0), \hat{e}_2 = (0,1,0), \hat{e}_3 = (0,0,1),$$

$$\vec{v} = \sum_{k=1}^{3} c_k \hat{e}_k \textbf{ where } c_k = \vec{v} \cdot \hat{e}_k$$

- For concreteness consider 3 dimensional space.
- Take the standard basis.
- Any vector can be written as a weighted sum of the basis vectors.
- Each coefficient is a projection of the vector onto the correct basis vector.

Matrix Algebra II: other bases

$$\hat{e}_1 = (1,0,0), \hat{e}_2 = (0,1,2), \hat{e}_3 = (2,0,1),$$

$$\vec{v} = \sum_{k=1}^{3} c_k \hat{e}_k \textbf{ where } c_k = \vec{v} \cdot \hat{e}_k$$

What property must a basis have? It should span the whole space of interest, so that the vectors in it are linearly independent

- We could have started with a different basis.
- We could then use the Gramm-Schmidt process to get an orthonormal basis (basis vectors have length one and are orthogonal to each other).
- We could also choose any valid inner product.

Fourier Series: Generalized

$$F_n(x) = \sum_{k=1}^{n} c_k \phi_k(x) \qquad c_k = \langle f(x), \phi_k(x) \rangle$$

- What if didn't want to use sine and cosine?
- The idea is the same for a general basis.
- We just have to choose a convenient basis and an inner product we are prepared to work with.

Fourier Series: machinery

$$\int_{-1}^{1} \sin(n\pi x)^2 dx = \int_{-1}^{1} \cos(n\pi x)^2 dx = 1$$

$$< f, g >= \int_{-1}^{1} f(x)g(x)dx$$

$$b_n =< f, \cos(n\pi x) > \quad , \; a_n =< f, \sin(n\pi x) >$$

- On [-1,1] it is easy to show that the sines and cosines are an orthonormal basis and we get back the Fourier series.
- So there is a theoretical reason to think of a function (or signal) one frequency at a time as well.

Practice versus Theory

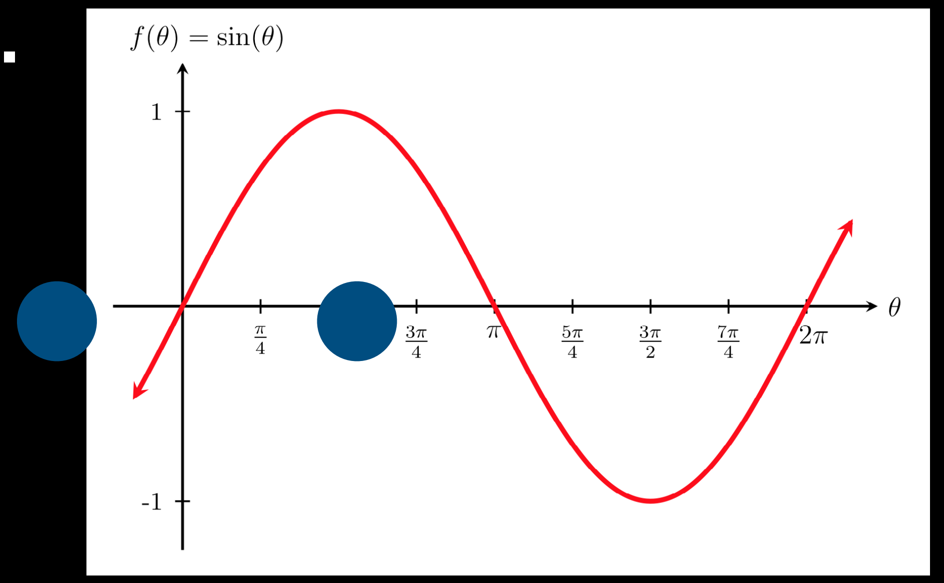
- Given how often spectral analysis is carried out you have some choices if you want to do it yourself.
- You could use various packages (e.g. Matlab's Signal Processing Toolbox). This means you need to learn and use the notation of the authors (often electrical engineers).
- You can also build simple analyses from scratch (my preference).
- Historically, time series analysis exploded in terms of usefulness with the (re)discovery of the Fast Fourier Transform in them mid 1960s.
- https://en.wikipedia.org/wiki/Fast_Fourier_transform (a classical reference)
- https://jakevdp.github.io/blog/2013/08/28/understanding-the-fft/ (more python based)

FFT basic frequencies

- Let's consider a time series x_i=f(dt*i) where dt is a time step.
- We assume that we have measured (or simulated) so as to have the function we want at times $t = \Delta t, 2\Delta t, \ldots, T = N\Delta t$
- We can ask what are the smallest and largest frequency we can represent in terms of sinusoids?
- Once we know the smallest frequency we can then consider discrete frequencies $\omega_n = n\omega_{min}$

FFT basic frequencies

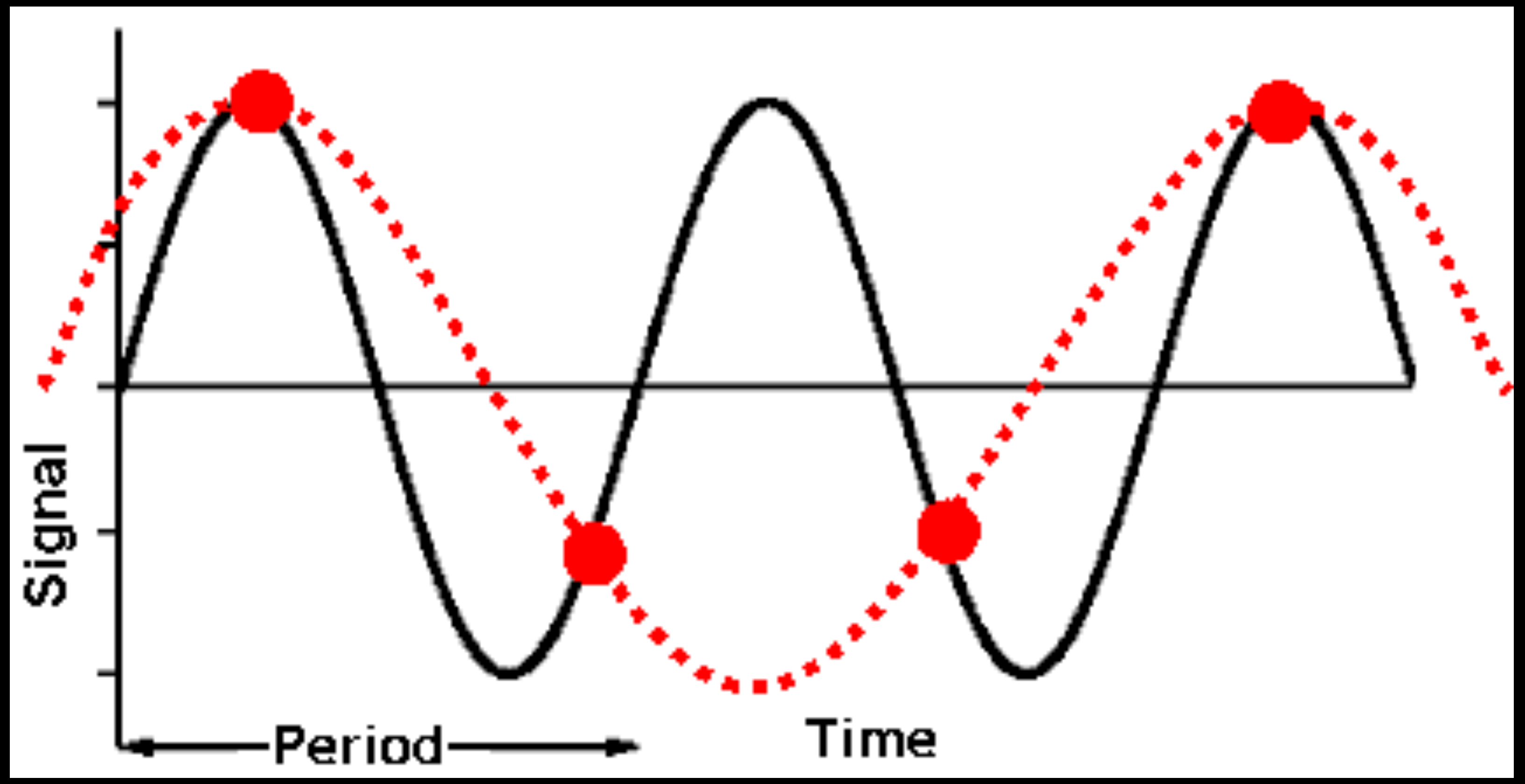
- We can ask what are the smallest and largest frequency we can represent in terms of sinusoids?
- The smallest is easy to find just set $\omega_{min} = \dfrac{2\pi}{T}$. Note that the average of the signal has $\omega = 0$ and FFT implementations will most often give that as the first entry of the transformed vector.
- To find the biggest think about how you'd characterize a sinusoid: At least you would want to know when it hits zero, so two points per period (the blue dots below).
- Thus the highest possible frequency is $\omega_{max} = \dfrac{2\pi}{2\Delta t} = \dfrac{\pi}{\Delta t}$.



- Of course just because it's allowable doesn't mean the representation is accurate, and in practice we rarely look at frequencies near ω_{max}

Aliasing

- If we only have discrete data more than one sinusoid could fit that data (the four red points on the right).
- This is the phenomenon of aliasing.
- In practice all simulations and measurments have a cut off.
- This one has to be very careful about the high frequencies.
- https://en.wikipedia.org/wiki/Aliasing
- Aliasing is a broadly observed phenomena with many amelioration techniques.



Frequencies versus wavenumbers

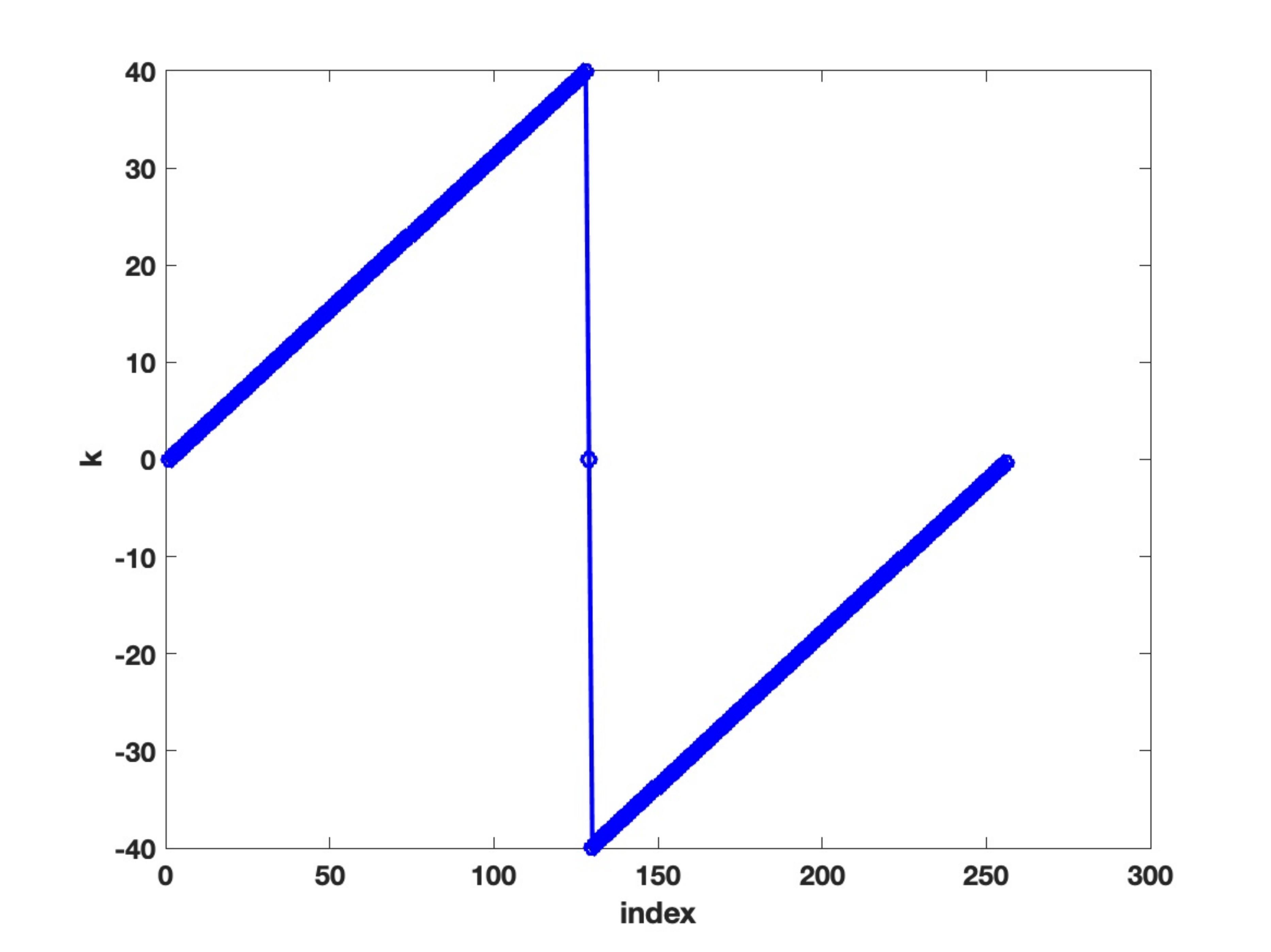
- It is also possible to consider Fourier/spectral analysis in space.
- This is often done with idealized simulations since periodicity in at least one direction is common.
- The grid would thus be x_k and the sample function/data $g_k = g(x_k)$
- The smallest wavenumber would be $k_{min} = \frac{2\pi}{L}$ where L is the length of the domain. Note for a symmetric domain $[-L, L]$ we would get $k_{min} = \frac{2\pi}{2L} = \frac{\pi}{L}$.
- Note that $k = 0$ corresponds to the spatial average.

Wavenumber ordering

- There is a common way to store wavenumbers that FFT implementations often use.

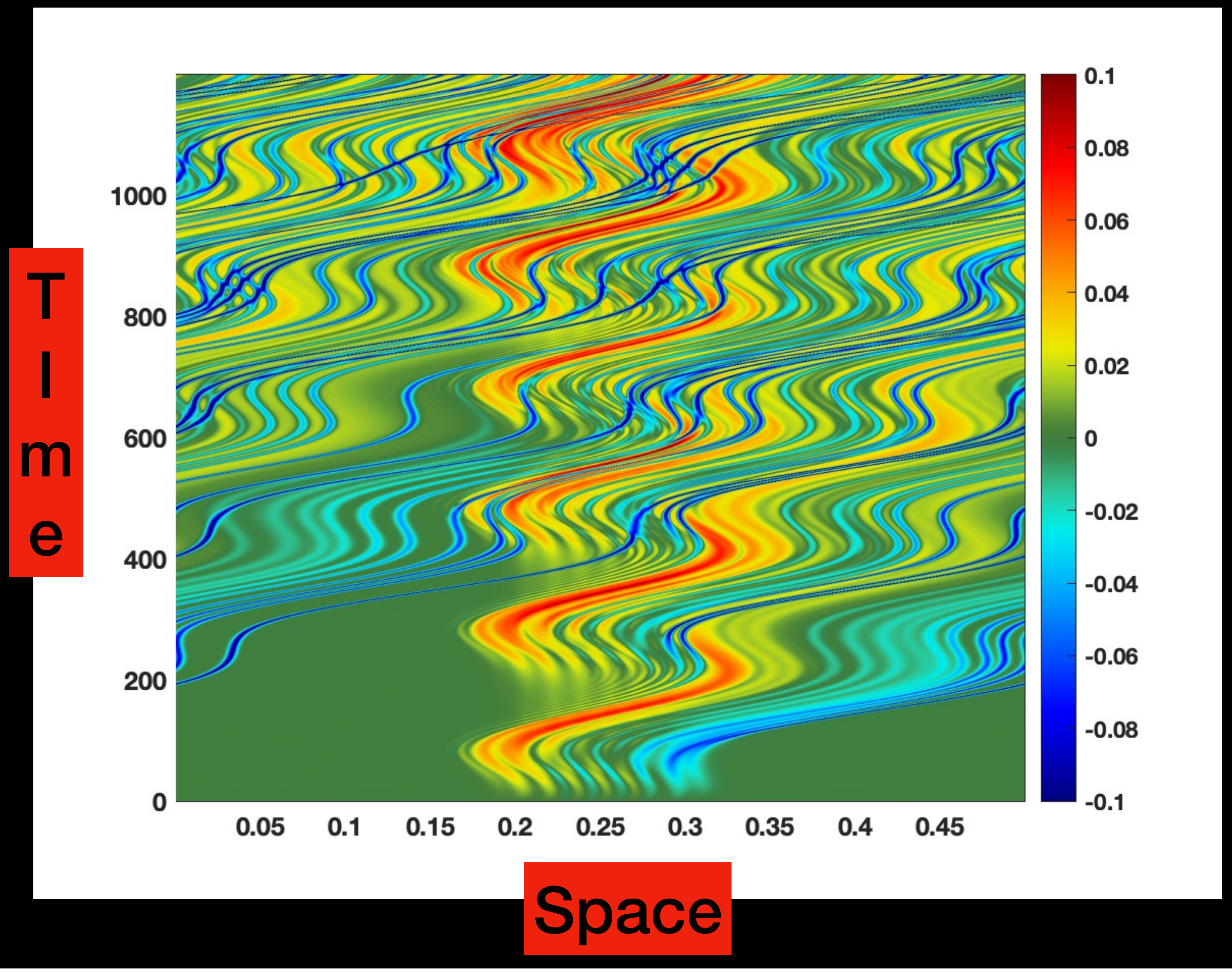
- The idea is to store $k = 0$, then the positive wavenumbers, then the negative wavenumbers.

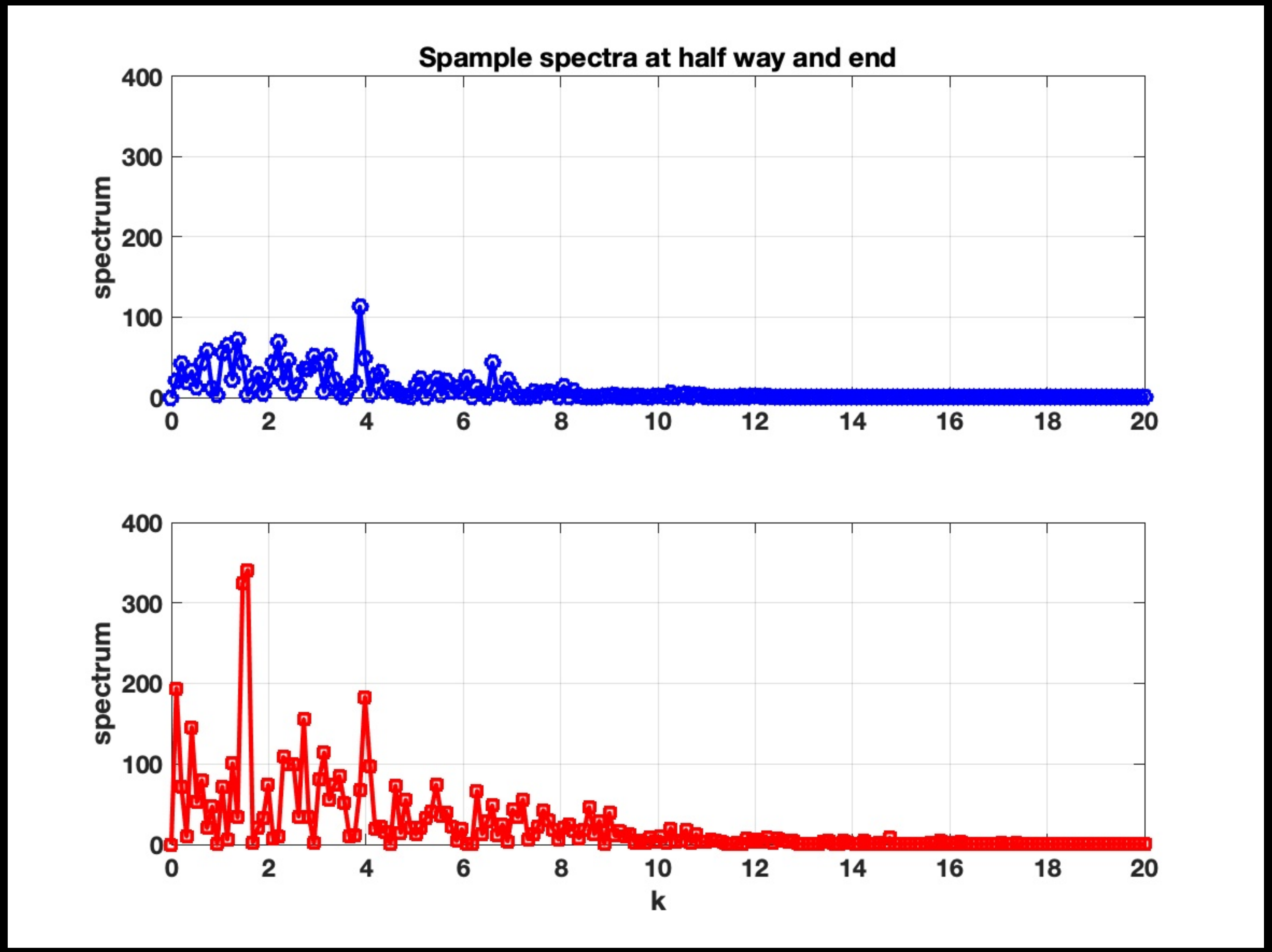
- If you only want to look at the first bit of the spectrum you do not need to build all of the k vector. You could just write myks=(0:numks)*kmin to get the average and the first numbs k values.



Sample spectra I

- I consider the data half way through the simulation and at the end.
- I compute the spectrum of the spatial FFT.
- I show the spectra focusing on low k values.
- You can see the spectrum is pretty choppy.
- The total “energy” seems to increase as the simulation gets busier.
- It is a bit tough to make stronger conclusions than that.



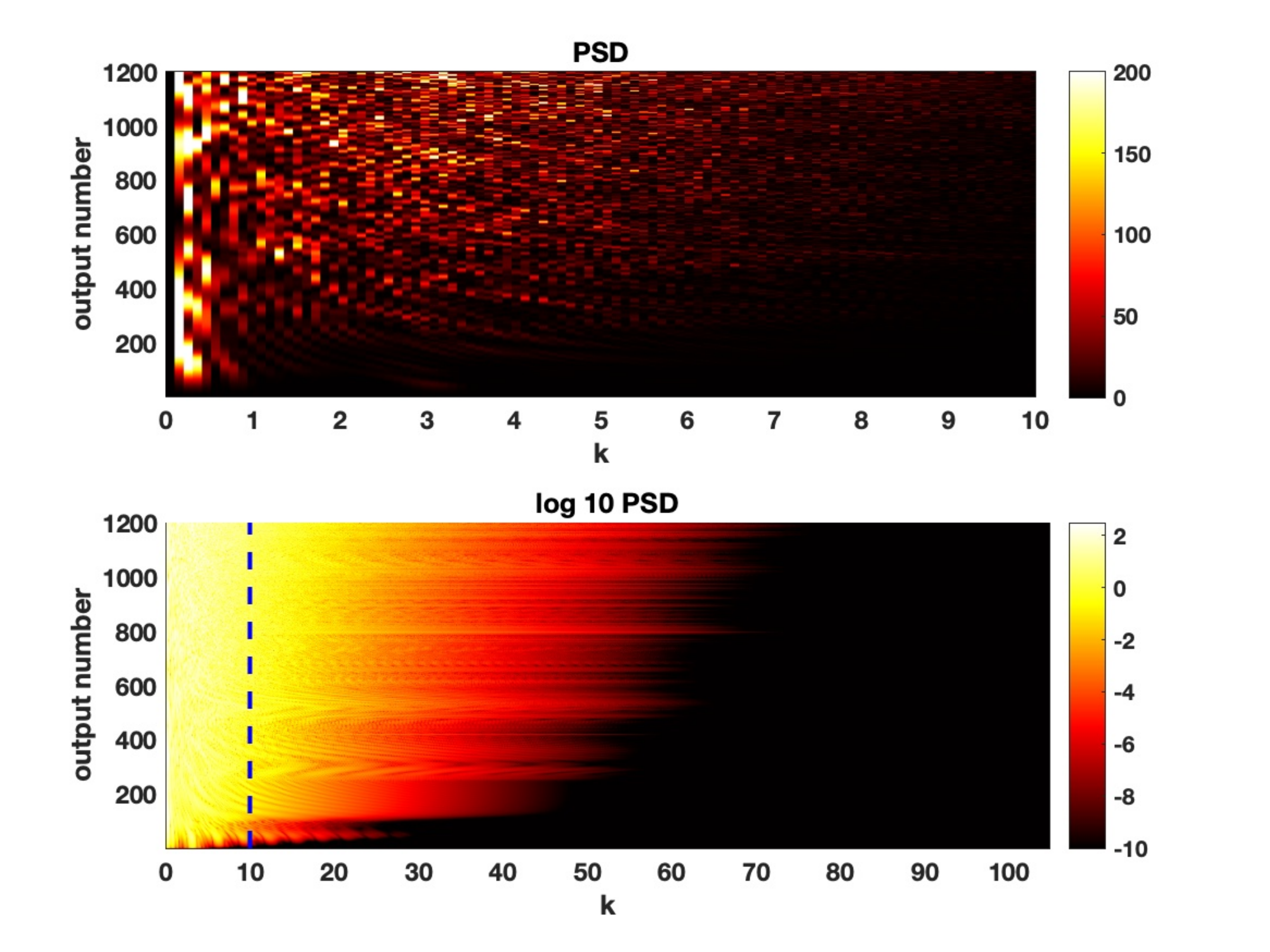


Sample spectra II

- One way to get around the messy picture of the spectra at one time is to consider the time evolution of the spectrum through what are called space-time or Hovmoeller plots.

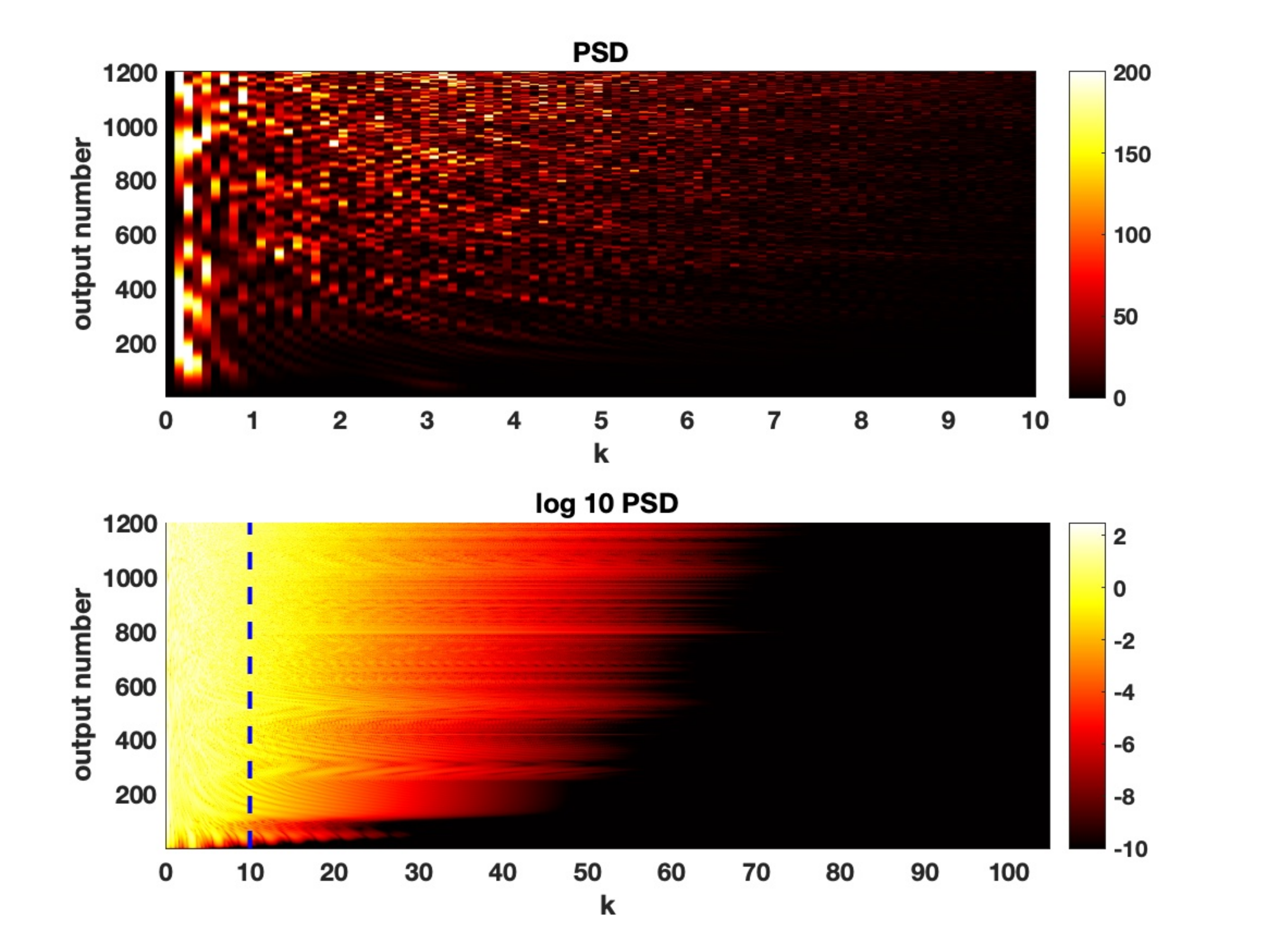
- In the upper panel we colour in the spectrum for k values up to 10. You can see the very low values of k (1, 2, 3) with some splotchy behaviour at higher k.

- For longer times we seem to get larger values of k excited.



Sample spectra III

- In the lower panel we colour in the log10 of the spectrum for k values up to 100.
- For longer times the shift to larger values of k excited is clearly visible.
- You do have to be careful when interpreting because the colorbar is now powers of 10. So the deep oranges are actually pretty small values of power.

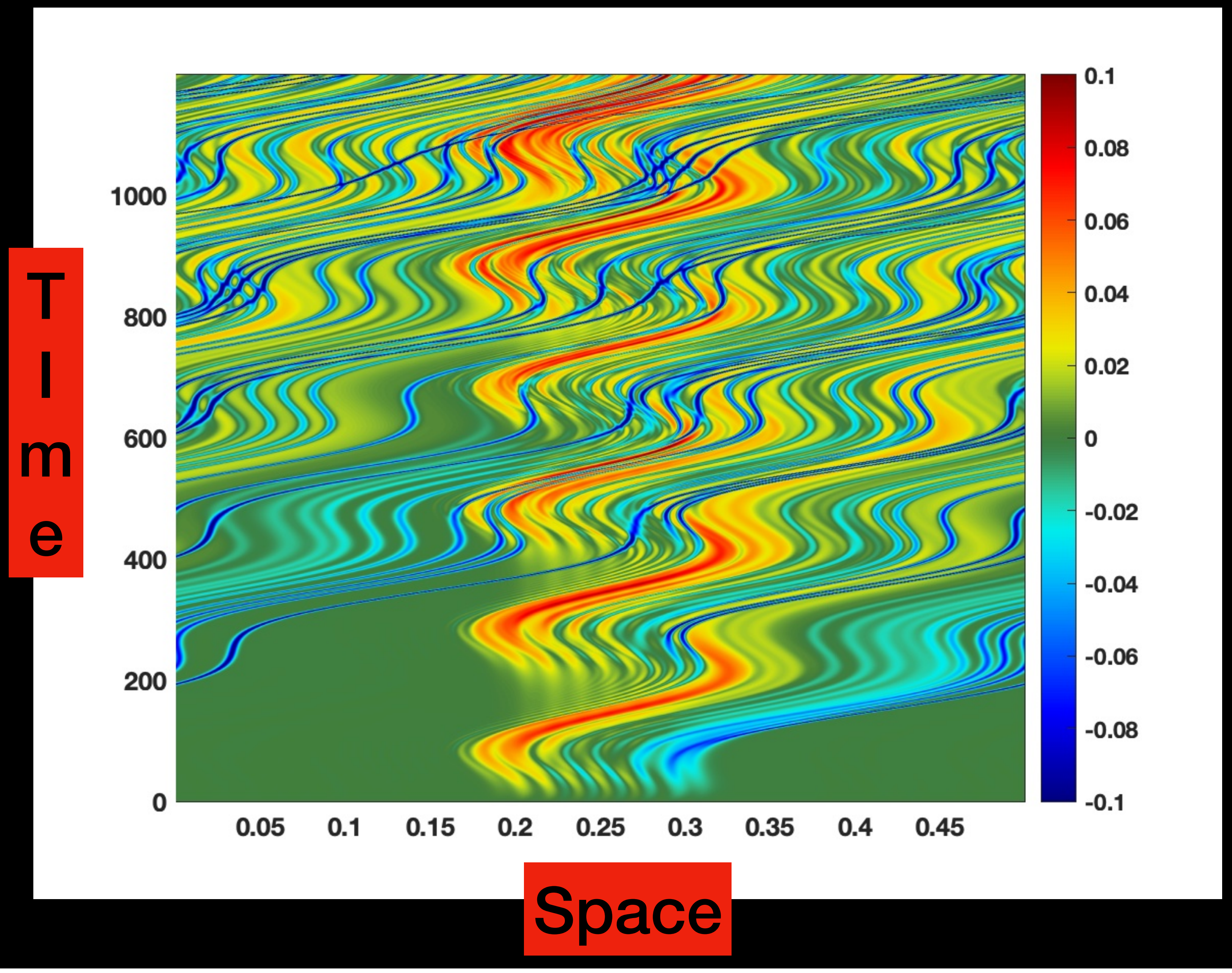


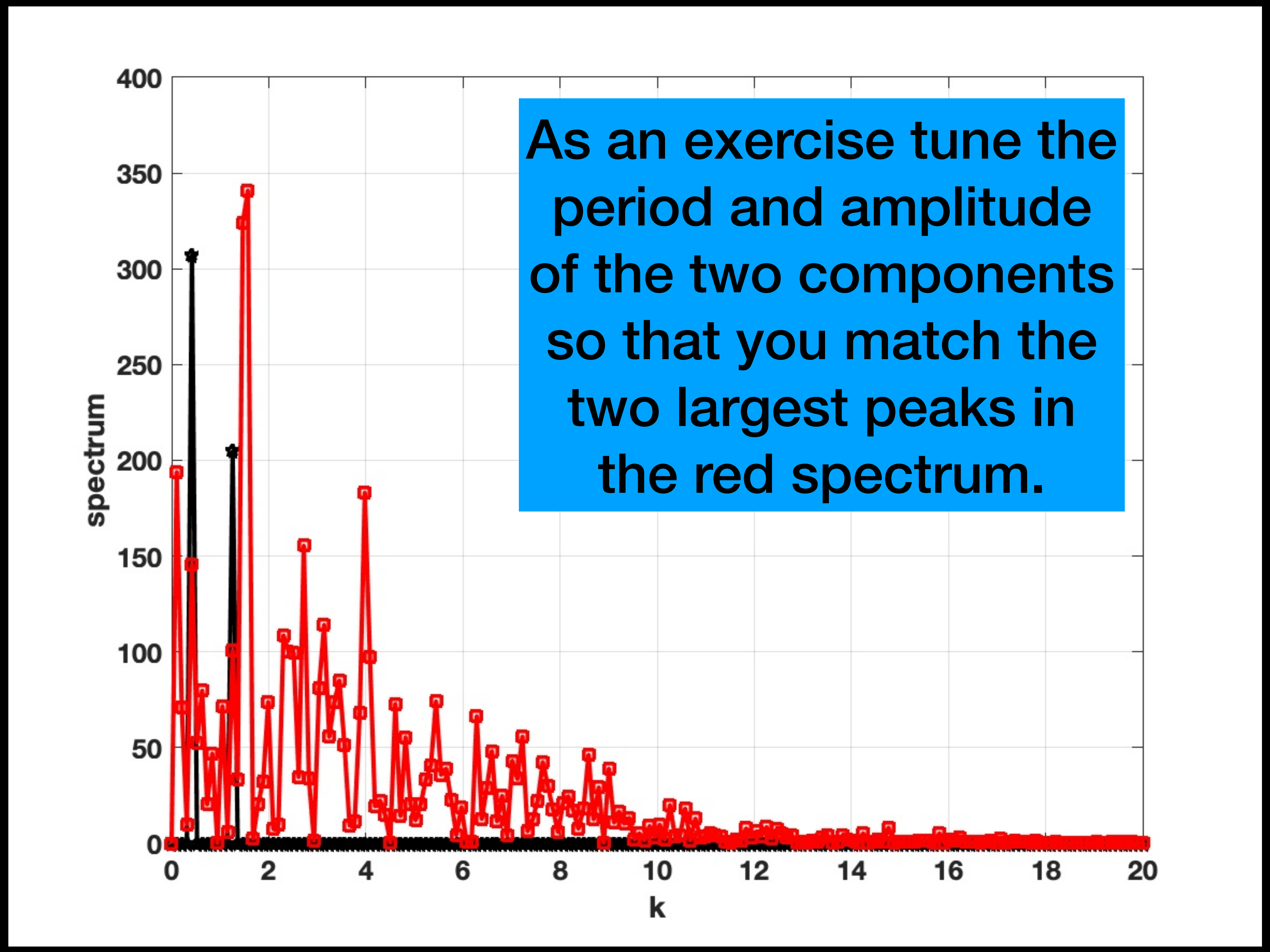
Back to sample spectra

- Let's reconsider the spectrum from the data and compare it to the spectrum of a function we know.
- In the picture on the right I plot the spectrum from data in red, and the spectrum of

$$g(x) = 0.3\sin\left(\frac{2\pi x}{15}\right) + 0.2\cos\left(\frac{2\pi x}{5}\right)$$

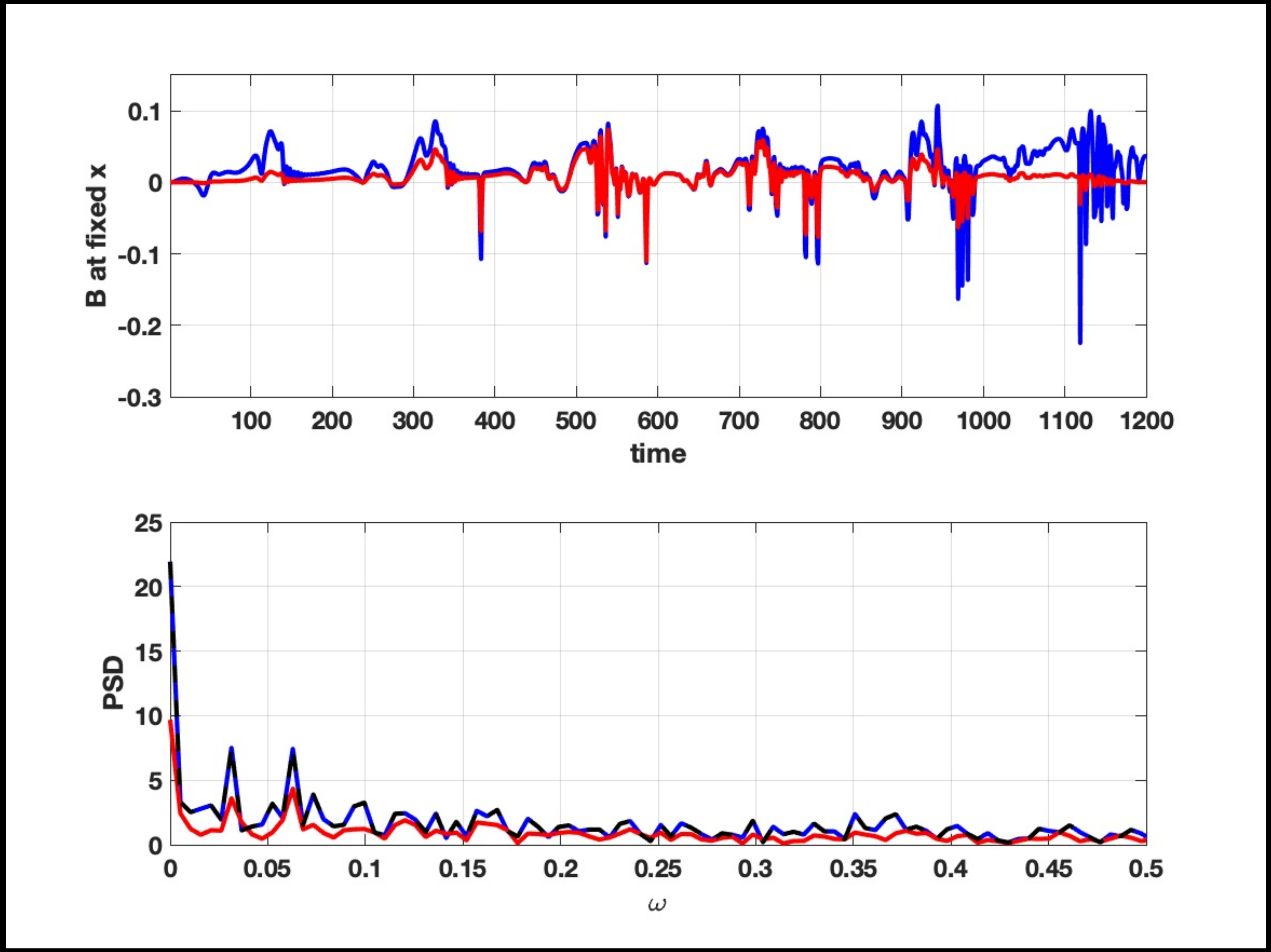
- You can see the simple function has a simple spectrum, so that the busy spectrum of the data is genuinely indicative of how busy the data is.





Spectra in time I

- OK now let's consider spectra in time.
- We fix a location near the source (the 920th grid point) and extract a time series.
- The blue line in the upper plot is the raw data.
- You can see right away the data is not periodic.
- Nevertheless we compute the spectrum shown by the blue line in the lower panel, and everything looks pretty good.

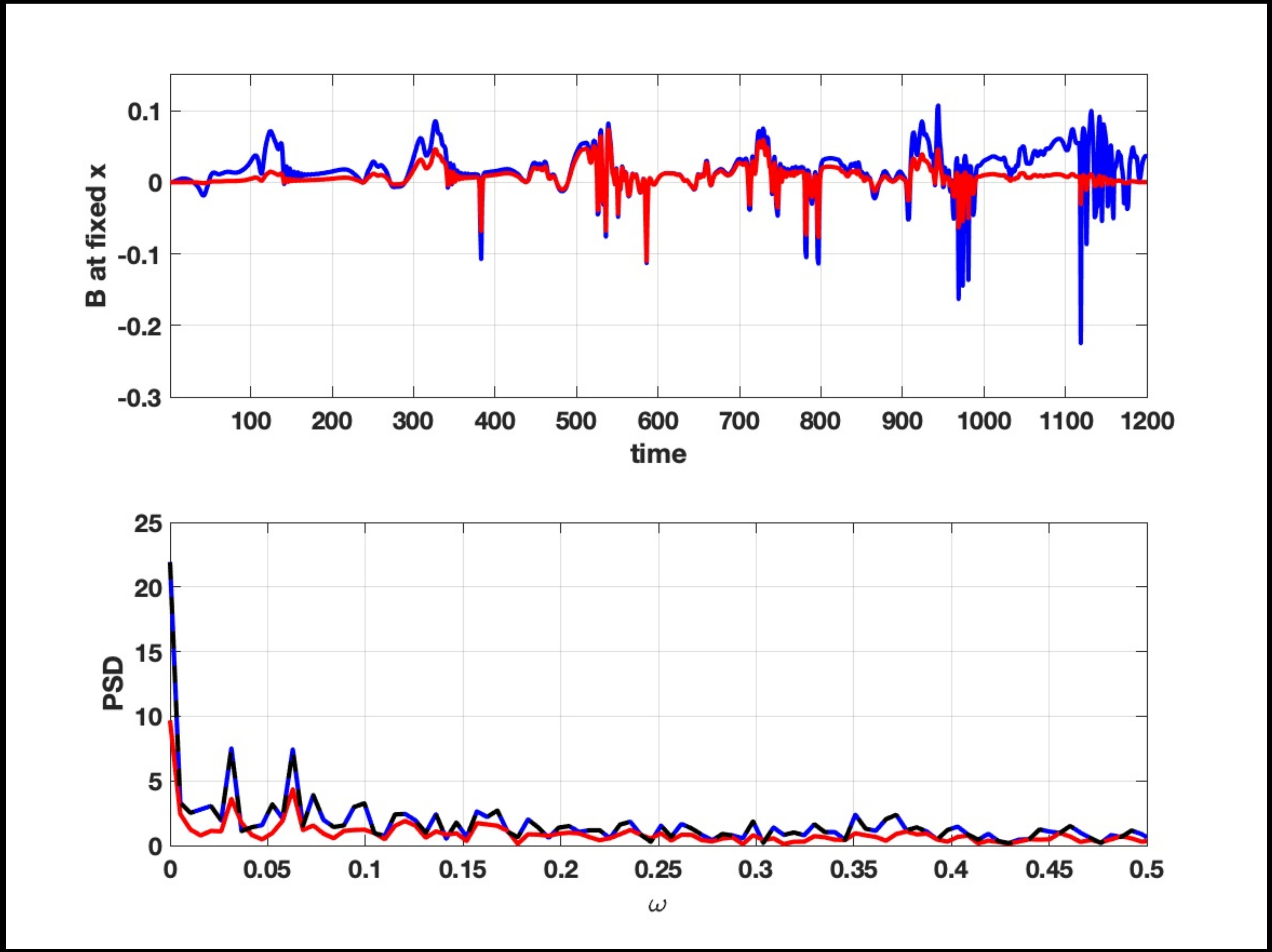


Spectra in time II

- Despite of the fact that the spectrum looks good, if we return to our Fourier series days we may recall we expect a problem from the jump in the periodic extension of the data (Gibbs phenomenon).

- There are two practical ways to deal with this. The result of the first is shown as a black dashed line in the lower panel.

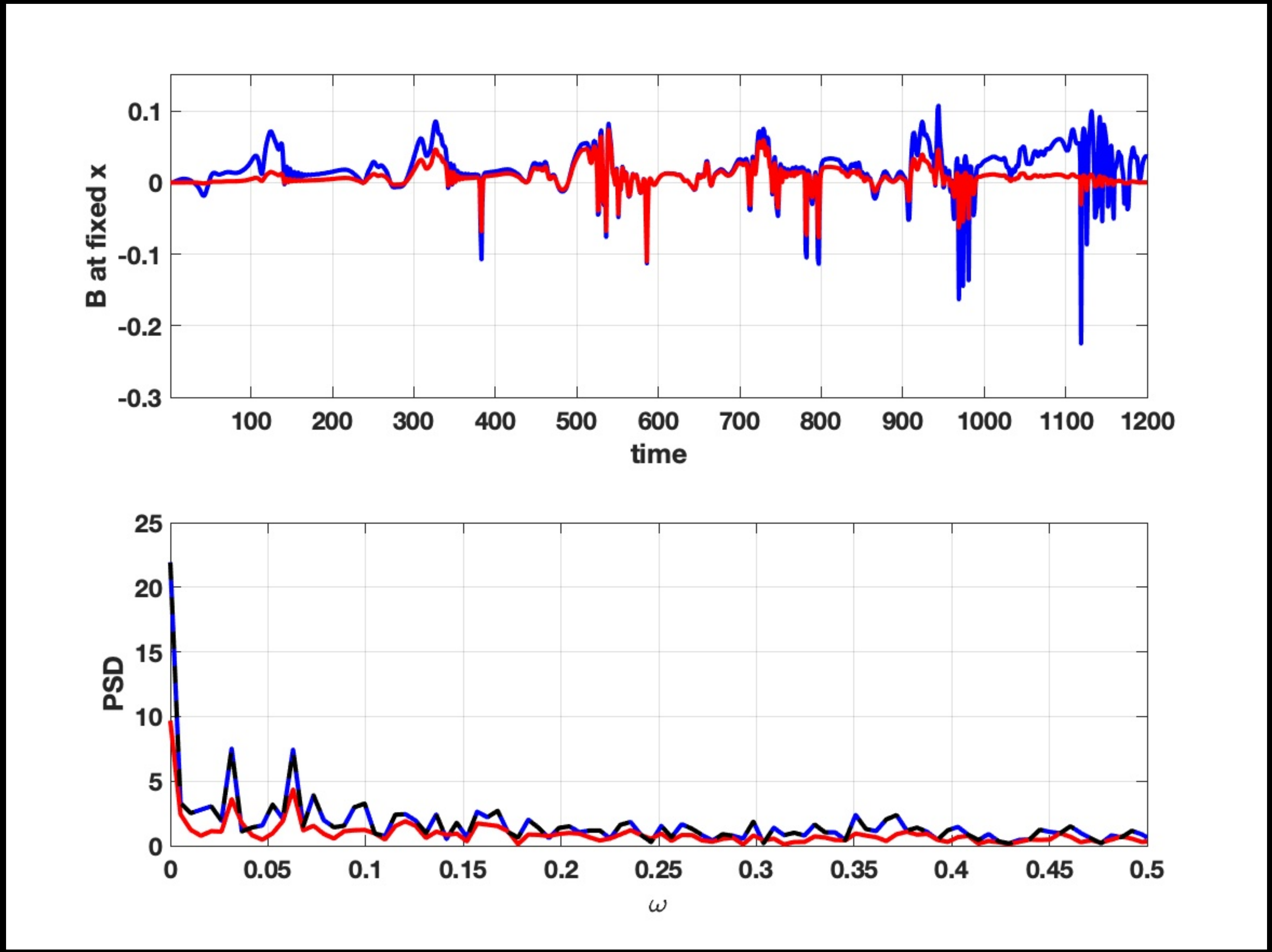
- We simple create an even extension of the data by doubling the length of our time series and then compute the spectrum of this periodic (by construction) function.



Notice the original and even extension spectra match for the frequencies shown.

Spectra in time III

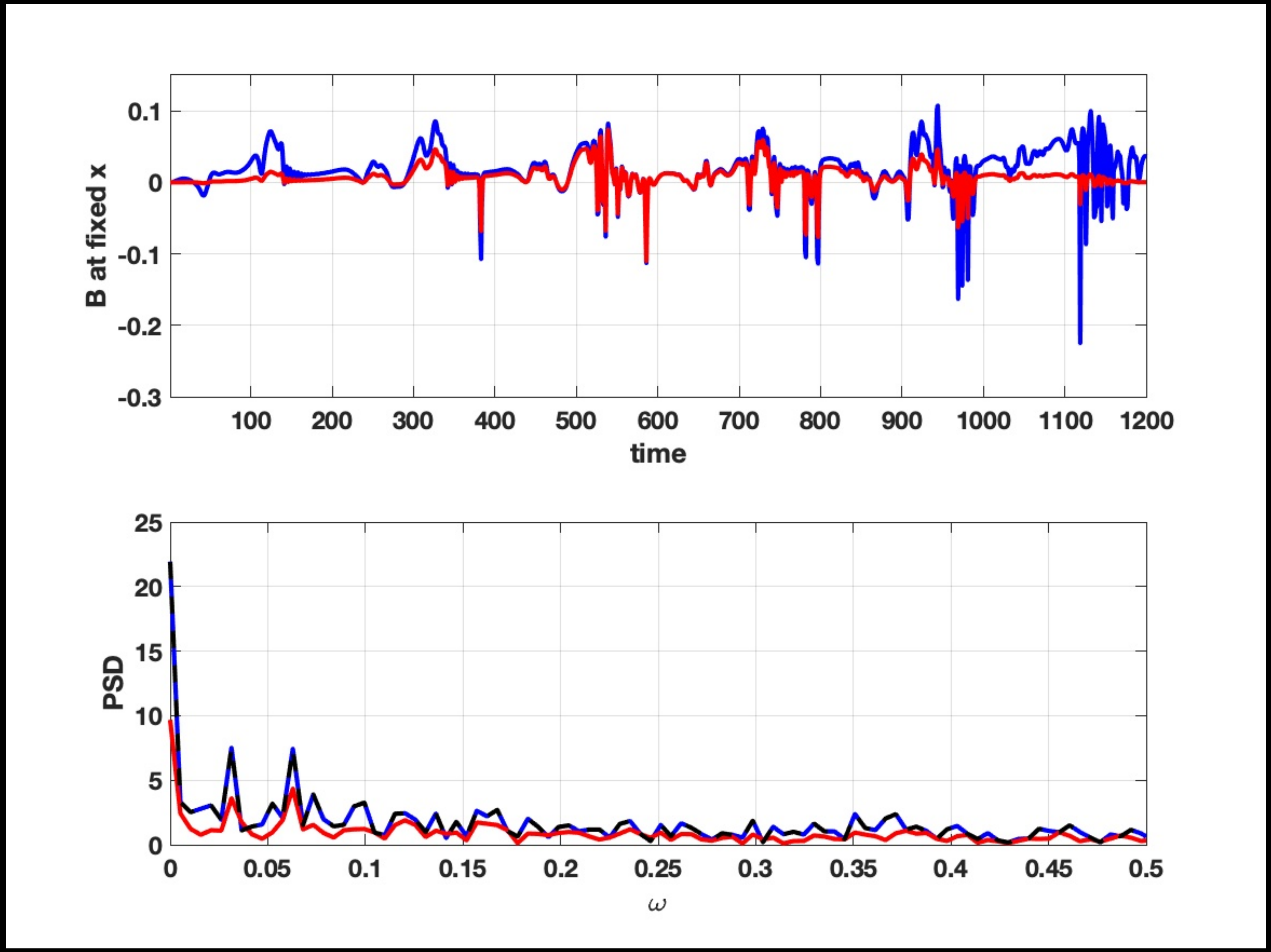
- A more general alternative from the engineering literature is to “window” the original data.
- Windowing means multiply by a function that is zero at the ends and one in the middle. Here I use a piecewise linear (triangle) function and the result is shown by the red curve.
- You can see that the spectrum has the same peaks as the blue spectrum in the lower panel.
- Windowing has a large engineering literature .



The windowed spectrum is a bit lower than the other two but retains the main peaks.

Variance reduction I

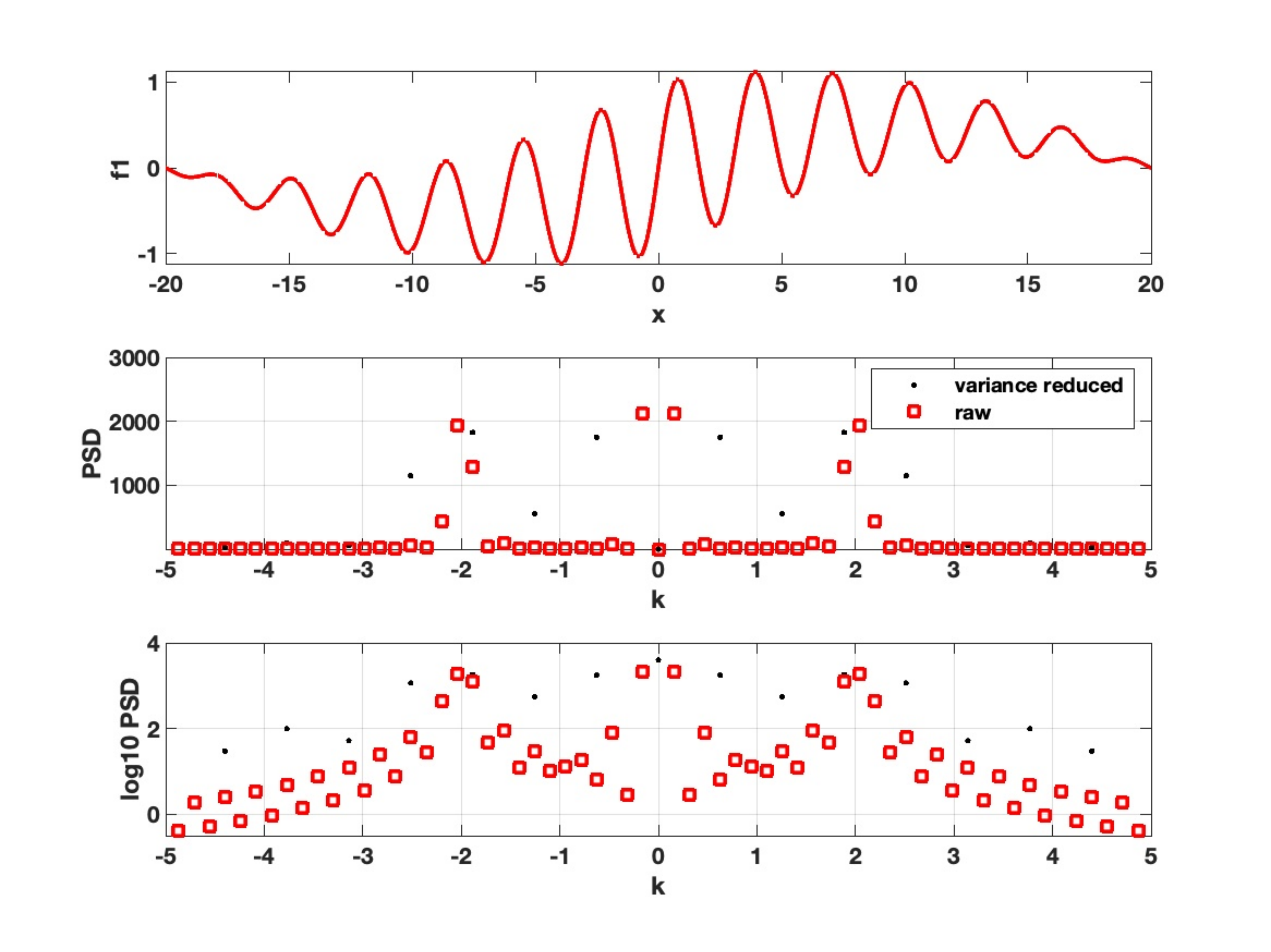
- We have noticed that spectra of data are quite choppy.
- Indeed theoretical results confirm that spectra can exhibit variance that is 100%.
- This sounds alarming, though in practice we also saw that when a frequency is truly dominant the spectrum clearly captures it.
- So what techniques have been developed for reducing variance?
- Well we start by noting that these apply to situations in which we have a signal with a lot of grid points.



The windowed spectrum is a bit lower than the other two but retains the main peaks.

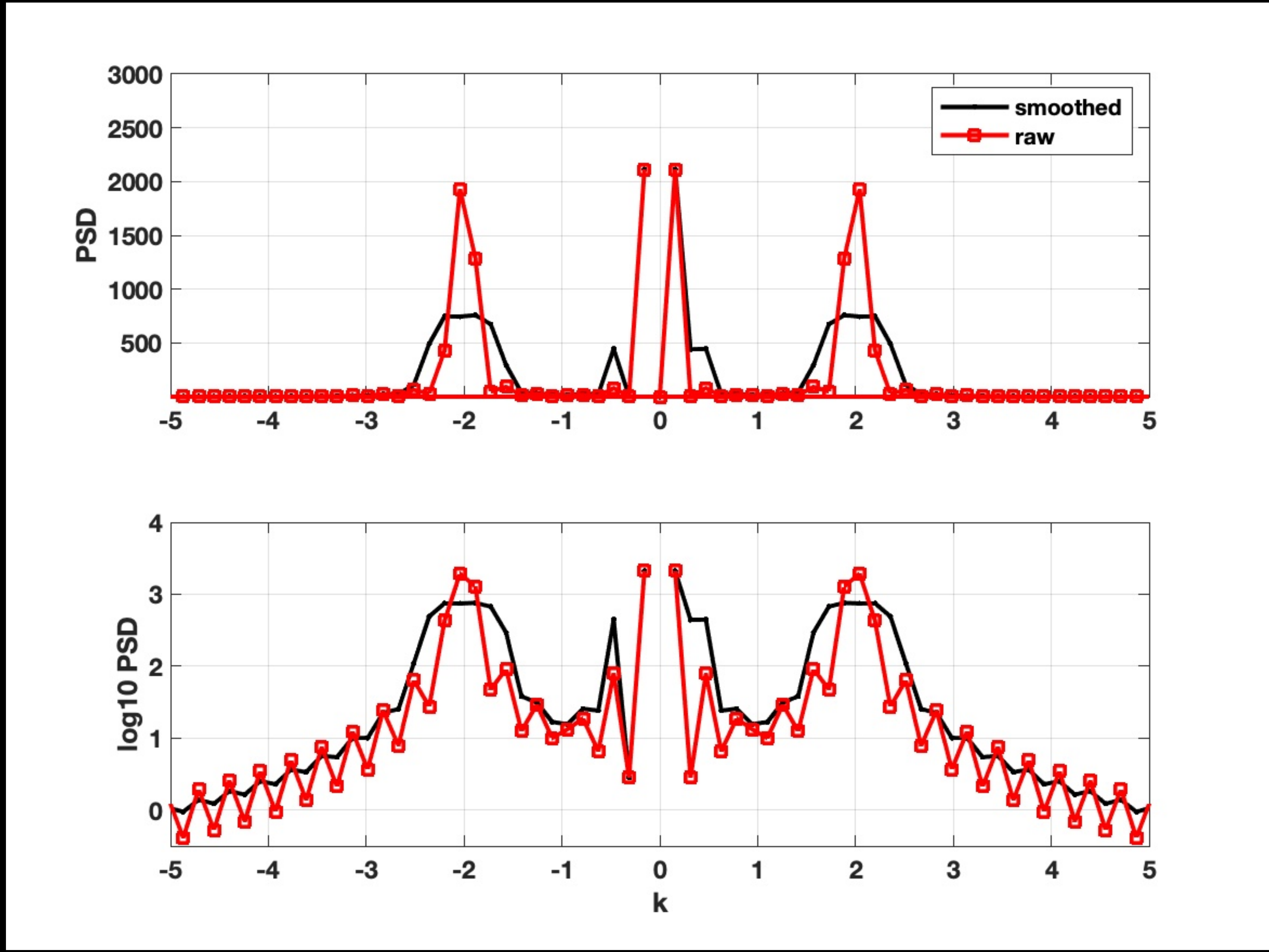
Variance reduction II

- We use a model function which excites a number of frequencies, a Jacobi elliptic function commonly used in studies of waves.
- The variance reduction strategy splits the signal into quarters, windows each section and computes the spectrum of this. It then sums these four spectra.
- You can see that this is OK in that it smooths the spectrum, but it also broadens



Variance reduction III

- Alternatively we could just compute the full spectrum and smooth it.
- In the picture on the right we use 5 point smoothing.
- You can see that this does broaden the peaks, but is quite effective at getting rid of the extra oscillations.
- In practice, the sophistication of your desired conclusions will drive how much manipulation of the spectrum you want to do.



Conclusions

- To really get a handle on what I have just shown you, you should go through the M-files (bbm_tide_specs.m and specexamps.m) to see how Matlab does it.
- If you prefer python then translate the code to python to see how the syntax changes, but the overall structure does not.
- Fourier analysis is very powerful, and ultimately very easy to use.
- It is not, however, omnipotent and is really at its strongest if it is coupled with some physical intuition for the data or model.