Problem 4: Now that we can generate fake correlated noise, let's use this in some actual fits. For this, we'll use a noise matrix that has two components: a diagonal term, plus correlated noise with a correlation length. In particular:

$$N_{ij} = a \exp(-\frac{(i-j)^2}{2\sigma^2}) + (1-a)\delta(i-j)$$

The variance of each point is always 1, but if a is very small, the data are nearly uncorrelated, while if a is very large (*i.e.* nearly one), they are almost perfectly correlated. If σ is large, the data points will be correlated over large distances, while if σ is small, they will only be correlated with their near neighbors.

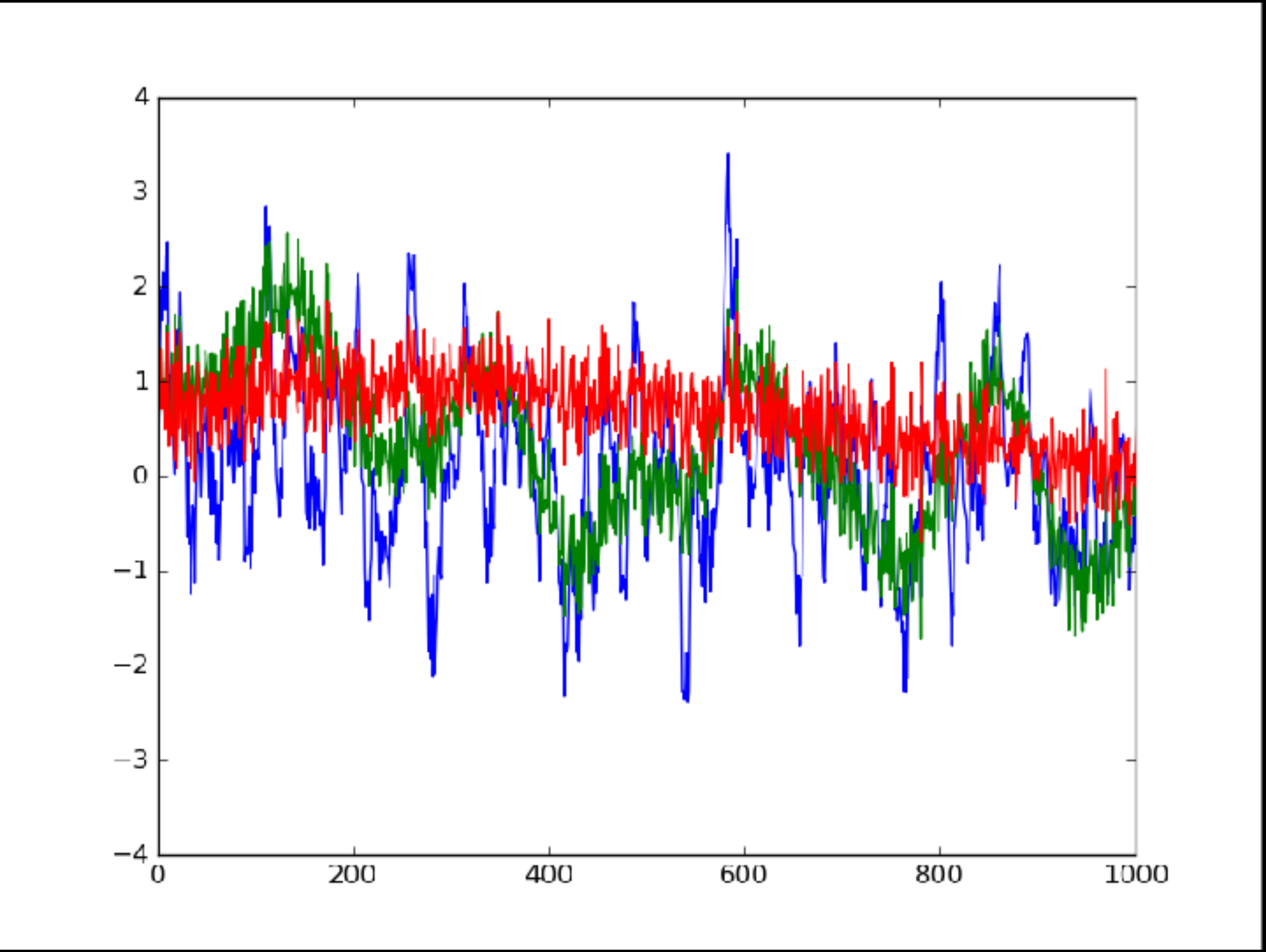
part a) Let's say we have 1000 points (so x goes from 0 to 999), and a Gaussian source with amplitude of 1 and σ_{src}=50 points in the center. Let's take the values of a to be (0.1,0.5,0.9) and the values of σ to be (5,50,500). Write a python script that generates the noise matrix for these values and uses it to report the error bar on the fit amplitude for each pair of a and σ. As a sanity check, I get that the error for $a = 0.5$ and correlation length $\sigma = 5$ samples is 0.276.

part b) Explain why the errors behave the way they do (comments in your code will suffice, no need to kill trees). Which set of parameters has the worst error bar? Which has the best? What sort of noise should you be most worried about? You might want to use your code from Problem 3 to generate noise realizations of these noise matrices, and plot the realizations with/without a signal added in.

How did errors behave?

- Why is short correlation length good?
- Why is long correlation length better?
- Why is intermediate correlation length worst?

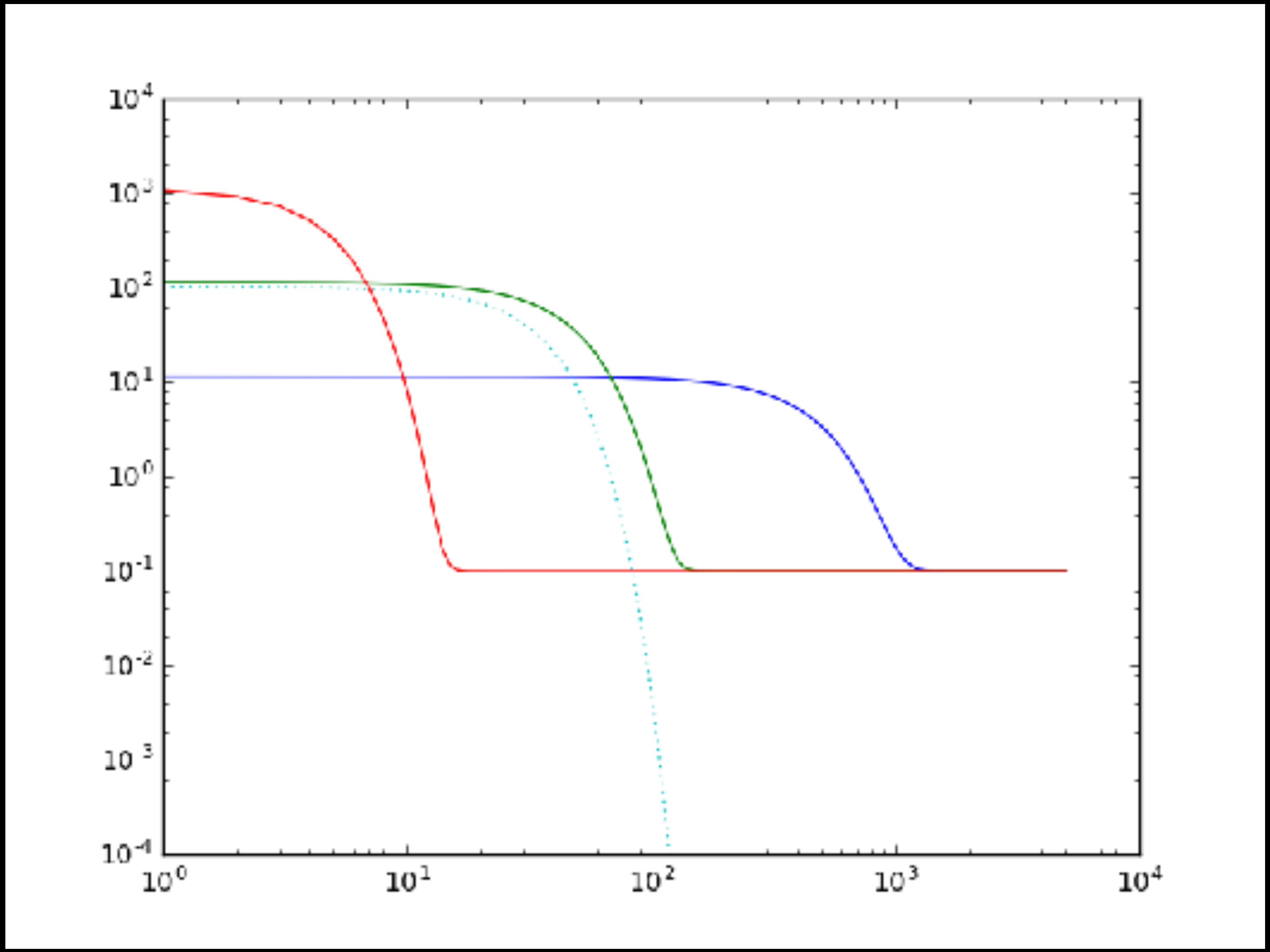
What the data look like



Power Spectrum Description

- Modes are uncorrelated in Fourier space
- SNR^2/mode is set by (template FT)^2/noise PS
- Noise PS is just FT of correlation function

PS of Noise Models



Red=? correlation length
Green=? Blue=?
Dotted = (scaled) template transform

Fourier Interpretation

- Noise model has same total variance independent of correlation length.

- Looking at FT, long length packs noise power into many long wavelengths. Template has more power on high-frequency scales (good SNR)

- Short length spreads out power over many many modes, dropping average noise power. Template well above noise on large scales (good SNR).

- Intermediate packs all its noise into same scales as template. Never have good SNR.

When your noise looks like your signal,
you're going to have a bad day…

Back to LIGO

- Reminder - if we have stationary noise, best fit amplitude for 1-D template is $A^T N^{-1} d / A^T N^{-1} A$
- We can search many possible locations of template with matched filter, replacing top by correlation of A with $N^{-1}d$ (or $N^{-1}A$ with d)
- Alternatively, could take correlation of $N^{-1/2}A$ with $N^{-1/2}d$.
- What would the noise in $N^{-1/2}d$ look like?

Back to LIGO

- Reminder - if we have stationary noise, best fit amplitude for 1-D template is $A^T N^{-1} d / A^T N^{-1} A$
- We can search many possible locations of template with matched filter, replacing top by correlation of A with $N^{-1}d$ (or $N^{-1}A$ with d)
- Alternatively, could take correlation of $N^{-1/2}A$ with $N^{-1/2}d$.
- What would the noise in $N^{-1/2}d$ look like?
    - Uncorrelated, with unit variance. This is called "prewhitening"

LIGO Data

- https://www.gw-openscience.org/tutorials/
- Download: “file with data” will get you everything
- simple_read_ligo.py will read for you (once you have h5py installed and working)
-

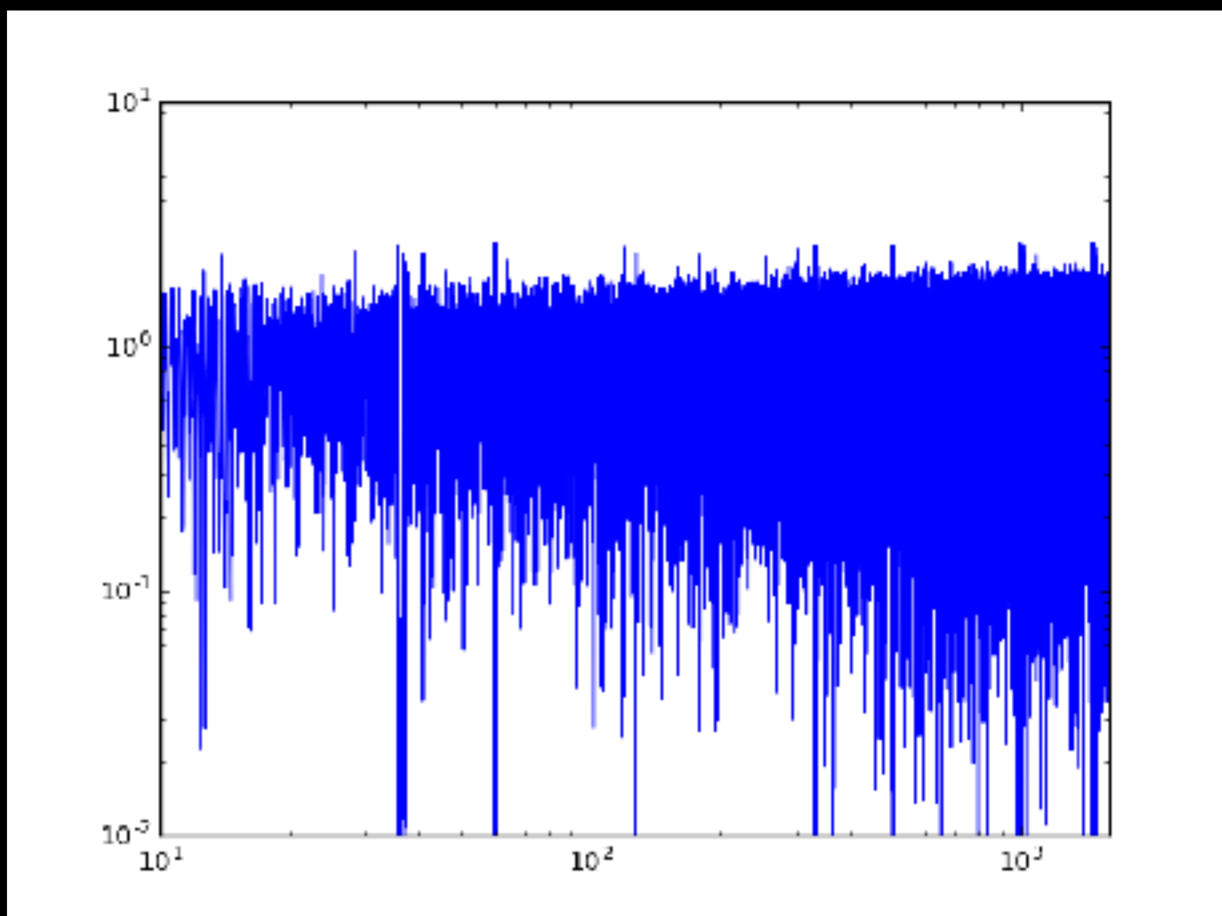
How Should We Estimate Noise?

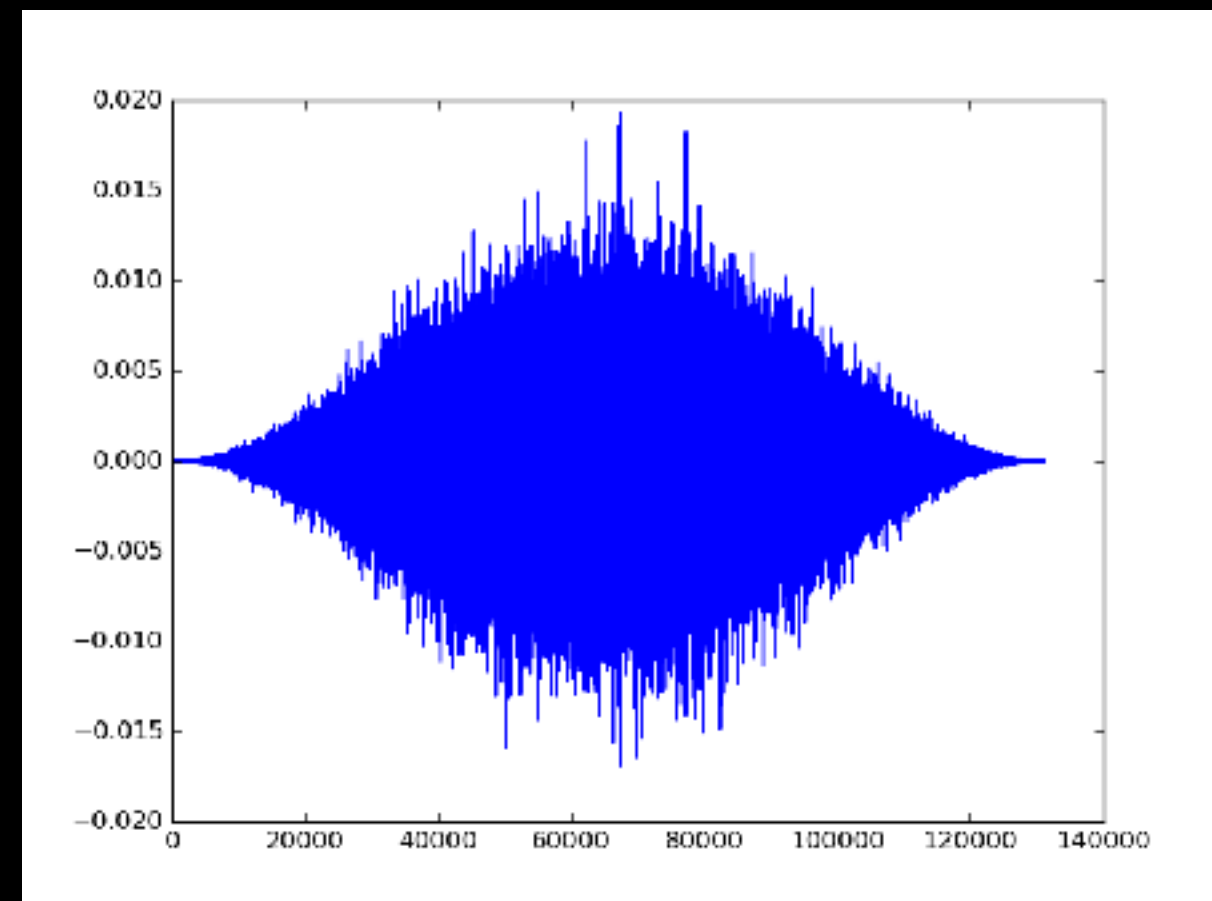
- Windowing key to avoiding FFT ringing
- smooths out spectral features
- Noise large per mode in FT, so we have to average
- What are your thoughts on averaging?

Smoothing PS

- Take $|FT|^2$, which is an estimate
- Smooth by convolving with an extended function.
- Thoughts on the function?

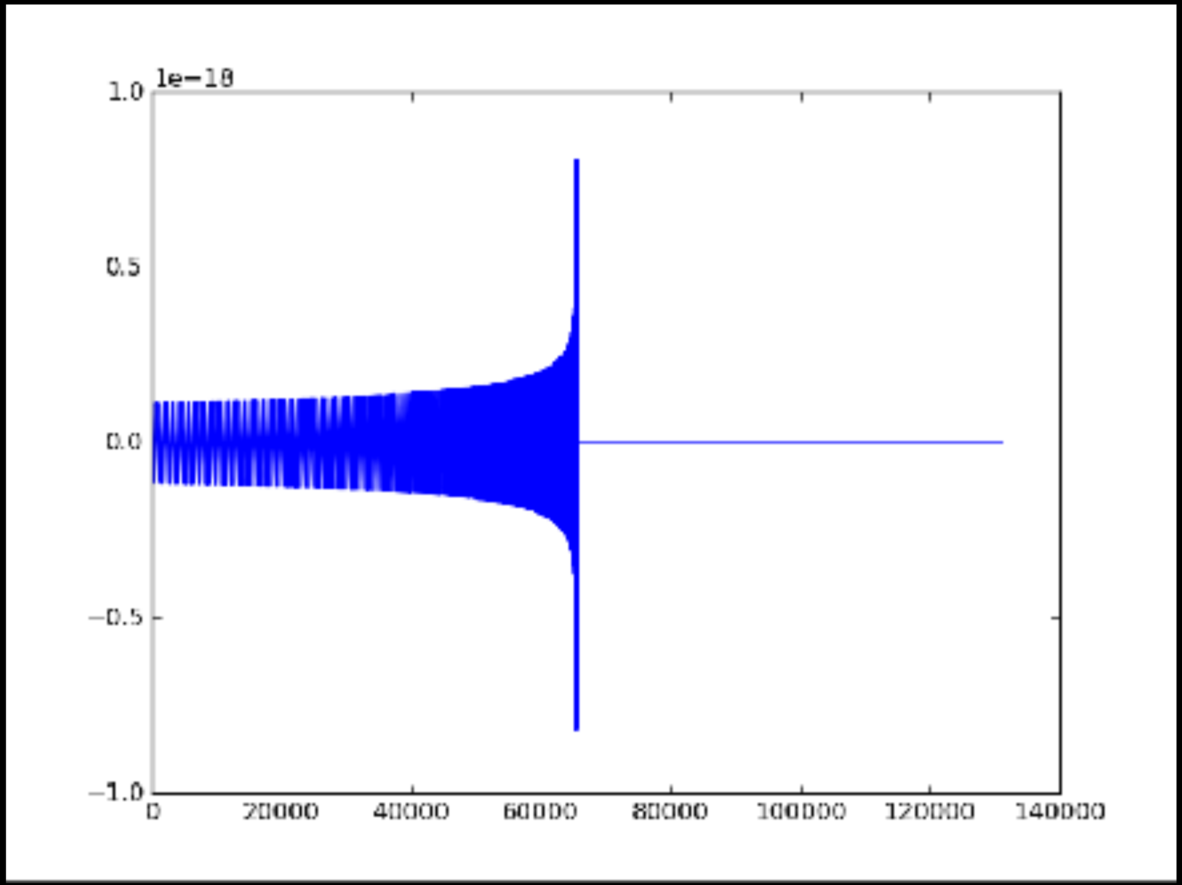
Pre-Whitened Data from Smoothed PS

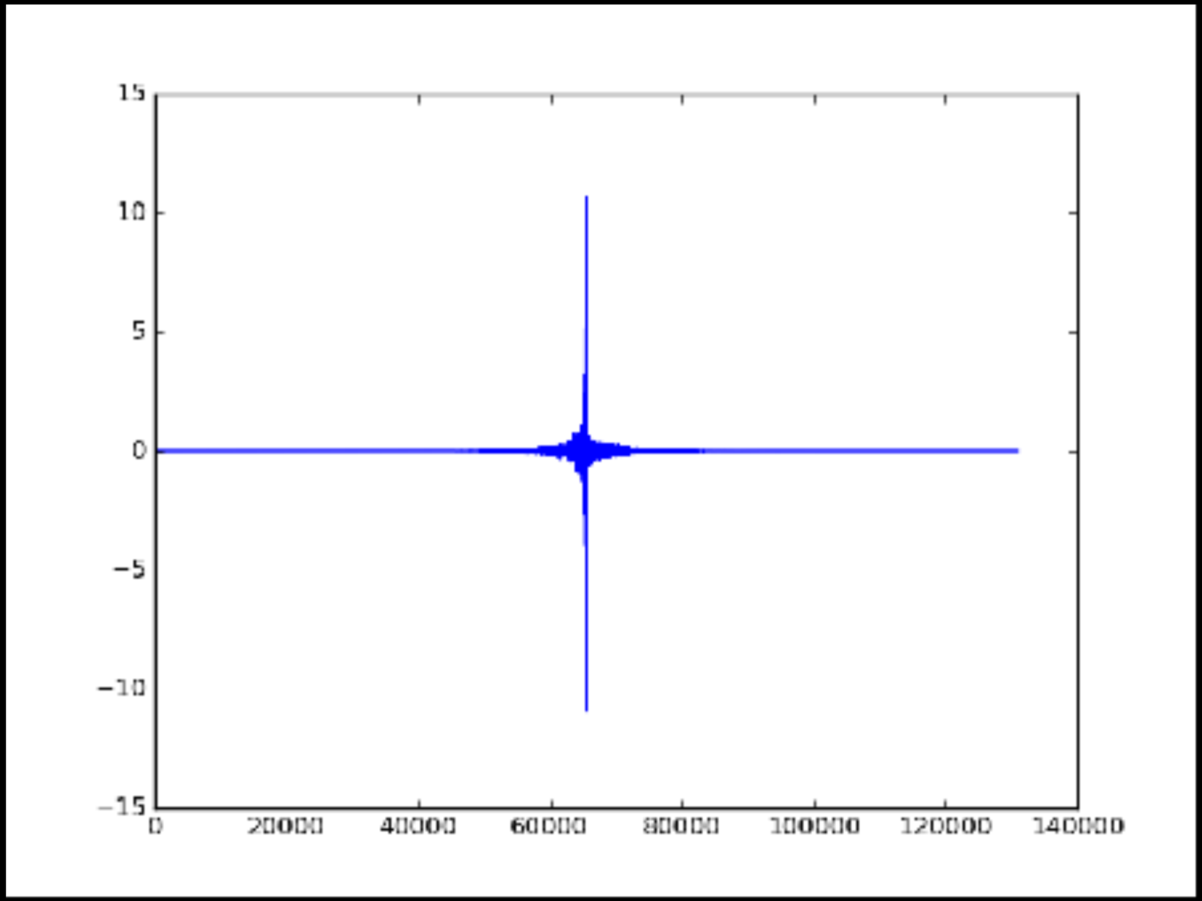




Left: Whitened FT of data. Looks not crazy. What are little nubbins sticking up?
Right: whitened data. Window shape is pretty obvious.

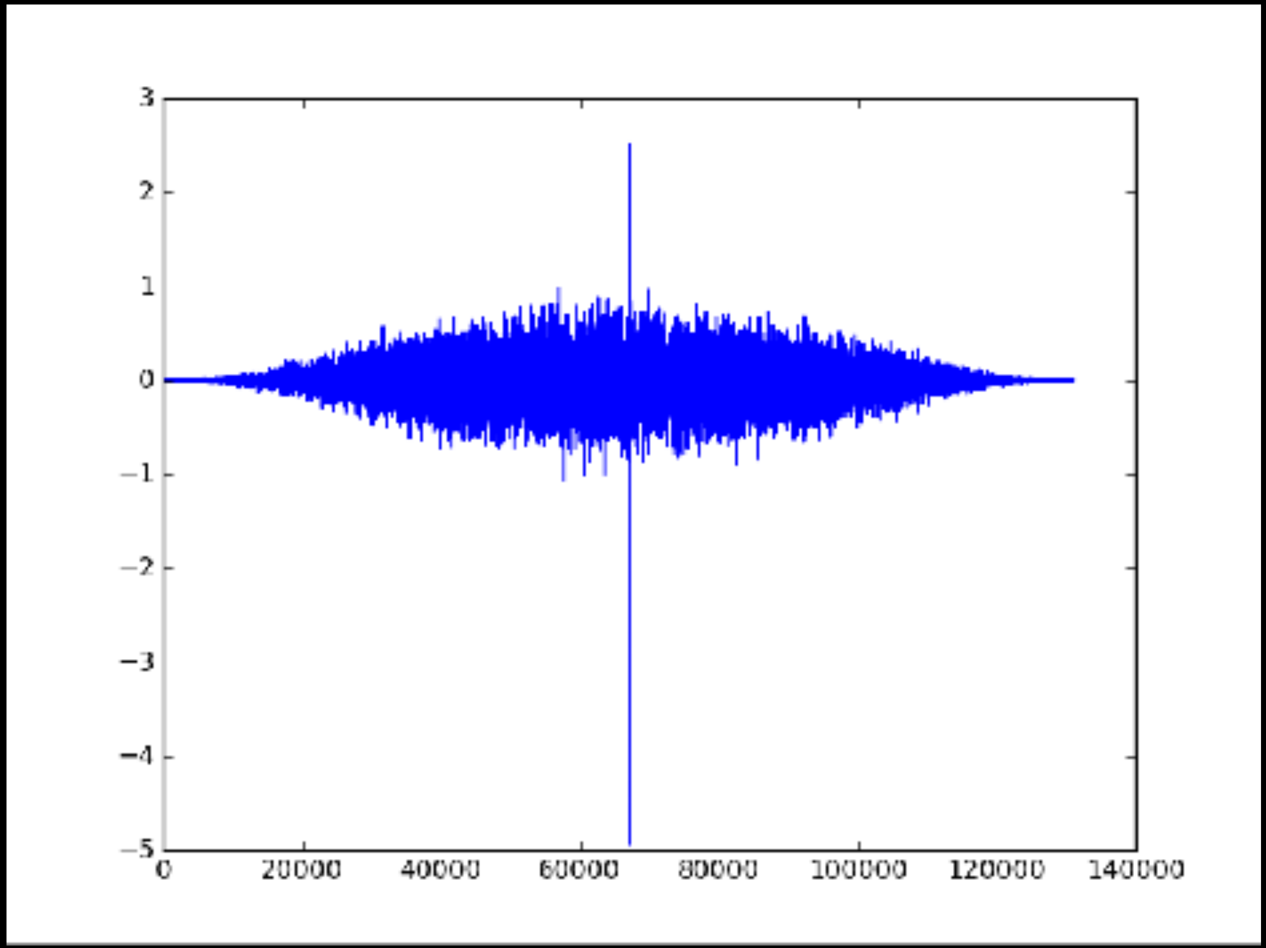
Pre-whitened template





- For this event, template is not small at start of data. Will this be a problem?
- Can look at pre-whitened version of template to get an idea.

Can Use for MF now



FFT Shift of matched filter output. We found a GW!

Averaging PS

- Break PS up into small chunks so we have many
- Take the FT of each chunk
- Add the FT^2s together.
- How do we apply this (short) PS to original data?
- Qualitatively, how do we relate this PS estimate to smoothed one?

More Windowing

- Usual windows taper every sample.
- Reduces power in a way we probably aren’t happy with
- How could we modify window to make this less of an issue?
- Let’s try this on data…