Phys 641 Lecture 4

Recap: Correlated Noise

- Correlated noise is common in wide variety of physical systems.
- Linear algebra formulation of χ^2 is accurate if $N_{ij}=\langle n_i n_j \rangle$ and noise is Gaussian
- Rotating into eigenvalues of noise matrix makes noise uncorrelated.

Stationary Noise

- For case of $N_{ij}=g(i-j)$, noise becomes uncorrelated in Fourier space, $<F^*_k F_{k'}>=G_k\delta(k-k')$
- If T is (normalized) Fourier transform operator, then we can write $\chi 2=r^T T^T N^{-1} T r$ where the residual r=(data-model) and N is now diagonal.
- As per problem set, this can still be extremely useful approximation even if noise isn't quite stationary.

Sample Problem

- We have seen error bars in LLS come from $(A^TN^{-1}A)^{-1}$. If condition number is bad then this may cause problems.
- Slightly more subtle - different columns of A could have different units, so condition number could change.
- Show that can rewrite A^TNA as BCB where C has ones along the diagonal and B is diagonal. What are the elements of B? What is inverse of BCB?
- In this case, C is often called the correlation matrix. Its condition number is usually more relevant to instability. Diagonal elements of C^{-1} tell you the error bar hit from correlated parameters.
- (came up on Friday cosmo-ph…)

χ^2 Example - template fitting

- Let's fit a 1-parameter template to data, but possibly want to shift it (e.g. fitting a for a source amplitude at various positions).

- If A is n by 1, then $A^T N^{-1} A$ is a scalar.

- now we have $m = A^T N^{-1} d / (A^T N^{-1} A)$.

- If N is stationary (i.e. $N_{ij} = f(i-j)$) and we shift the template $A_i \text{->} A_{i+\delta}$, how does the denominator depend on δ?

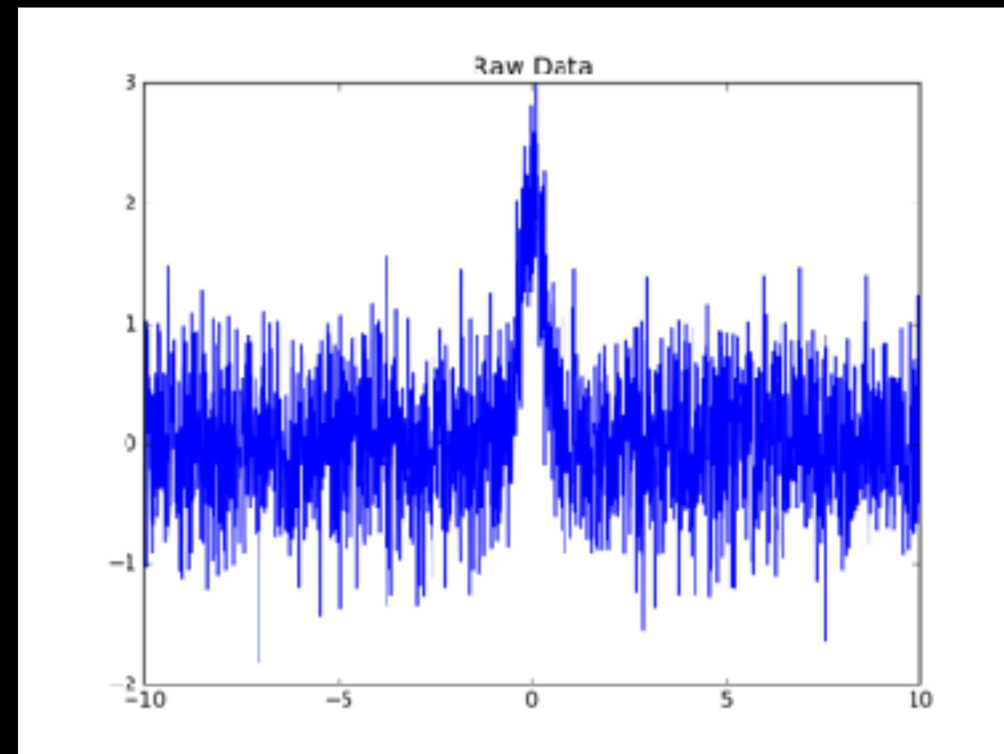
- Up to an overall constant, we can make $N^{-1}d$ and dot it against the various shifted A's.

Template fitting ctd.

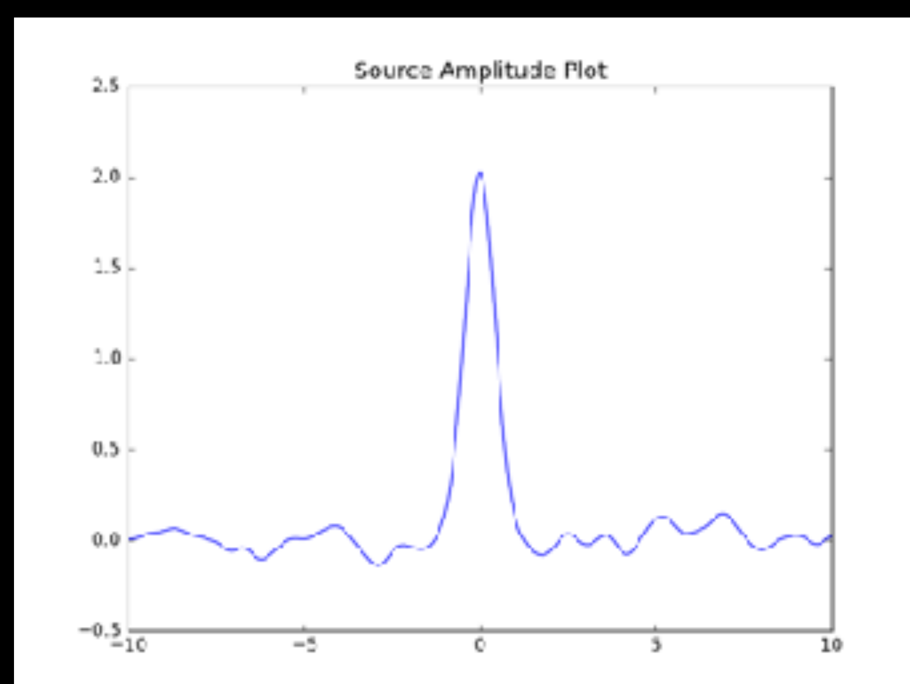
- What if we wanted to plot amplitude in terms of SNR?
- recall $Var(m)=(A^TN^{-1}A)^{-1}$, so $\sigma(m)=(A^TN^{-1}A)^{-1/2}$.
- So, $SNR=signal/noise=A^TN^{-1}d/A^TN^{-1}A/(A^TN^{-1}A)^{-1/2}=A^TN^{-1}d/\sqrt{(A^TN^{-1}A)}$.

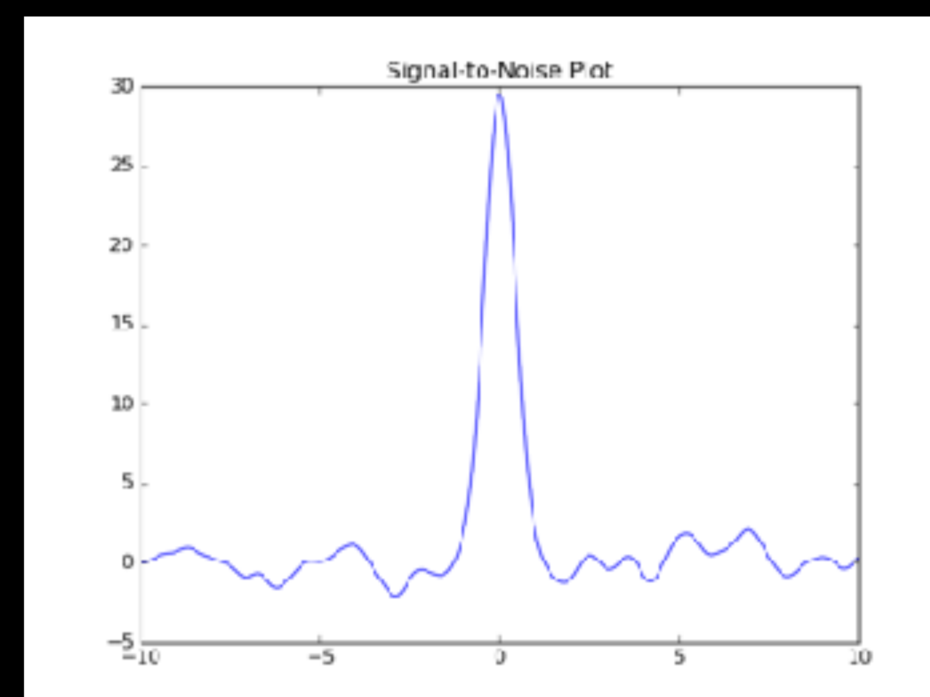
Code Example

```
import numpy
from matplotlib import pyplot as plt

dx=0.01
noise=0.5
Ninv=1.0/noise**2
x=numpy.arange(-10,10,dx)
n=len(x)


x0=0
amp_true=2.0

sig=0.3
template=numpy.exp(-0.5*(x-x0)**2/sig**2)

dat=template*amp_true+numpy.random.randn(n)*noise

snr=numpy.zeros(n)
amp=numpy.zeros(n)
dat_filt=Ninv*dat
denom=(numpy.dot(template,Ninv*template))
rt_denom=numpy.sqrt(denom)
for i in range(n):
    template=numpy.exp(-0.5*(x-x[i])**2/sig**2)
    rhs=numpy.dot(template,dat_filt)
    snr[i]=rhs/rt_denom
    amp[i]=rhs/denom
```




```
plt.clf();
plt.plot(x,snr);
plt.title('Signal-to-Noise Plot')
plt.savefig('snr_plot.png')

plt.clf();
plt.plot(x,amp)
plt.title('Source Amplitude Plot')
plt.savefig('amp_plot.png')

plt.clf();
plt.plot(x,dat)
plt.title('Raw Data')
plt.savefig('dat_raw.png')
```





But wait!

- We took $\sum d(t)a(t-\tau)$. But, this is just the correlation of d with a. We can do this quickly using Fourier transforms.

- Alternatively, let $a^{\lozenge}=a(-t)$. Then this is $\sum d(t)a^{\lozenge}(\tau-t)=d\otimes a^{\lozenge}$. By convolution theorem, this is $IFT(FT(d)FT(a^{\lozenge}))$.

- However, $FT=\sum f(x)\exp(-2\pi ikx/N)$. $FT(f(-x))=\sum f(-x)\exp(-2\pi ikx/N)=\sum f(x)\exp(2\pi ikx/N)=F^{*}(k)$. So, our output is just $IFT(FT(d)FT^{*}(a))$.

- NB - if a is symmetric, then A(k) is real (why?) and we can skip the conjugates.

How about changing noise?

- Often we're searching for things where noise is varying - say, searching for sources in a map that is deeper in the middle.
- Top: $A^T N^{-1} d$ is easy - correlate A with $N^{-1}d$.
- Bottom: $A^T N^{-1} A$ we can correlate N^{-1} with A^2.
- Amplitude stays ratio of the two (bottom is now a vector instead of a scalar), and SNR is top/√(bottom)

Matched Filter

- We are beginning on a very widespread class of techniques called *matched filters*.
- In various forms, they show up in GW analysis, photometry in optical images, finding galaxy clusters in CMB maps, finding radar return echos…
- Extendable to non-independent noise, multiple (possibly correlated) simultaneous/multi-frequency datasets, many others.
- All arises from writing down χ^2 and minimizing. Make a habit of this.

Fourier Transforms

- On computer, discrete Fourier transform usually defined as $\sum f(x)\exp(-2\pi ikx/N)$, sum goes from 0 to N-1
- Can write this as a matrix multiply where ij^{th} element is $\exp(-2\pi ikx/N)$
- What is i^{th} column dotted with j^{th} column?
- what is the inverse of this matrix?
- Strongly encourage you to just memorize this. Numerical factors important in real life…

Some Fourier Theorems

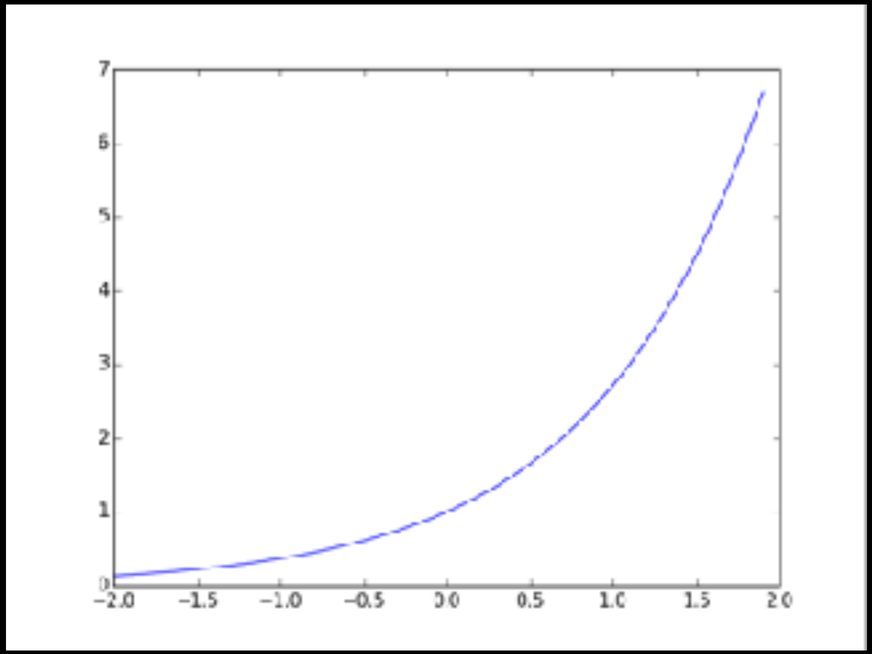
- $x \to x+\delta$, $F(k) \to F(k)\exp(-2\pi i\delta k/N)$
- $x \to -x$, $F(k) \to F^*(k)$
- Convolution theorem: $FT(f \otimes g) = FT(f)^* FT(g)$ where $f \otimes g = \int f(\tau) g(t-\tau) d\tau$.

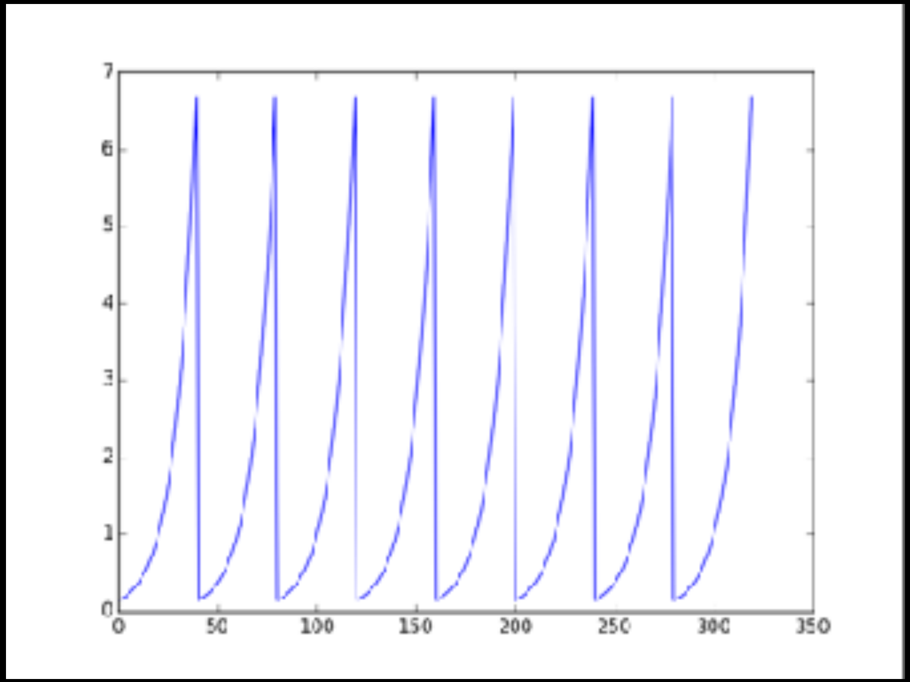
Periodicity

- $f(x)=\sum F(k) \exp(-2\pi ikx/N)$
- What is $f(x+N)$? $\sum F(k) \exp(-2\pi ik(x+N)/N)$
- $=\sum F(k) \exp(-2\pi ik)\exp(-2\pi ikx/N)$.
- $\exp(-2\pi ik)=1$ for integer k, so $f(x+N)==f(x)$
- DFT's are periodic - they just repeat themselves ad infinitum.

Periodicity

```
import numpy
from matplotlib import pyplot as plt

x=numpy.arange(-2,2,0.1)
y=numpy.exp(x)
plt.plot(x,y)
plt.savefig('fft_exp')
plt.show()
yy=numpy.concatenate((y,y))
yy=numpy.concatenate((yy,yy))
yy=numpy.concatenate((yy,yy))
plt.plot(yy)
plt.savefig('fft_exp_repeating')
plt.show()
```





- You may think you're taking top transform. You're not - you're taking the bottom one.

- In particular, jumps from right edge to left will strongly affect DFT

Aliasing

- $f(x)=\sum F(k)\ \exp(2\pi ikx/N)$
- What if I had higher frequency, k>N? let $k^*=k-N$ (i.e. k^* low freq.)
- $f(x)=\sum F(k)\ \exp(2\pi i(k^*+N)x/N=\sum F(k)\exp(2\pi ix)\exp(2\pi ik^*x/N)$
- for x integer, middle term goes away: $\sum F(k^*+N)\exp(2\pi ik^*x/N)$
- High frequencies behave exactly like low frequencies - power has been *aliased* into main frequencies of DFT.
- Always keep this in mind! Make sure samples are fine enough to prevent aliasing.

Negative Frequencies

- All frequencies that are N apart behave identically
- DFT has frequencies up to (N-1).
- Frequency (N-1) equivalent to frequency (-1). You will do better to think of DFT as giving frequencies (-N/2,N/2) than frequencies (0,N-1)
- *Sampling (Nyquist) theorem*: if function is band-limited - highest frequency is ν - then I get full information if I sample *twice* per frequency, $dt=1/(2\nu)$. Factor of 2 comes from aliasing.

Exercise

- What is the PDF of the sum of two independent random variables?

- Let's now add n independent identically distributed random variables together.

- Without (much) loss of generality can rescale so mean is 0 and variance is 1.

Continue…

- If I want to rescale sum so mean stays zero and variance stays 1, what factor should I rescale by?
- If $F(k)$ is FT of probability distribution, what if $F(0)$? What is $F(\delta)$ like if $\delta << 1/\sigma$?
- To leading order, $F(k)=1-ak^2+\ldots$
- What happens to $F(k)$ if $x_i \rightarrow x_i/a$?
- Leaves us with $\sum x_i/\mathrm{sqrt}(n)$ has FT $(1-ak^2/n)n$ as n goes to infinity. This turns into $\exp(-ak^2)$, so FT of distribution of sum is a Gaussian.
- If that's true, FT of a Gaussian is a Gaussian, so distribution of sum must also be a Gaussian. ergo central limit theorem.

Comin up..

- Getting very close to running on actual data.
- LIGO tutorial very handy - will be using things from https://losc.ligo.org/tutorials/
- Think about how to estimate power spectra…