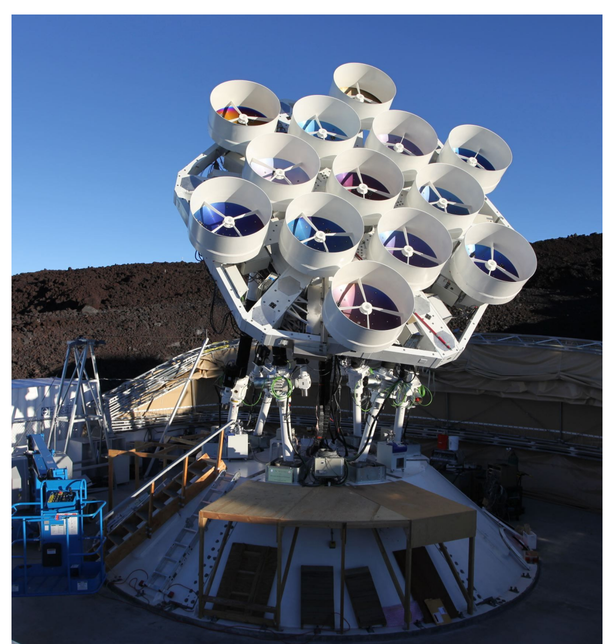
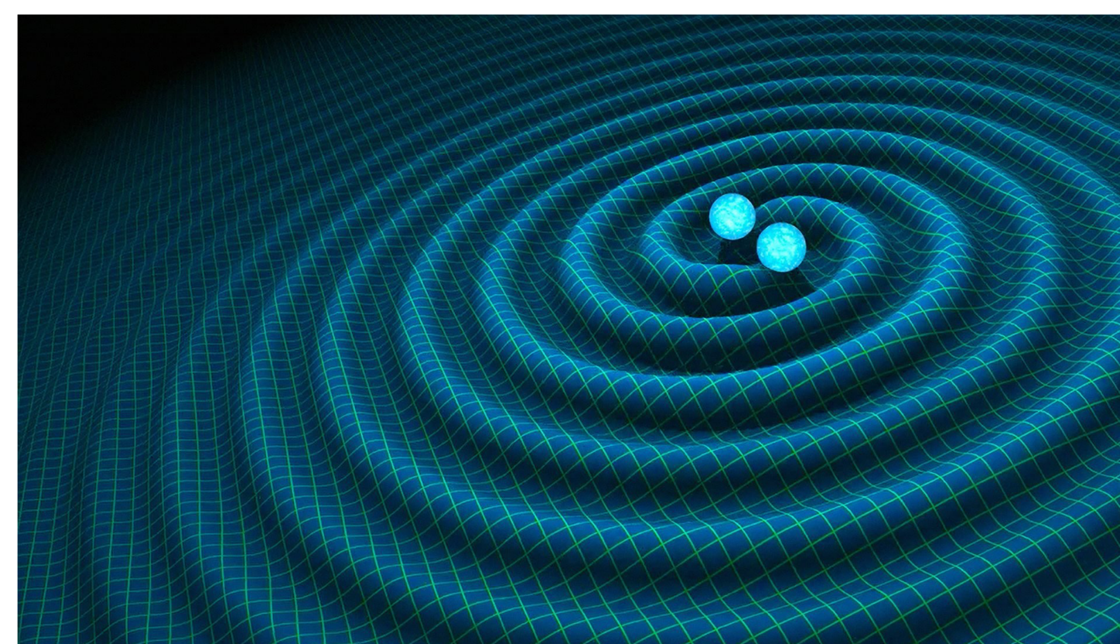
Detection of gravitational waves and data pipeline for compact binary coalescences and stochastic gravitational wave background

Guo Chin Liu
(Tamkang University)

2019 APCTP School on Gravitational-Wave Cosmology

Data analysis groups in Taiwan

- Feng-Li Lin's group @ National Taiwan Normal University
  - CBC Search pipeline with GPU
  - CBC search/parameter estimation with deep learning
  - Parameter estimation
- Cosmology group @ Tamkang University
  - Theoretical studies on boson star
  - Data analysis of stochastic gravitational wave background (SGWB)

Sources of LIGO-VIRGO-KAGRA

- Compact Binary Coalescence (CBC)
  - BNS, BBH, BH-NS..
- Burst: poorly modeled signal
  - Core-collapse supernovae
  - Long Gamma Ray Burst
  - Cosmic string
- Continuous Waves: long lived, nearly sinusoidal and weak
  - Rotating neutron stars
- Stochastic Background
  - Cosmology: inflation, cosmic string
  - Astrophysics: unresolved CBC, supernovae, and neutron stars

Search Methods

- CBC: matched filter
- Burst: identify coherent excess power in the data from multiple detectors
- CW: long time coherent matched filtering
- SGWB: cross-correlation

Packages

- Python (including libraries such as numpy, scipy, matplotlib)
- PyCBC
- lalsuite
- Healpix

Outlines

- Basics of Fourier Transformation
- Data analysis of SGWB
- Search of CBC

Basics of Fourier Transformation

Random processes

- A sequence of random variable
    - Instrument noise, SGWB
- Ensemble average $\langle x \rangle = \int x p(x) dx$ p(x): probability density function
- Stationary random process: statistical properties do not change with time

$$\langle x \rangle = \lim_{T \to \infty} \frac{1}{T} \int_{-T/2}^{T/2} x(t) dt$$

Without loss of generality, we can set

$$\langle x \rangle = 0$$

Fourier Transformation

For a random processed data x(t), Fourier transformation of x(t) is defined as

$$\tilde{x}(f) = \int_{-\infty}^{\infty} x(t) e^{-i2\pi ft}\, dt$$

And the inverse Fourier transformation is defined as

$$x(t) = \int_{-\infty}^{\infty} \tilde{x}(f) e^{i2\pi ft}\, df$$

Sign convention:
Same as FFTW, Numpy, but different with numerical recipes

If x(t) is real,

$$\tilde{x}(-f) = \tilde{x}^*(f)$$

Auto-correlation of x(t)

Auto-correlation of stationary x(t) is defined as

$$R(\tau) \equiv \left\langle x(t)x(t+\tau)\right\rangle = \lim_{T\to\infty}\frac{1}{T}\int_{-T/2}^{T/2} x(t')x(t'+\tau)dt'$$

<>: take average

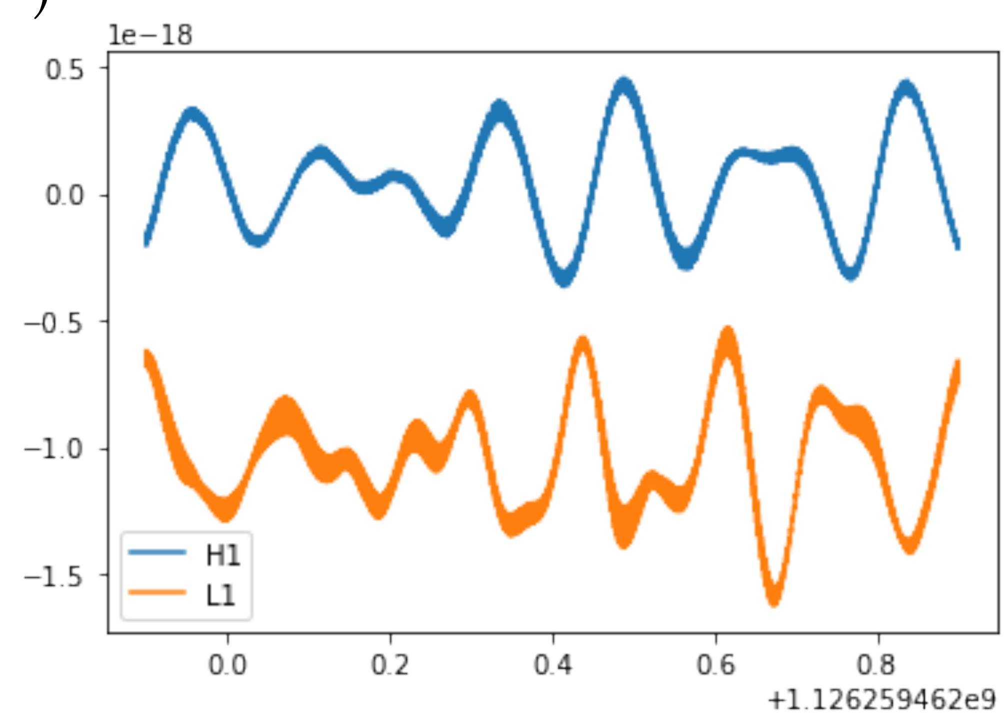
Stationary: the statistics depends only on time difference but not on the choice of time origin

$$R(\tau) = \lim_{T\to\infty}\frac{1}{T}\int_{-T/2}^{T/2} x(t')x(t'+\tau)dt' = \lim_{T\to\infty}\frac{1}{T}\int_{-T/2}^{T/2}\int_{-\infty}^{\infty}\tilde{x}^*(f)\tilde{x}(f')e^{-i2\pi ft'}e^{i2\pi f'(t'+\tau)}\,df\,df'\,dt'$$

$$= \lim_{T\to\infty}\frac{1}{T}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\tilde{x}^*(f)\tilde{x}(f')e^{i2\pi f'\tau}\,df\,df'\,\delta(f-f')$$

$$= \int df\left[\lim_{T\to\infty}\frac{1}{T}\tilde{x}^*(f)\tilde{x}(f)\right]e^{i2\pi f\tau}$$

White noise describes a random process whose mean is zero and whose autocorrelation is a delta-function



Power Spectrum Density

Define Power Spectrum Density

$$P(f) \equiv \lim_{T\to\infty} \frac{1}{T} \left| \tilde{x}(f) \right|^2$$

We have

$$R(\tau) = \int_{-\infty}^{\infty} df P(f) e^{i2\pi f\tau} \qquad P(f) = \int_{-\infty}^{\infty} d\tau R(\tau) e^{-i2\pi f\tau}$$

Note:

1. **The auto-correlation function is the FT of the power spectrum density**
2. **The auto-correlation function is real** $R(\tau) = R(-\tau)$
3. $P(-f) = P(f)$
4. **One-sided power spectrum density:**

$$\begin{aligned} S(f) &= P_{\text{one}}(f) = 2P(f) \quad & f \geq 0 \\ &= 0 & f < 0 \end{aligned}$$

$$R(0) = \left\langle x^2 \right\rangle = \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} x^2(t)\, dt = \int_0^{\infty} S(f)\, df$$

Stationary

$$\left\langle \tilde{x}^*(f')\tilde{x}(f)\right\rangle = \left\langle \int_{-\infty}^{\infty} x(t')e^{i2\pi f't'}dt' \int_{-\infty}^{\infty} x(t)e^{-i2\pi ft}\, dt \right\rangle$$

Replace t by $t = t' + \tau$

$$\left\langle \tilde{x}^*(f')\tilde{x}(f)\right\rangle = \left\langle \int_{-\infty}^{\infty} x(t')e^{i2\pi f't'}dt' \int_{-\infty}^{\infty} x(t'+\tau)e^{-i2\pi f(t'+\tau)}\, d\tau \right\rangle$$

$$= \int_{-\infty}^{\infty} e^{-i2\pi(f-f')t'}dt' \int_{-\infty}^{\infty} \left\langle x(t')x(t'+\tau)\right\rangle e^{-i2\pi f\tau}\, d\tau = P(f)\delta(f-f')$$

Discrete Fourier Transform

The real data can not be infinite and continuous

$$\tilde{x}(f) = \int_{-\infty}^{\infty} x(t) e^{-i2\pi ft}\, dt \qquad \longrightarrow \qquad \tilde{x}_k = \Delta t \sum_{j=0}^{N-1} x_j e^{-i2\pi t_j f_k} \qquad x_j = x(t_j)$$

$$x(t) = \int_{-\infty}^{\infty} \tilde{x}(f) e^{i2\pi ft}\, df \qquad \qquad x_j = \Delta f \sum_{k=0}^{N-1} \tilde{x}_k e^{i2\pi t_j f_k} \qquad \tilde{x}_k = \tilde{x}(f_k)$$

We sample the time and frequency uniformly

$$t_j = j\Delta t = j\frac{T}{N}$$

$$f_k = k\Delta f = k\frac{1}{T}$$

Then

$$t_j f_k = j\frac{T}{N} k\frac{1}{T} = \frac{jk}{N}$$

Discrete Fourier Transform

$$\tilde{x}_k = \Delta t \sum_{j=0}^{N-1} x_j e^{-i2\pi jk/N}$$

$$x_j = \Delta f \sum_{k=0}^{N-1} \tilde{x}_k e^{i2\pi jk/N}$$

What numerical libraries do for us

Discrete Fourier Transform

Noise power spectrum

$$\left\langle \tilde{n}^*(f')\tilde{n}(f)\right\rangle = \frac{1}{2}\delta(f-f')S_n(|f|)$$

In DFT

$$\left\langle \tilde{n}_i^*\tilde{n}_j\right\rangle = \frac{1}{2}D_{ij}S_n(|f|)$$

Normalization:

$$\int \delta(f-f')df = 1 \rightarrow \sum_i D_{ii}\Delta f = 1$$

Thus, we have

$$D_{ij} = 0 \quad i \neq j$$

$$D_{ij} = \frac{1}{\Delta f} = T \quad i = j$$

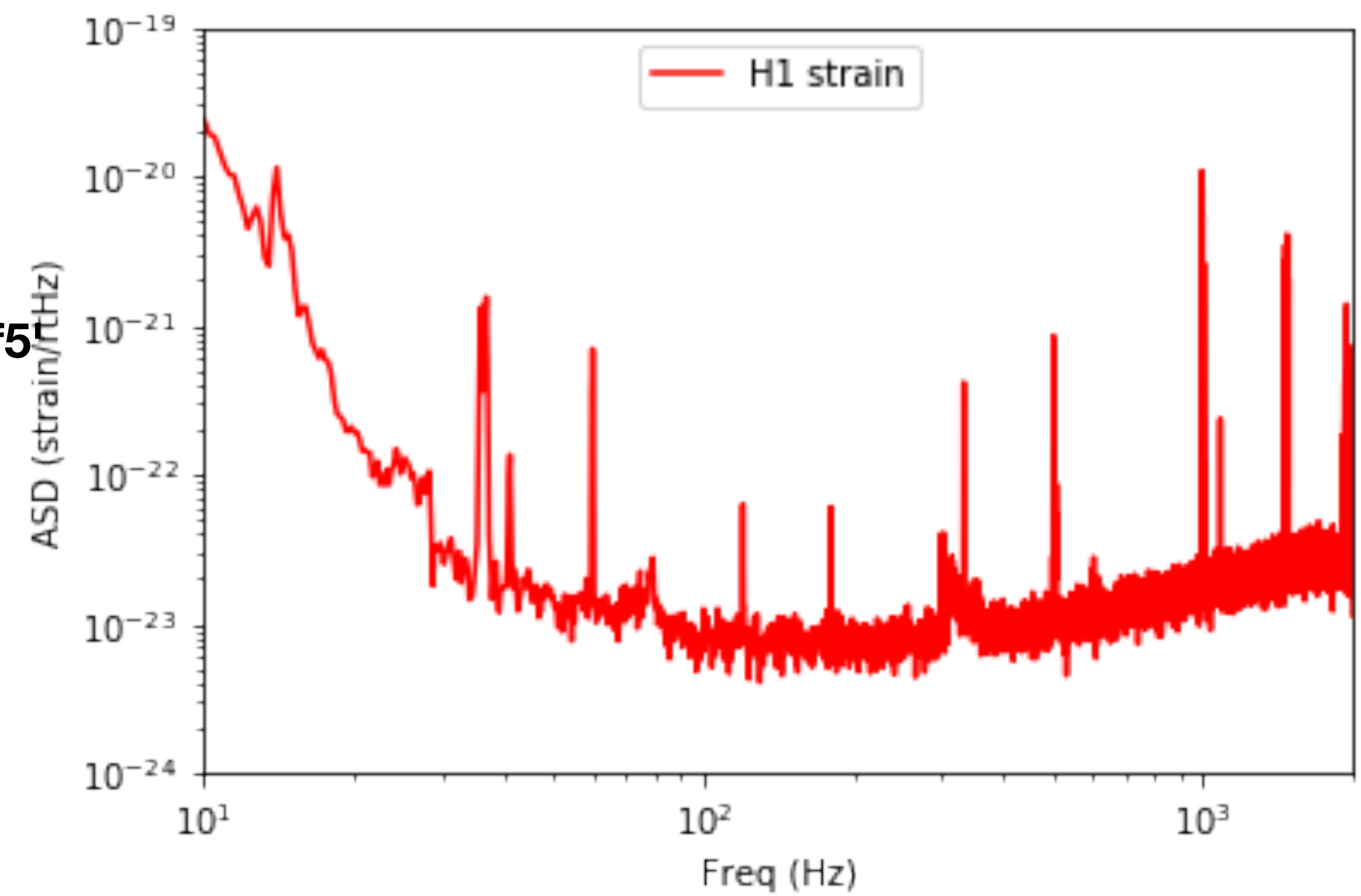
$$\left\langle \tilde{n}_i^*\tilde{n}_j\right\rangle = \frac{1}{2\Delta f}S_n(|f_i|)\delta_{ij} = \frac{T}{2}S_n(|f_i|)\delta_{ij}$$

Variance

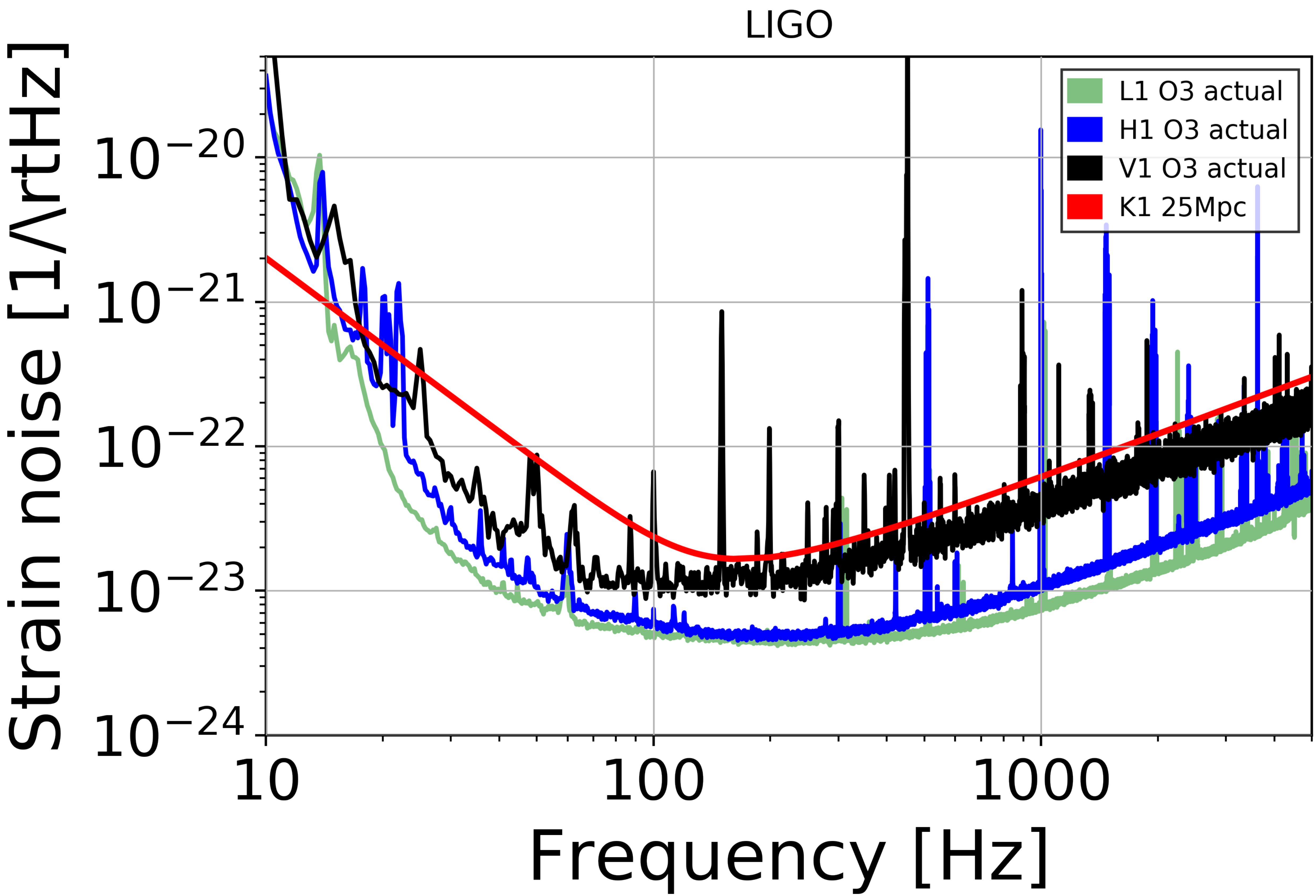
$$\sigma_i^2 = \sigma^2(f_i) = \left\langle |n_i^R|^2\right\rangle = \left\langle |n_i^I|^2\right\rangle = \frac{1}{4\Delta f}S_n(f_i)$$

Sensitivity Curve

```
import numpy as np
import readligo as rl
import matplotlib.pyplot as plt
import h5py
import matplotlib.mlab as mlab
fn_H1 = 'H-H1_LOSC_4_V1-1126259446-32.hdf5'
strain_H1, time_H1, chan_dict_H1 =
rl.loaddata(fn_H1, 'H1')
fs = 4096
NFFT = 4*fs
noverlap = 0
PSD_H1, freqs = mlab.psd(strain_H1, Fs = fs,
NFFT = NFFT, noverlap=noverlap)
```

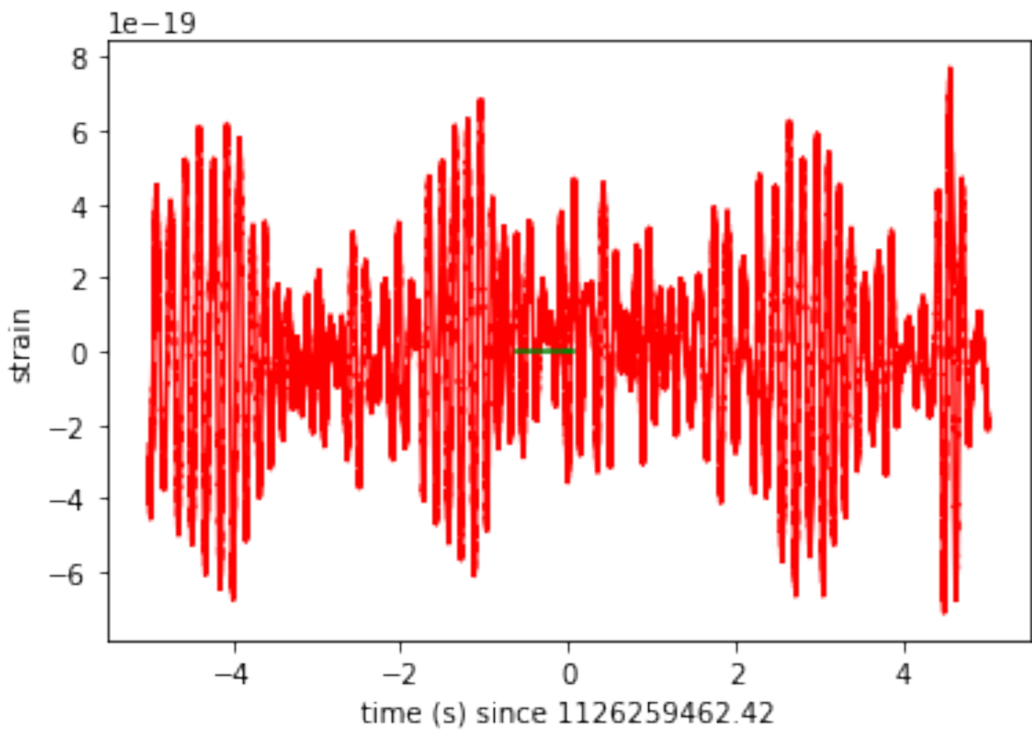


O3 Sensitivity Curves

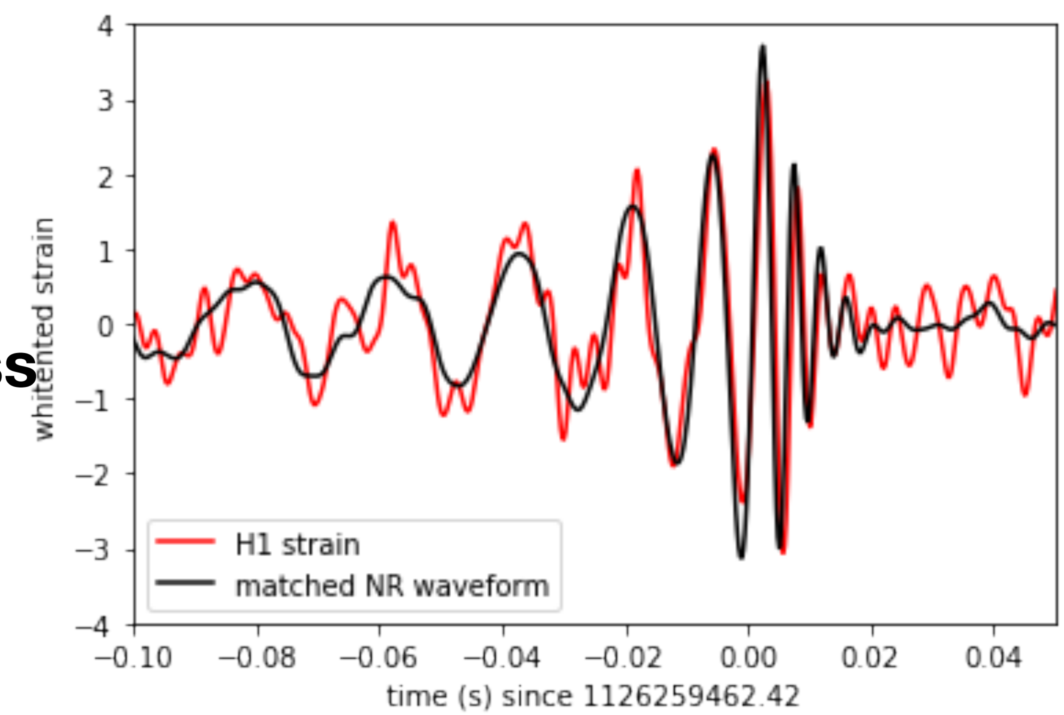


Stochastic Signals

Stochastic signal is randomly produced by a lot of number of weak, unresolved sources



After process



What if magnitude of signal is much smaller than this?

Search of SGWB

Cross-Correlation of data in two detectors

- Early papers of using cross-correlation to search SGWB: Grishchuk 1976; Helling and Downs 1983
- Recent papers:
  - Mitra et al., Gravitational wave radiometry: Mapping a stochastic gravitational wave background (2008)
  - Cornish & Romano, Towards a unified treatment of gravitational-wave data analysis (2013)
  - Thrane et al. Probing the anisotropies of a stochastic gravitational-wave background using a network of ground-based laser interferometers (2009)
  - Ain, Suresh & Mitra, Very fast stochastic gravitational wave background map-making using folded data(2018))

Cross-Correlation of data in two detectors

Consider we have to detectors:

$$d_1 = h_1 + n_1$$

$$d_2 = h_2 + n_2$$

Expected value of the cross-correlation

$$\langle C_{12} \rangle = \langle d_1 d_2 \rangle = \langle h_1 h_2 \rangle + \langle h_1 n_2 \rangle + \langle n_1 h_2 \rangle + \langle n_1 n_2 \rangle = \langle h_1 h_2 \rangle$$

GW Radiation

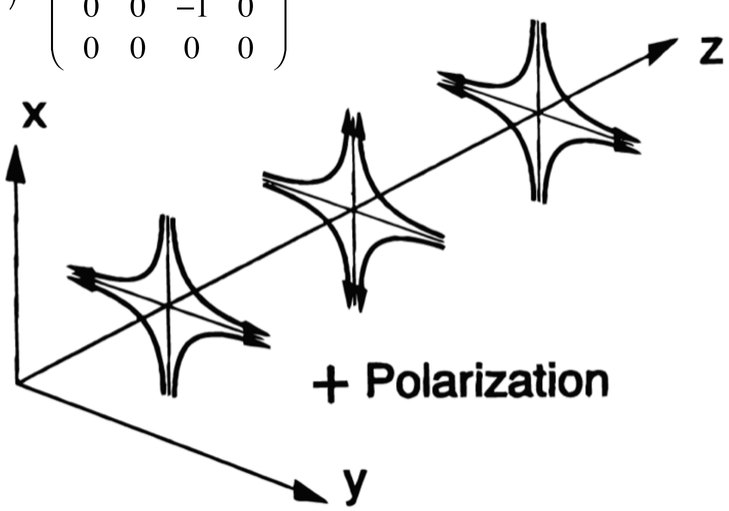
Metric perturbation by GW can be written as superposition of plan waves

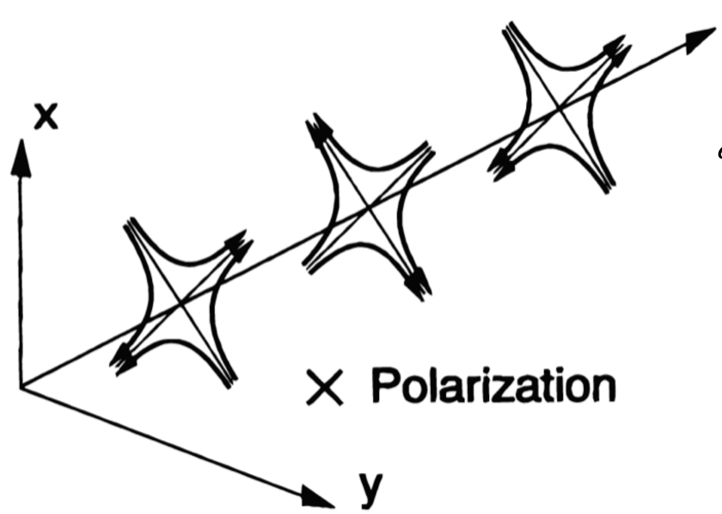
$$h_{ab}\left(t,\vec{x}\right)=\int_{-\infty}^{\infty}df\int_{S^2}d^2\hat{n}\,h_{ab}(f,\hat{n})e^{i2\pi f\left(t+\hat{n}\bullet\vec{x}/c\right)}$$

Expand the GW in the standard + and x polarization

$$h_{ab}\left(f,\hat{n}\right)=h_+e^{+}_{ab}\left(\hat{n}\right)+h_\times e^{\times}_{ab}\left(\hat{n}\right)$$

$$e^{+}_{ab}\left(\hat{n}\right)=\begin{pmatrix}0&0&0&0\\0&1&0&0\\0&0&-1&0\\0&0&0&0\end{pmatrix}$$





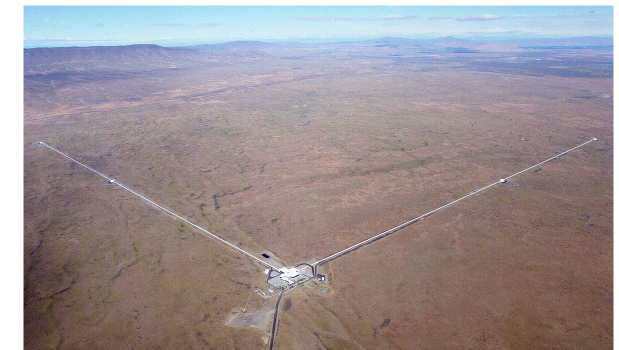
$$e^{\times}_{ab}\left(\hat{n}\right)=\begin{pmatrix}0&0&0&0\\0&0&1&0\\0&1&0&0\\0&0&0&0\end{pmatrix}$$

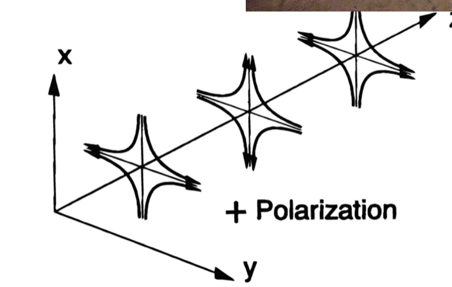
Abramovici et al. 1992

Detector Response

Detector Tensor

$$d^{ab} \equiv \frac{1}{2}\left(\hat{x}^a \hat{x}^b - \hat{y}^a \hat{y}^b\right)$$



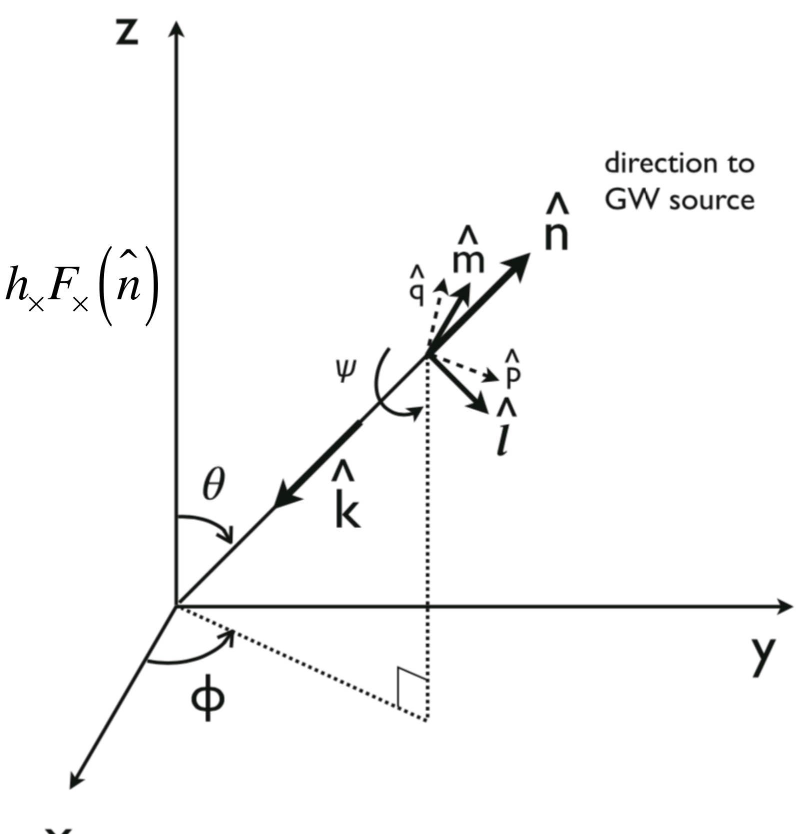


$$e^{+}_{ab}\left(\hat{n}\right) = \hat{l}_a \hat{l}_b - \hat{m}_a \hat{m}_b$$

$$e^{\times}_{ab}\left(\hat{n}\right) = \hat{l}_a \hat{m}_b + \hat{m}_a \hat{l}_b$$

Response of interferometer

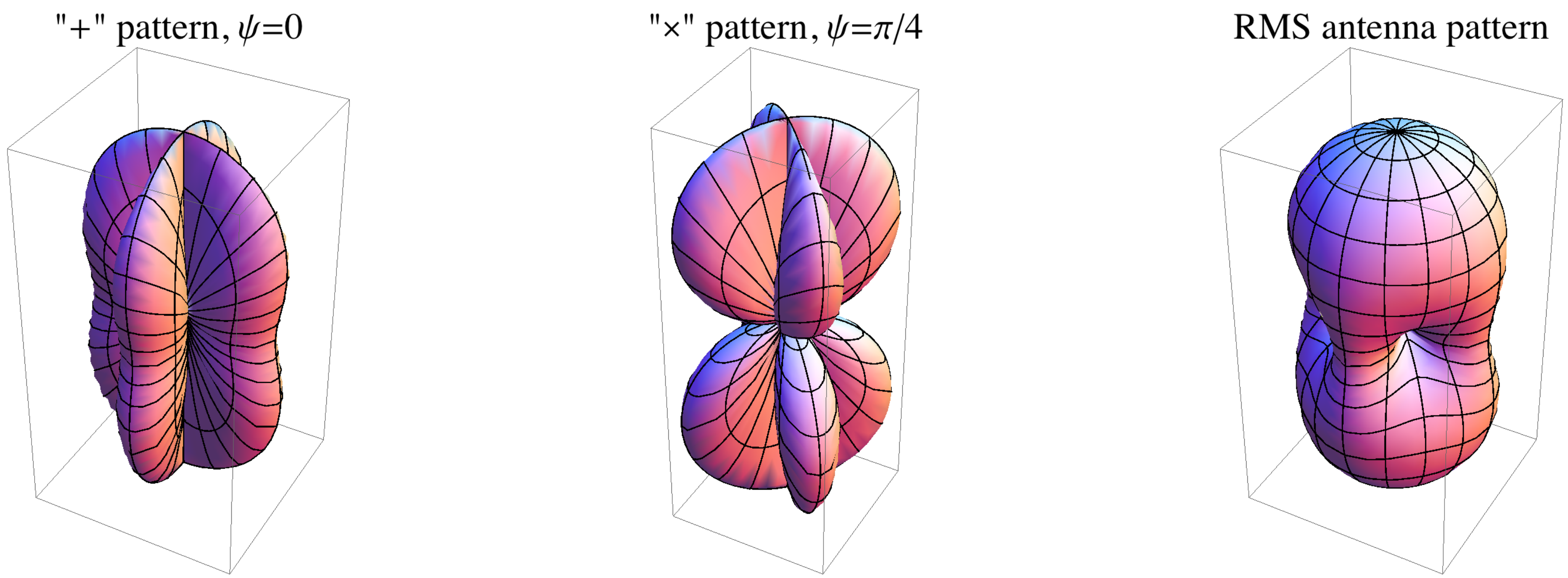
$$h\left(f,\hat{n}\right) = \frac{\Delta L}{L} = d^{ab} h_{ab}\left(f,\hat{n}\right) = h_+ d^{ab} e^{+}_{ab}\left(\hat{n}\right) + h_\times d^{ab} e^{\times}_{ab}\left(\hat{n}\right) = h_+ F_+\left(\hat{n}\right) + h_\times F_\times\left(\hat{n}\right)$$



Romano & Cornish 2017

Antenna Pattern

$F_{+/\times}$: antenna pattern

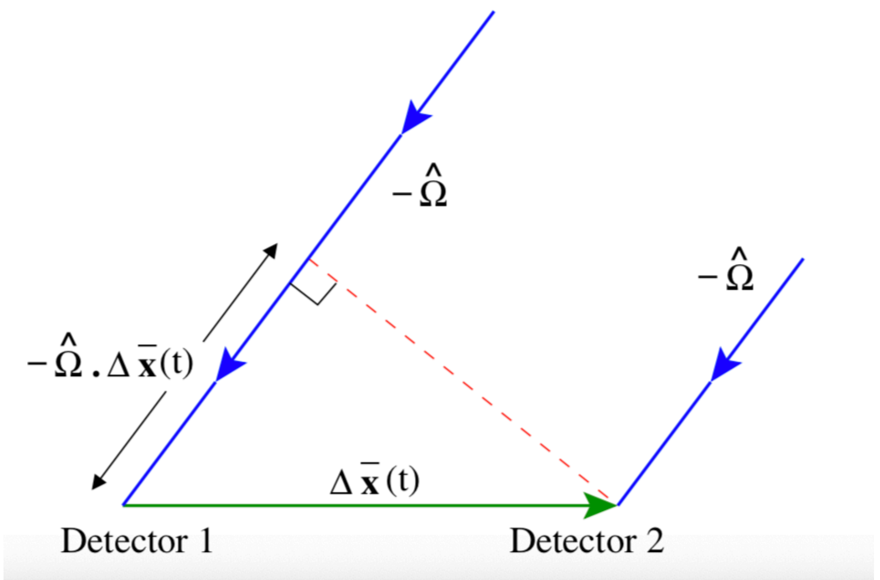


L. Singer arXiv:1501.03765

import pycbc.detector as Detector
f_plus, f_cross = Detector.overhead_antenna_pattern(ra, dec, polarization)

Cross-Correlation

Cross-Correlation of data in two detectors is very similar to the technique in radio astronomy. For a given source in the sky, the GW signal will arrive the two detector with a time delay.



$$\Delta t = \frac{\widehat{\Omega} \bullet \Delta x}{c} \qquad \Delta x : \text{baseline}$$

Mitra et al. 2008

The resolution of the image

$$\Delta\theta \sim \frac{\lambda_{GW}}{\Delta x \sin\theta}$$

The baseline changes as the Earth rotates. It changes the delay and cross-correlation of two GW signal as well.

Short Time Fourier Transform

Time series data of two detectors

$$d_I(t) = h_I(t) + n_I(t) \qquad I = 1,2$$

Short time Fourier transform

$$\tilde{d}(f;T) = \int_{T-\tau/2}^{T+\tau/2} d(t) e^{-i2\pi ft}\, dt$$

τ **is the duration of the time segment. It should be much longer than the traveling time of GW between two detectors but smaller enough so that the change of antenna-pattern function is not significant.**

Now we define the cross-correlation of the output as

$$C(f;T) \equiv \tilde{d}_1^*(f,T)\tilde{d}_2(f,T)$$

The expected value

$$\langle C(f;T) \rangle = \frac{\tau}{2} \langle \tilde{h}_1^*(f,T)\tilde{h}_2(f,T) \rangle$$

Inverse FFT of cross-correlation give the delay information

$$C(\tau)=\lim_{T\to\infty}\frac{1}{T}\int_{-T/2}^{T/2} h_1(t')h_2(t'+\tau)\,dt' = \lim_{T\to\infty}\frac{1}{T}\int_{-T/2}^{T/2}\int_{-\infty}^{\infty}\tilde{h}_1^{*}(f)\tilde{h}_2(f')e^{-i2\pi ft'}e^{i2\pi f'(t'+\tau)}\,df\,df'\,dt'$$

$$=\lim_{T\to\infty}\frac{1}{T}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\tilde{h}_1^{*}(f)\tilde{h}_2(f')e^{i2\pi f'\tau}\,df\,df'\,\delta(f-f')$$

$$=\int df\left[\lim_{T\to\infty}\frac{1}{T}\tilde{h}_1^{*}(f)\tilde{h}_2(f)\right]e^{i2\pi f\tau}$$

$$C(\tau)\ \underset{\text{iFT}}{\overset{\text{FT}}{\rightleftarrows}}\ \left\langle \tilde{h}_1^{*}\tilde{h}_2\right\rangle$$

Statistic of Sources of SGWB

GW radiation from different directions of the sky are uncorrelated, unpolarized, Gaussian-stationary

$$\left\langle h_A\left(f,\hat{n}\right)\right\rangle = 0$$

A: + or x

$$\left\langle h_A^*\left(f,\hat{n}\right)h_{A'}\left(f',\hat{n}'\right)\right\rangle = \frac{1}{4}\mathbf{P}\left(f,\hat{n}\right)\delta\left(f-f'\right)\delta_{AA'}\delta\left(\hat{n},\hat{n}'\right)$$

Define

$$S_h\left(f\right) = \int d^2\hat{n}\mathbf{P}\left(f,\hat{n}\right)$$

and

$$\Omega_{gw}\left(f\right) \equiv \frac{f}{\rho_c}\frac{d\rho_{gw}}{df} = \frac{2\pi^2}{3H_0^2}f^3 S_h\left(f\right)$$

$$\rho_c = \frac{3c^2H_0^2}{8\pi G}\text{ : critical energy density of the un iverse}$$

We can assume power from all the directions of the sky have the same frequency response

$$\mathbf{P}\left(f,\hat{n}\right) = H\left(f\right)P\left(\hat{n}\right)$$

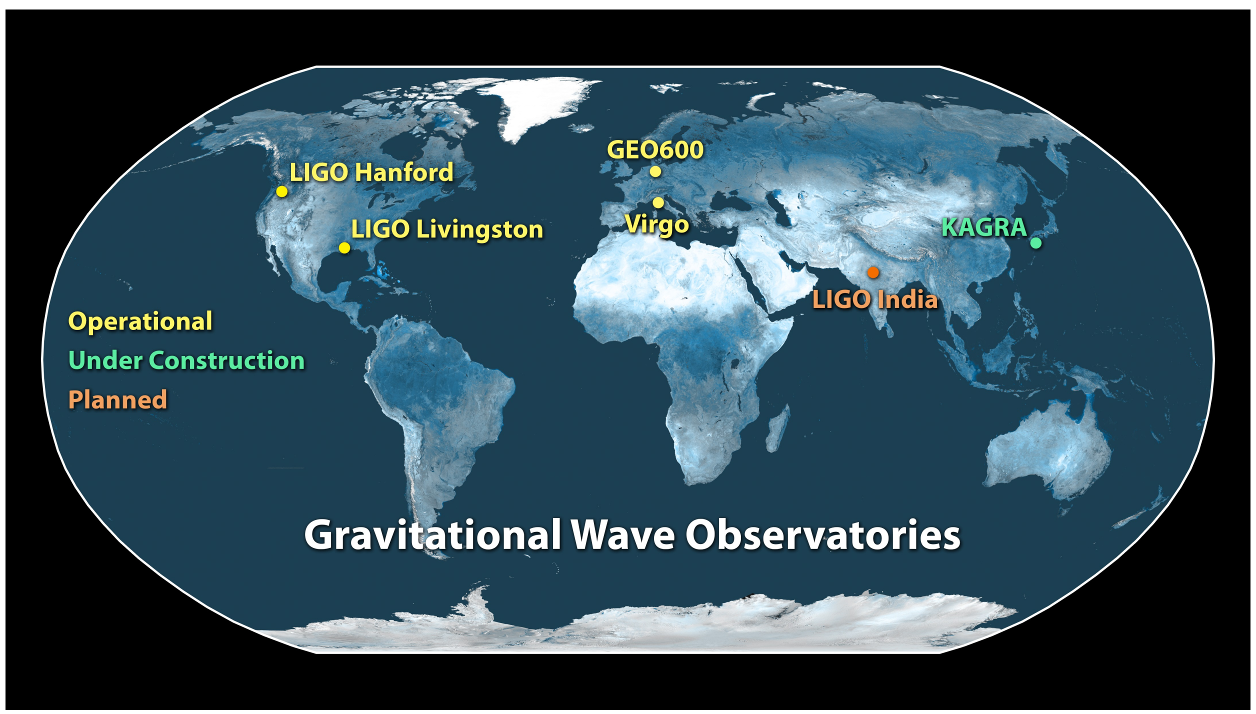
Directional Search of SGWB

- Mechanisms of GW production
- Astrophysical distribution of GW sources in galaxies
- Galaxy formation and distribution of the large scale structures of the universe

Cross-Correlation of signals

$$\left\langle h_1^*(f;T)h_2(f;T)\right\rangle = \int d^2\hat{n}\sum_A F_{1A}\left(\hat{n};T\right)F_{2A}\left(\hat{n};T\right)\mathbf{P}_A\left(\hat{n},f\right)e^{i2\pi f\left(\hat{n}\cdot\Delta\vec{x}/c\right)}$$

Correlation of signal from two detectors



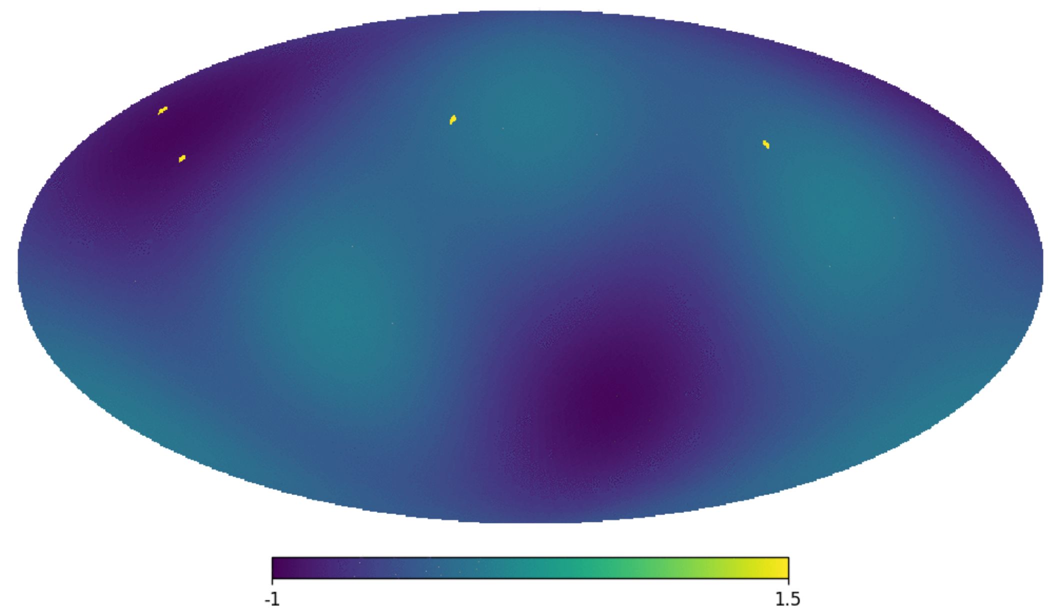
https://www.ligo.caltech.edu/image/ligo20160211c

Visibility $V\left(\Delta\vec{x}\right) = \int d\hat{\Omega}\Delta I\left(\hat{\Omega}\right)A\left(\hat{\Omega}\right)e^{i2\pi f\Delta\vec{x}\cdot\hat{\Omega}/c}$

$$\gamma\left(f,t,\hat{n}\right)=\sum_A F_{1A}\left(\hat{n};T\right)F_{2A}\left(\hat{n};T\right)e^{i2\pi f\left(\hat{n}\bullet\Delta\vec{x}/c\right)}$$

```python
import numpy as np
import healpy as hp
nside = 32
npix=hp.nside2npix(nside)
ipix=np.arange(npix)
theta, phi=hp.pix2ang(nside, ipix)
ra = -phi
dec = np.pi/2.0 - theta
```

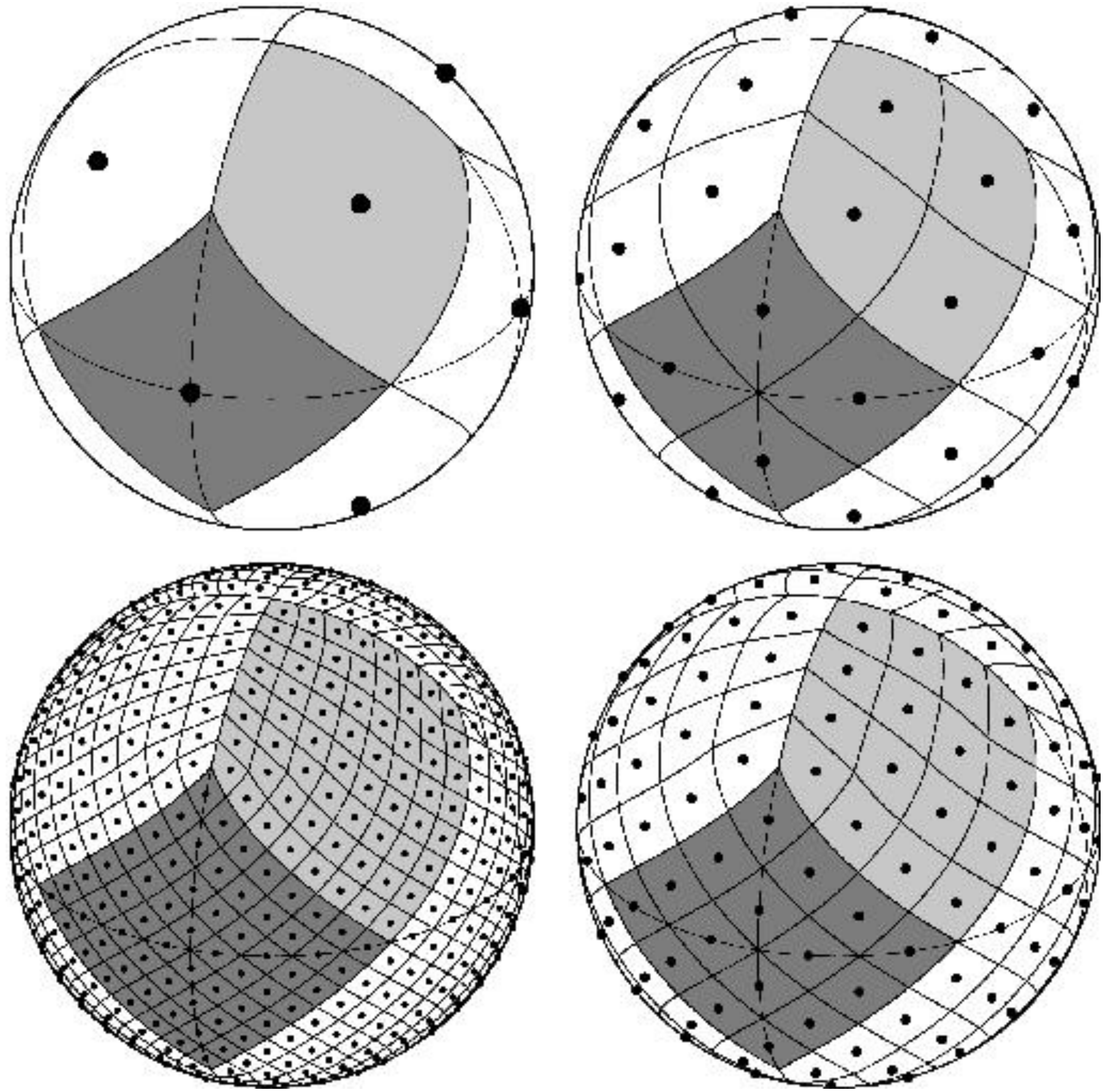
LH f=0




```python
from pycbc.detector import Detector
from lal.antenna import AntennaResponse
from pycbc.detector import Detector


def gamma_map(freq, nside, detector1, detector2, polarization, gps_time, ra, dec):
    npix=len(ra)
    domega= 4*np.pi/npix
    antenna_p=np.zeros([len(ra)], dtype=complex)
    antenna_c=np.zeros([len(ra)], dtype=complex)

    det1=Detector(detector1)
    det2=Detector(detector2)
    timedelay=det1.time_delay_from_detector(det2, ra, dec, gps_time)

    for i in range(npix):
        phase = 2.0 *np.pi * freq *timedelay[i]
        resp1 = AntennaResponse(detector1, ra[i], dec[i], psi=polarization, scalar=True, vector=True, times=gps_time)
        resp2 = AntennaResponse(detector2, ra[i], dec[i], psi=polarization, scalar=True, vector=True, times=gps_time
        antenna_p[i] = (resp1.plus*resp2.plus)*complex(np.cos(phase), np.sin(phase)) #*2.5
        antenna_c[i] = (resp1.cross*resp2.cross)*complex(np.cos(phase), np.sin(phase)) #*2.5
    return antenna_p, antenna_c
```

```python
    gmst = pycbc.detector.gmst_accurate(gps)
```

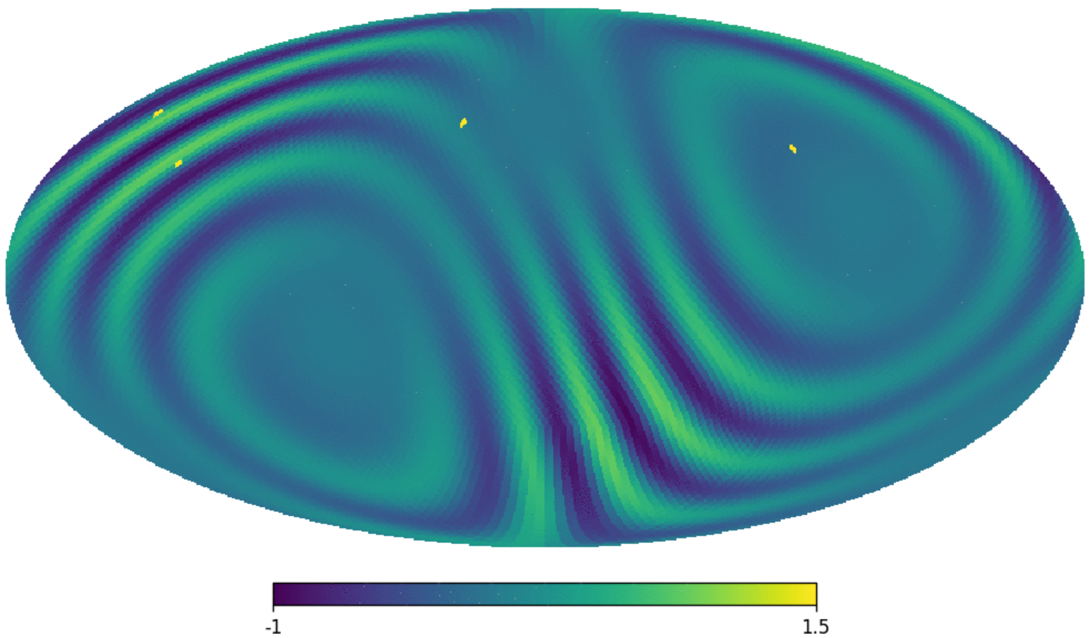
Healpix



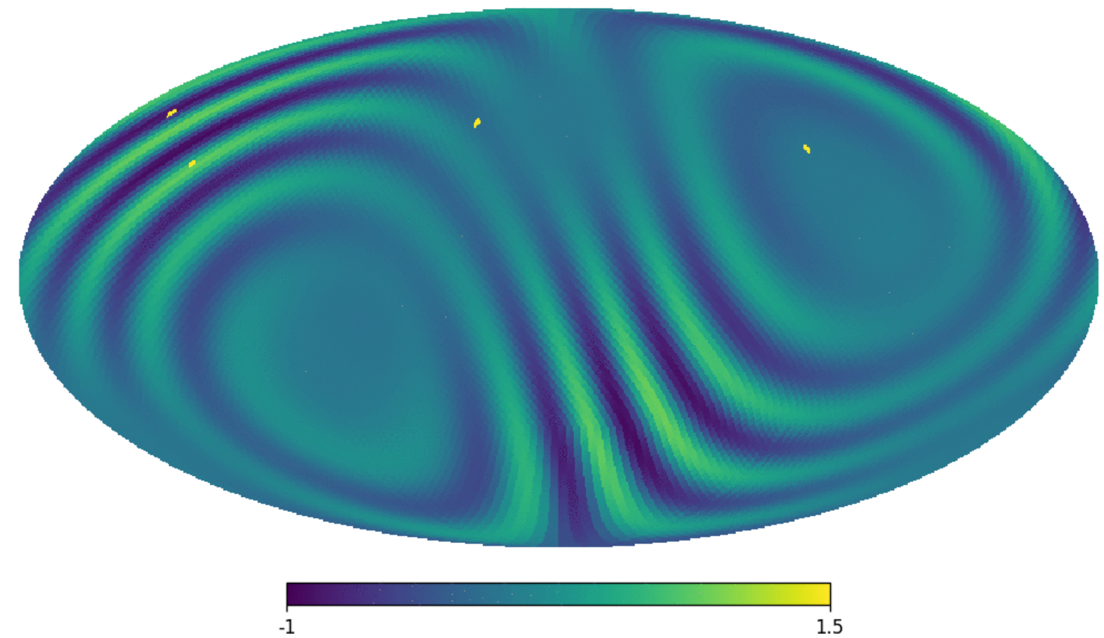
https://healpix.sourceforge.io

$$\gamma\left(f,t,\hat{n}\right)=\sum_A F_{1A}\left(\hat{n};T\right)F_{2A}\left(\hat{n};T\right)e^{i2\pi f\left(\hat{n}\bullet\Delta\vec{x}/c\right)}$$

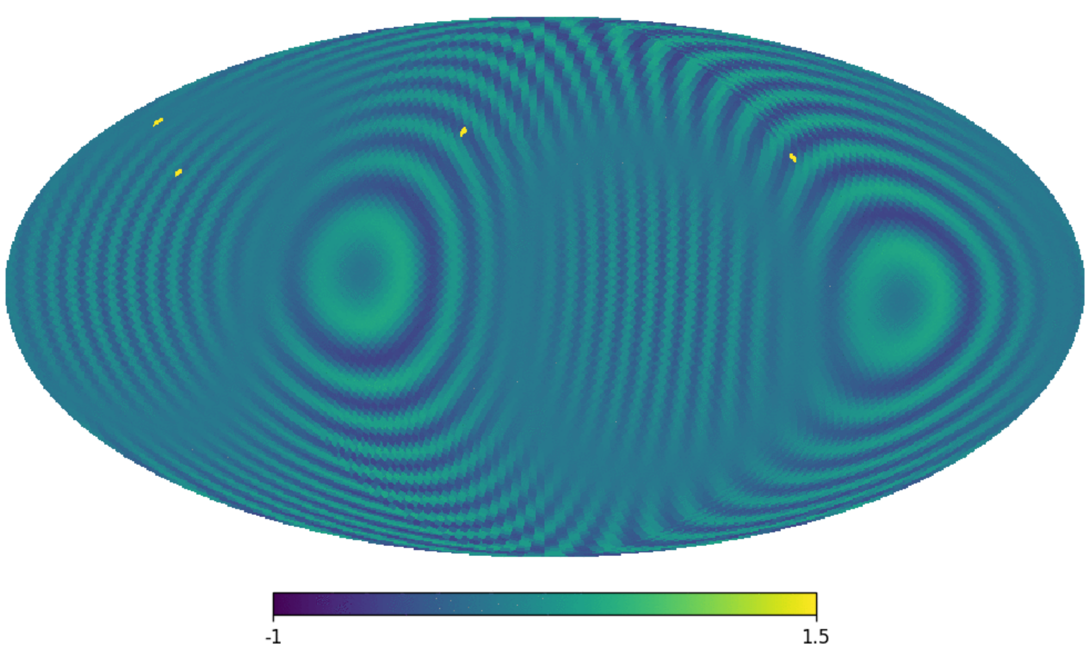
LH real f=200



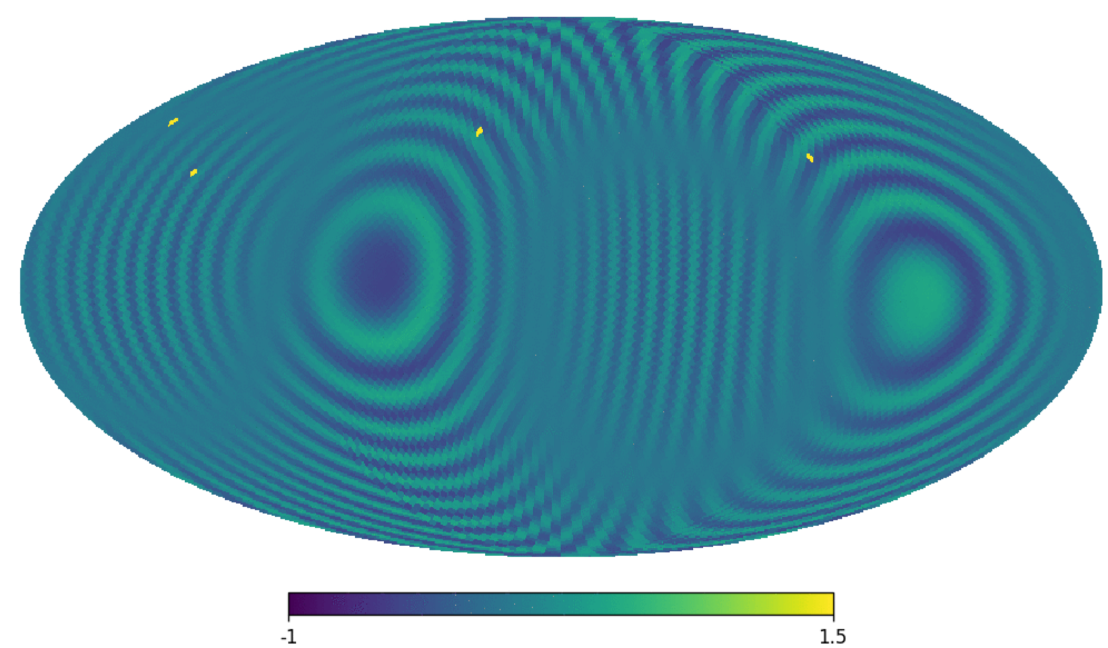
LH Imag f=200



VK real f=200



VK Imag f=200



Dirty Map

$$C = KP + n$$

Correlation **Map** **Noise**

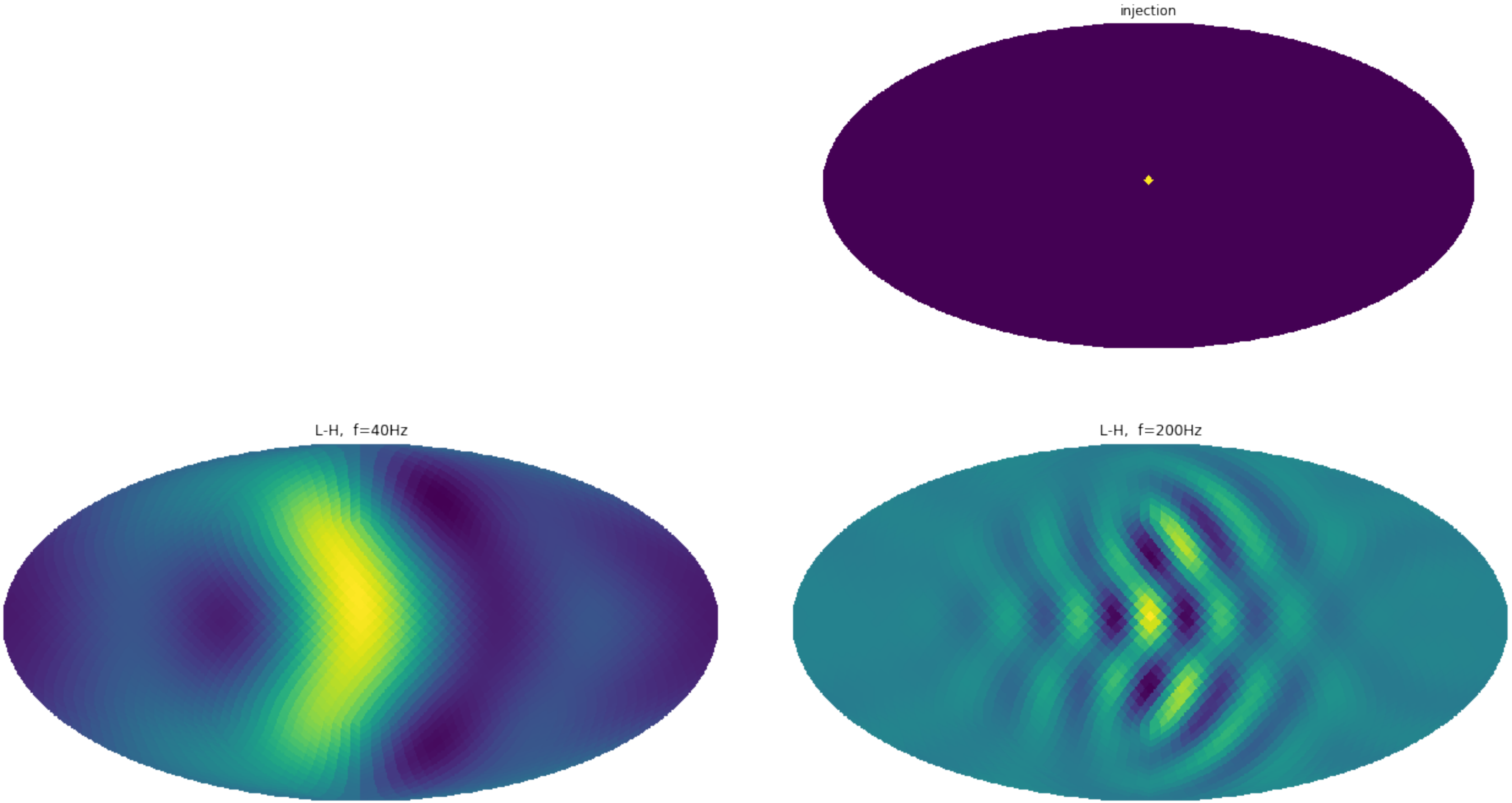
Dirty Map :

$$P_{Dirty} = K^{\dagger} N^{-1} C$$

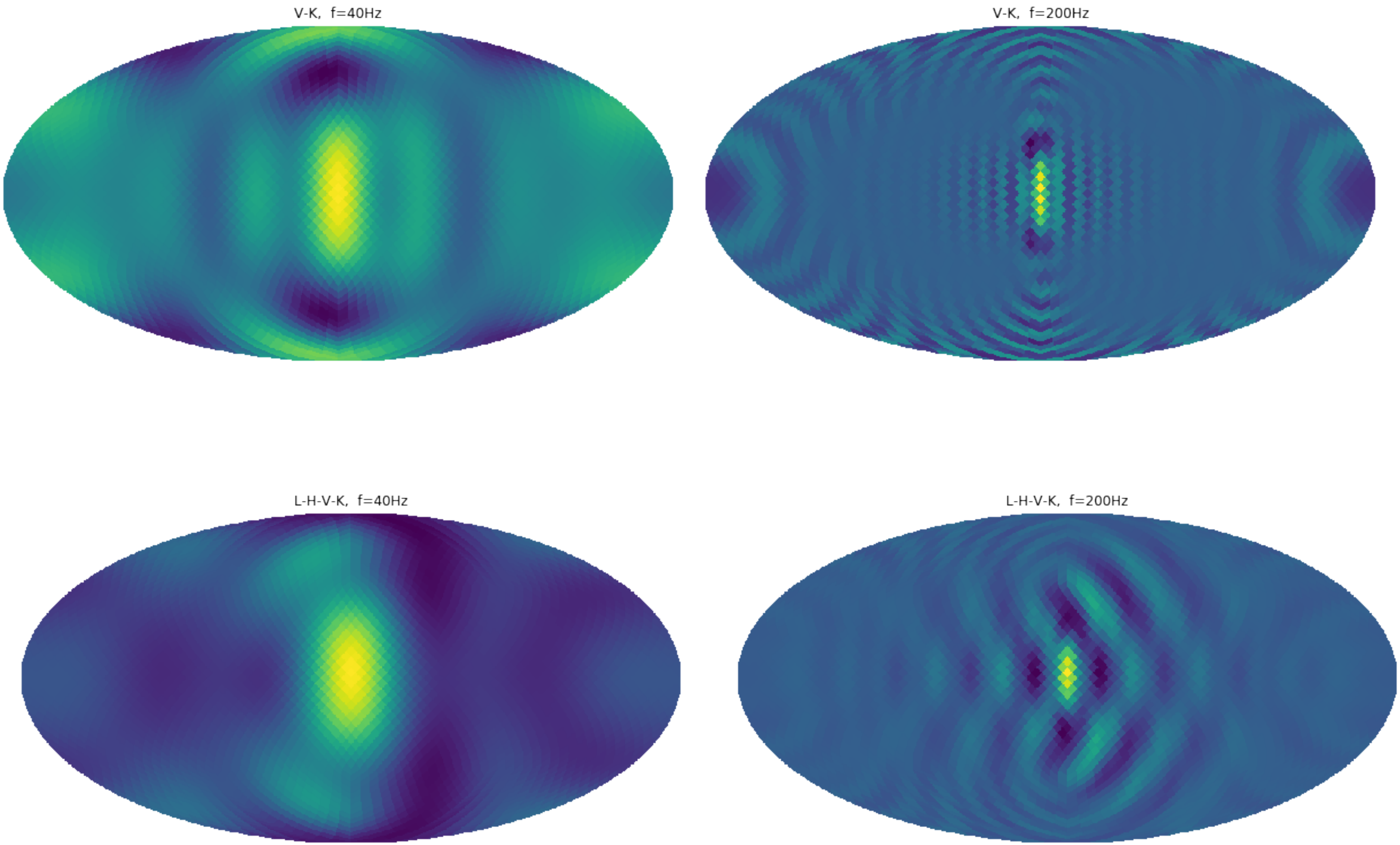
Covariant matrix of noise

$$N_{fT,f'T'} = \left\langle C^{*}{}_{fT} C_{f'T'} \right\rangle - \left\langle C_{fT} \right\rangle \left\langle C_{f'T'} \right\rangle \sim \frac{\tau^{2}}{4} \delta_{ff'} \delta_{TT'} S_{1}(f;T) S_{2}(f;T)$$

Dirty Map



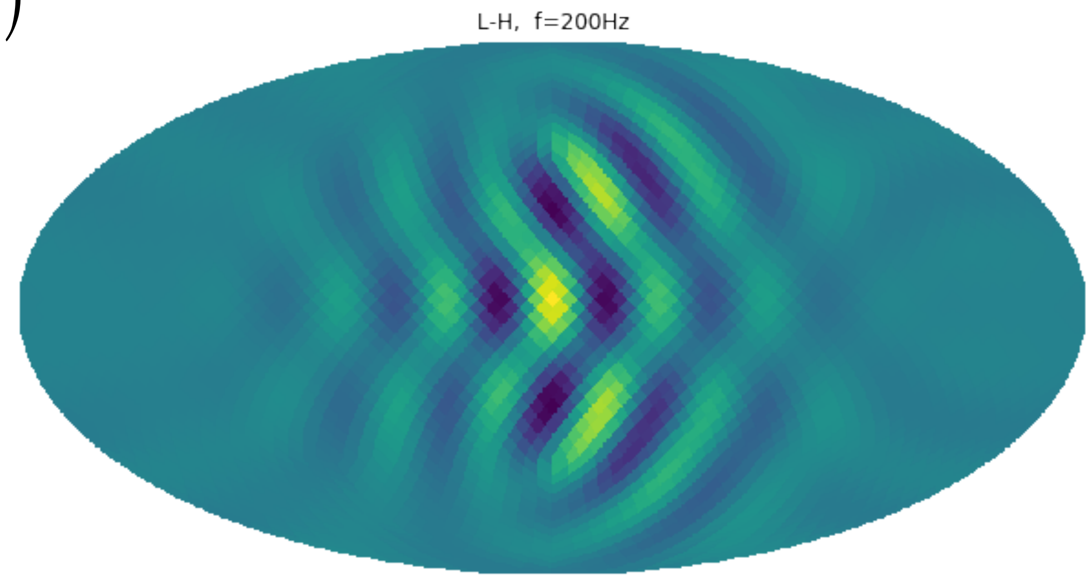
Dirty Map



Clean Map

To estimate the map P, we can minimize the chi square

$$\chi^2(P) = \left(C^*_{fT} - \left\langle C^*_{fT}(P)\right\rangle\right) N^{-1}_{fT,f'T'} \left(C_{fT} - \left\langle C_{fT}(P)\right\rangle\right)$$



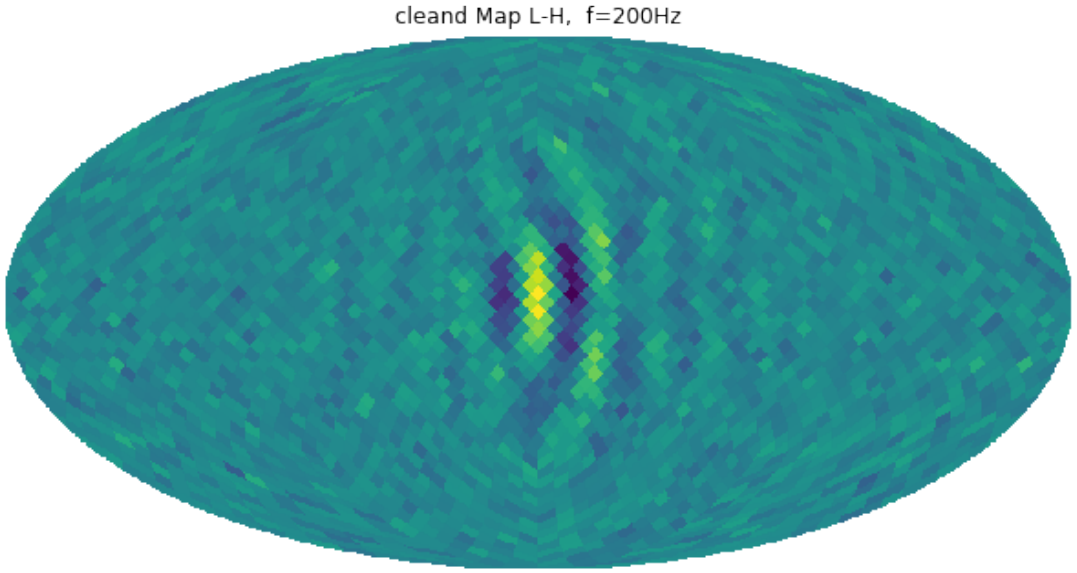
Clean map

$$\widehat{P} = \Gamma^{-1} X$$

$$\Gamma = 4 \sum_{fT} \frac{H^2(f)}{P_1(f,T) P(f,T)} \gamma^*_p(f,T) \gamma_{p'}(f,T)$$

Or in matrix

$$\Gamma = \left(K^\dagger N^{-1} K\right)$$



Clean Map

Indeed, Γ is a singular matrix. We need SVD method to inverse the matrix.

$$\Gamma = USU^\dagger$$

$$S = diag\left(s_i\right)$$ **Si are the eigenvalues of Γ**

Set a threshold s_{min}, replace si < s_{min} by s_{min}.

$$\Gamma' = US'U^\dagger$$