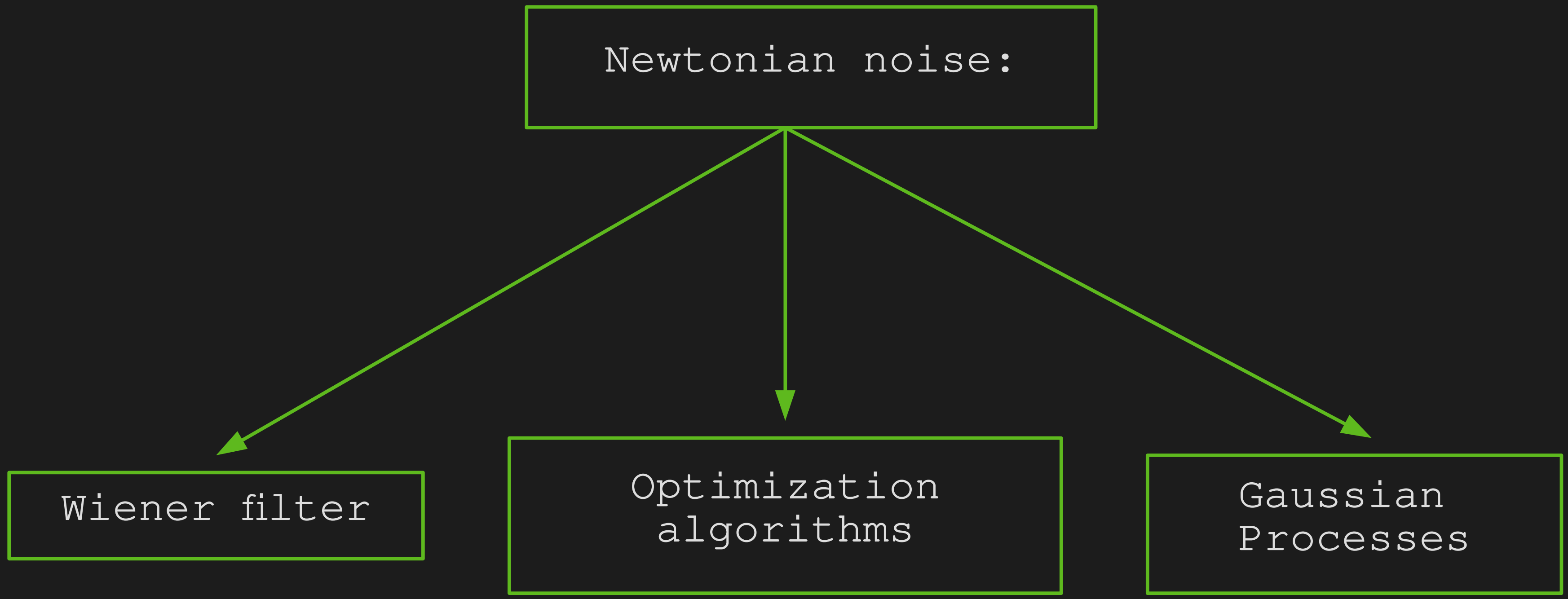
Newtonian Noise Cancellation Strategies and Optimization Problems

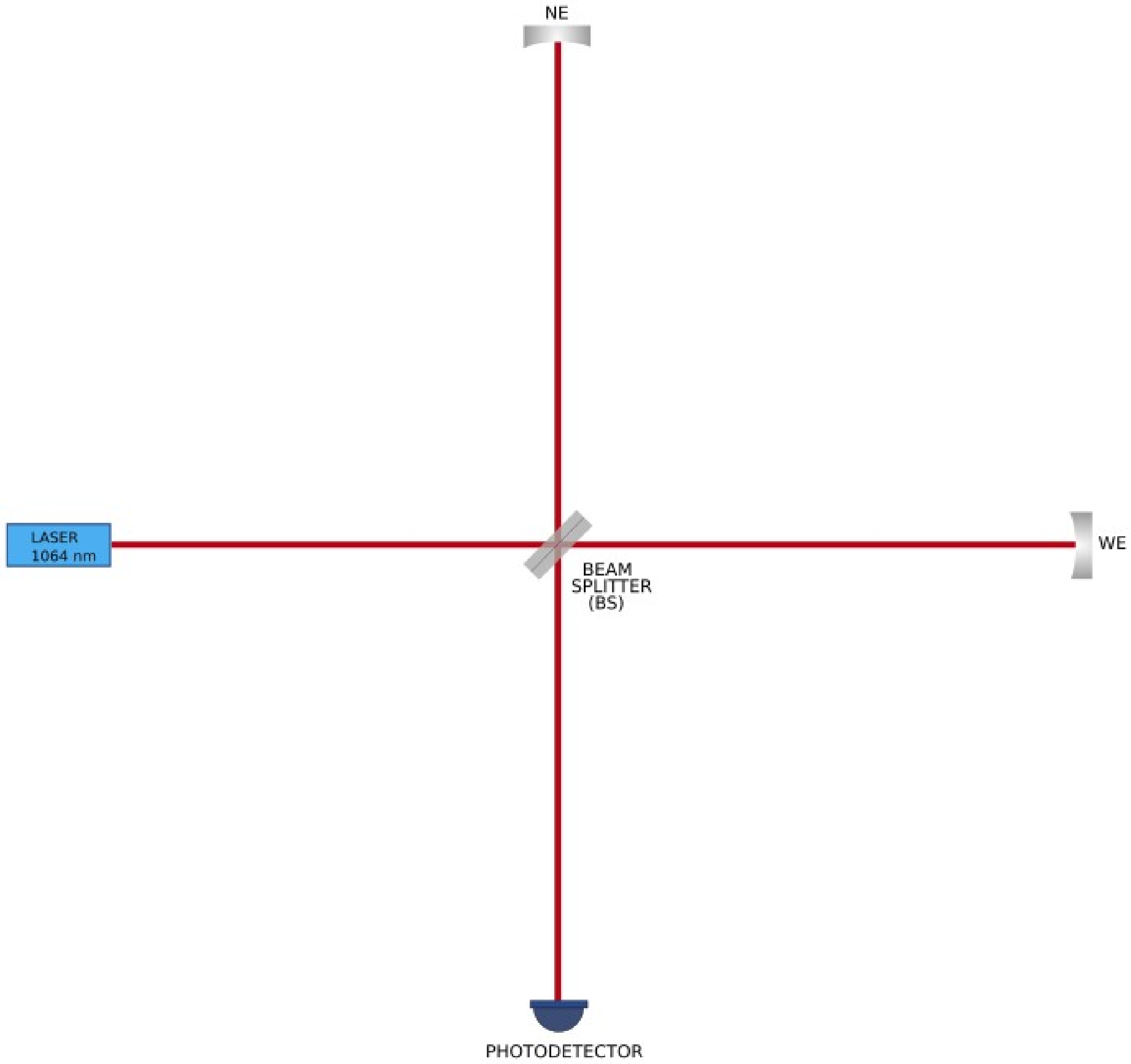
G2net, August 2021

Author: Francesca Badaracco

UCLouvain

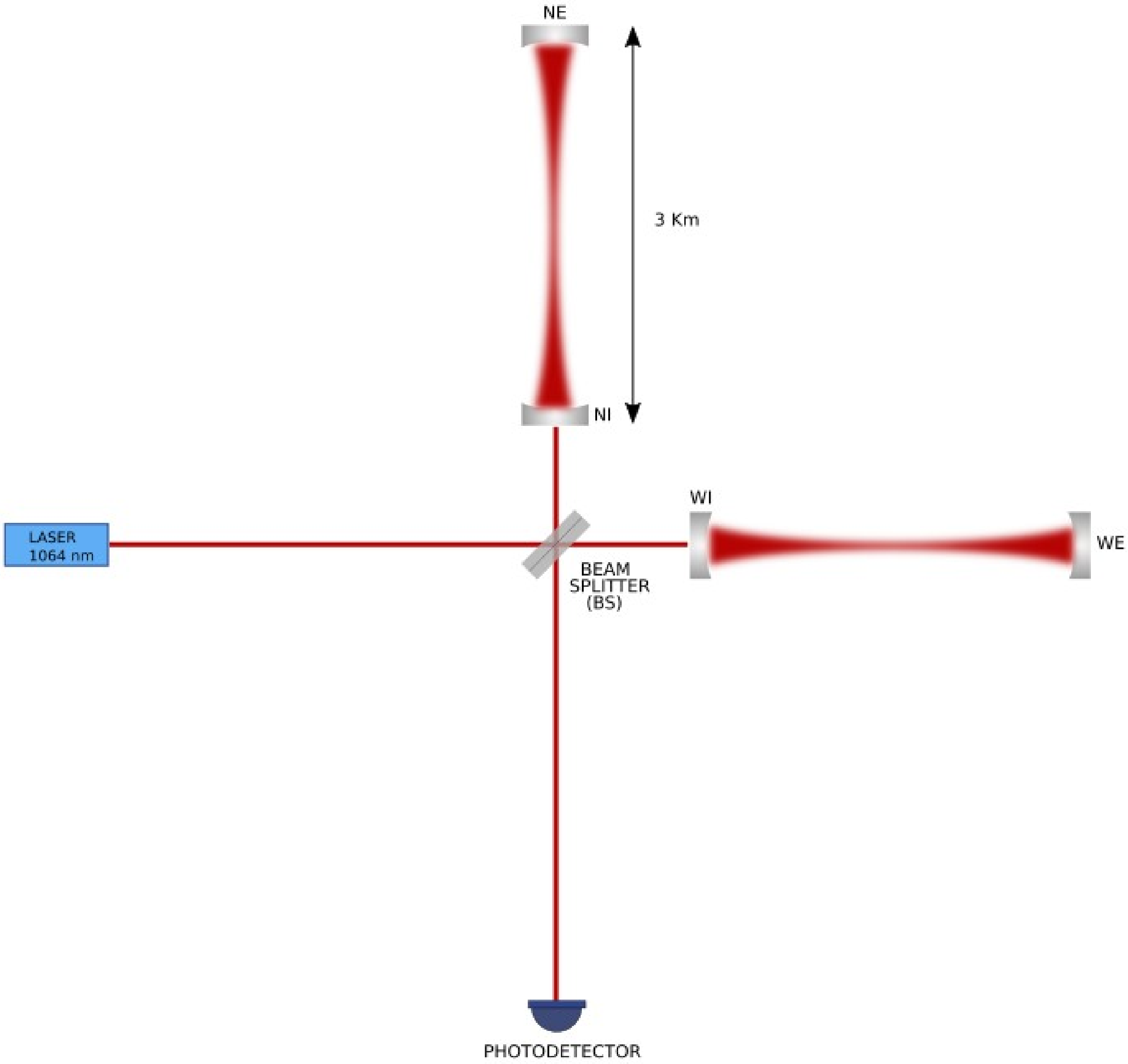
Newtonian Noise Cancellation Strategies and Optimization Problems





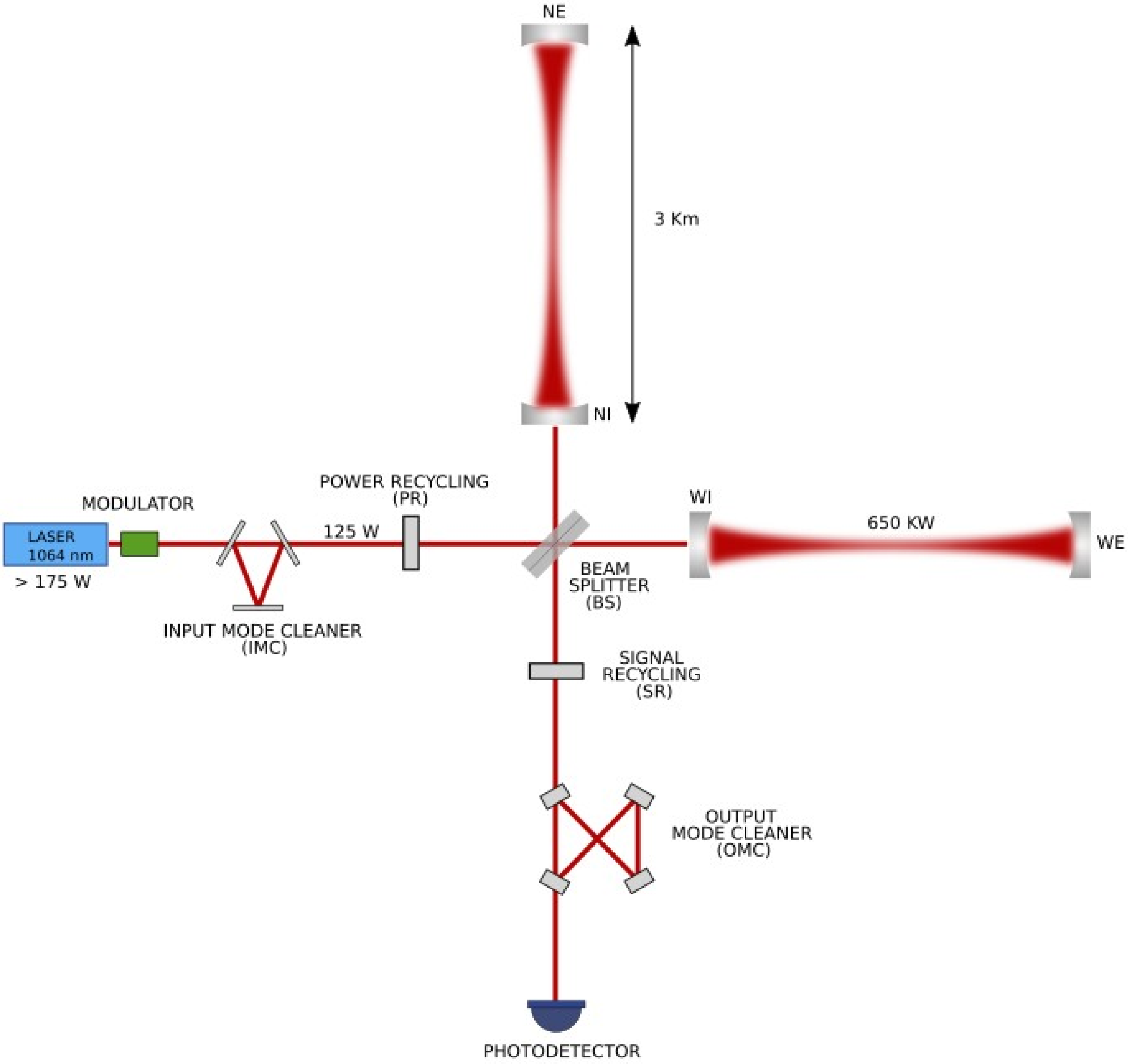
Gravitational wave detector’s **working principle**

We want to measure the **phase change**



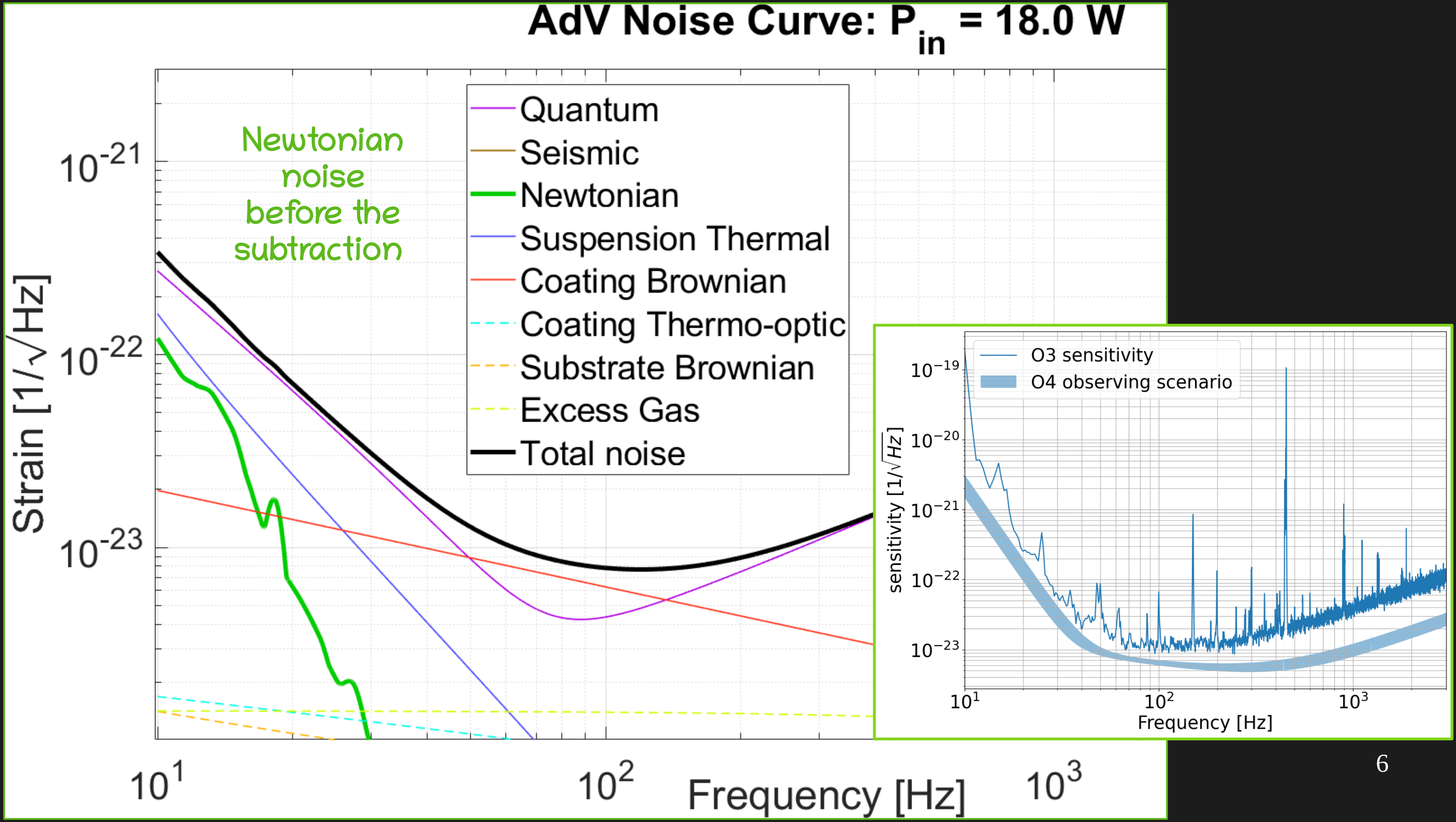
Gravitational wave detector's **working principle**

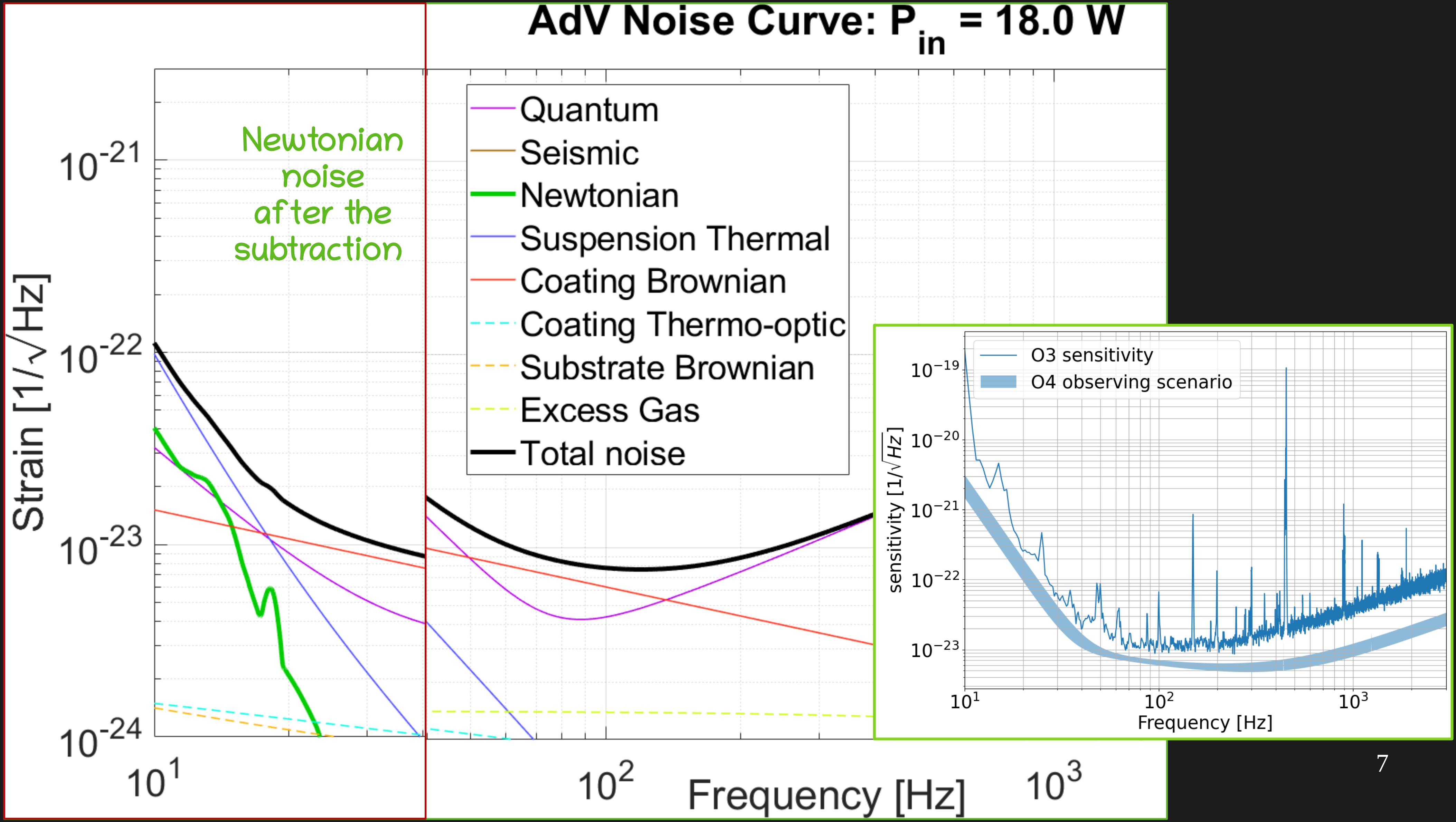
We want to measure the **phase change**



Gravitational wave detector's **working principle**

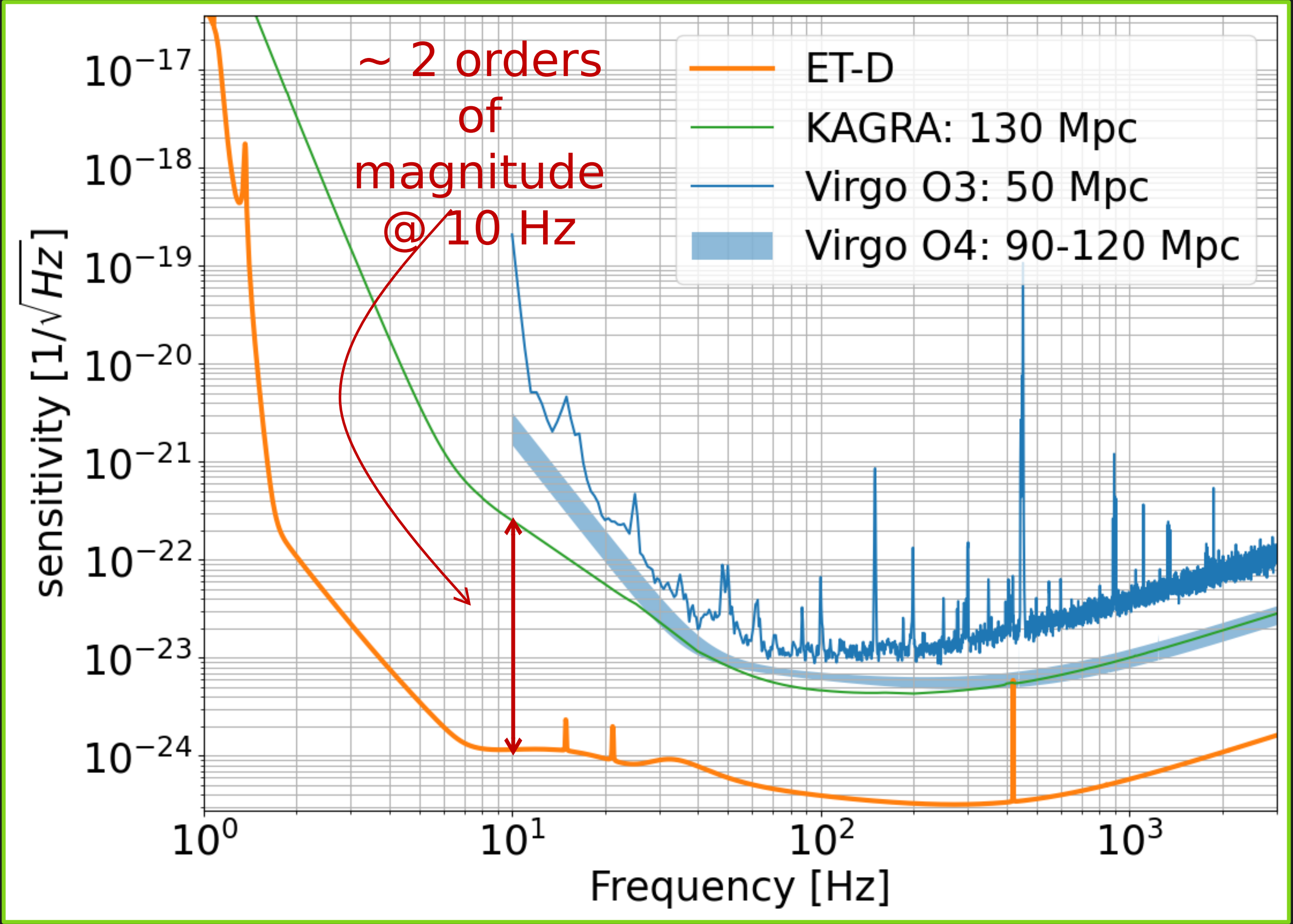
We want to measure the **phase change**


AdV Noise Curve: P_{in} = 18.0 W
Newtonian noise before the subtraction
Quantum
Seismic
Newtonian
Suspension Thermal
Coating Brownian
Coating Thermo-optic
Substrate Brownian
Excess Gas
Total noise
Strain [1/√Hz]
Frequency [Hz]
O3 sensitivity
O4 observing scenario
sensitivity [1/√Hz]
Frequency [Hz]



AdV Noise Curve: P_{in} = 18.0 W
Newtonian noise after the subtraction
Quantum
Seismic
Newtonian
Suspension Thermal
Coating Brownian
Coating Thermo-optic
Substrate Brownian
Excess Gas
Total noise
Strain [1/√Hz]
Frequency [Hz]
O3 sensitivity
O4 observing scenario
sensitivity [1/√Hz]
Frequency [Hz]

Improving the **low frequency** band is very expensive: do we really **need** it?



New possible discoveries

BNS: Hours - Days
Parameter estimation
EM early warning
Sky localization with only ET

Massive BBHs:
Higher redshift PBHs?

Search of stochastic background

More stable interferometer!

$$\delta\rho_{\mathrm{temp}}(\mathbf{r},t) = -\frac{\rho_0}{T_0}\delta T(\mathbf{r},t)$$

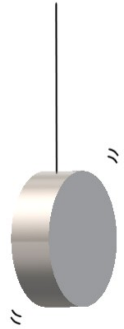
$$\delta\rho_{\mathrm{press}}(\mathbf{r},t) = \frac{\rho_0}{\gamma p_0}\delta p(\mathbf{r},t)$$

Adiabatic index

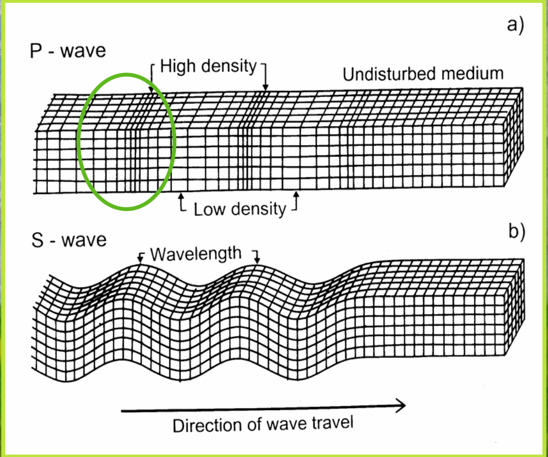
ATMOSPHERIC NN

$$\delta\phi(\mathbf{r}_0,t) = -G\int dV\,\frac{\delta\rho(\mathbf{r},t)}{|\mathbf{r}-\mathbf{r}_0|}$$

Newtonian Noise (NN):
Perturbation of the gravity field due to a variation in the density (δρ) of the surrounding media.

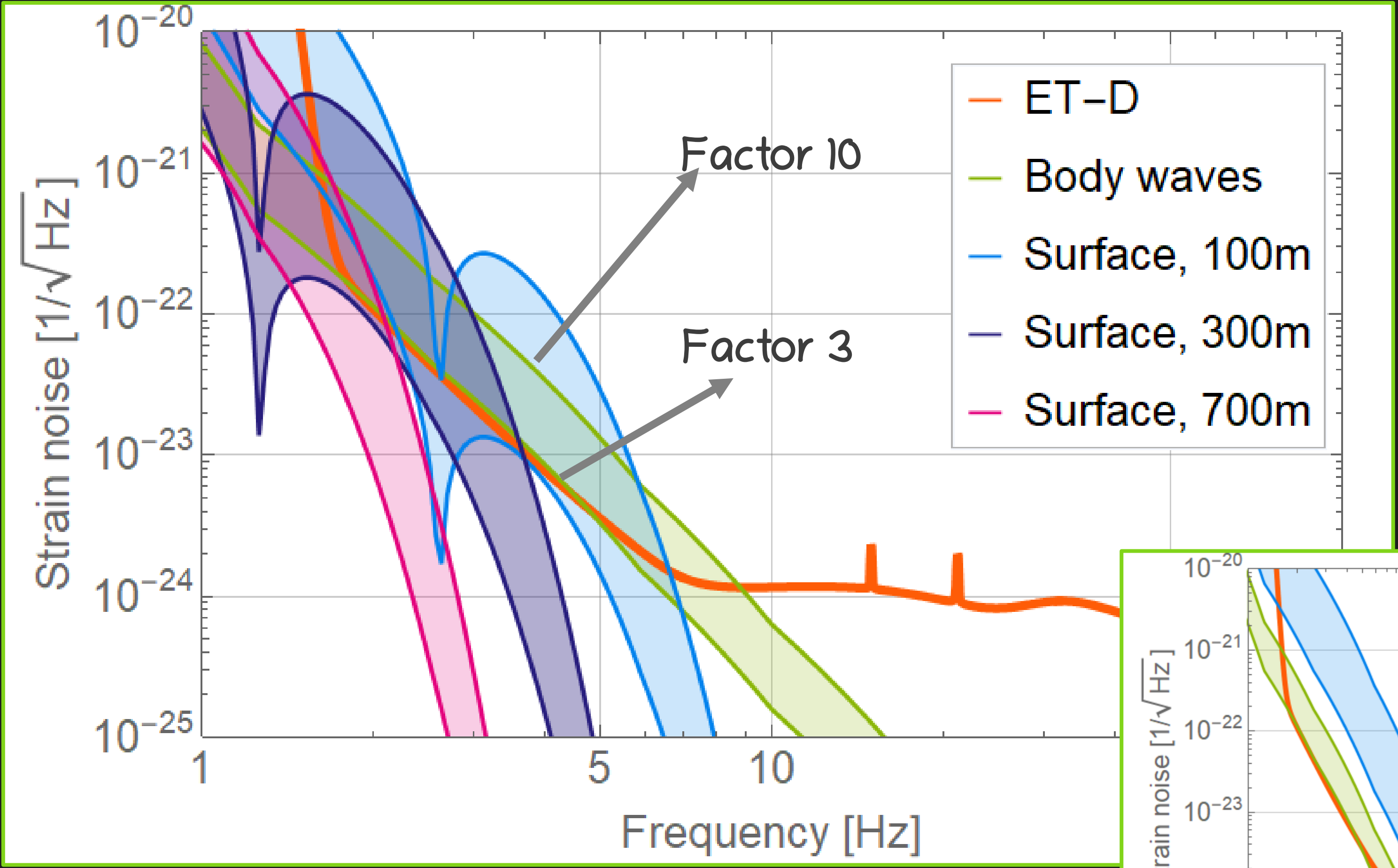


$$\delta\rho \rightarrow NN$$

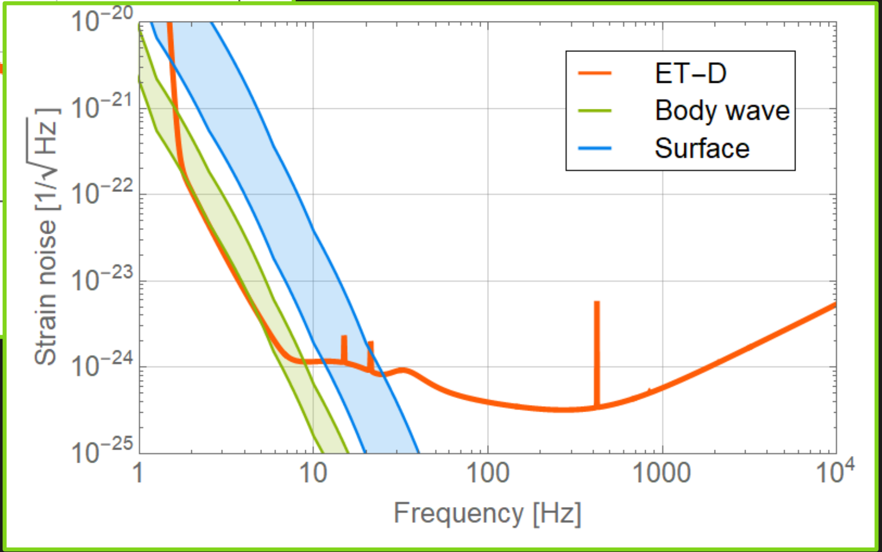


$$\delta\rho_{\mathrm{seis}}(\mathbf{r},t) = -\nabla\cdot(\rho(\mathbf{r})\boldsymbol{\xi}(\mathbf{r},t))$$

SEISMIC NN



Strain noise [1/√Hz]
Frequency [Hz]
Factor 10
Factor 3
ET-D
Body waves
Surface, 100m
Surface, 300m
Surface, 700m

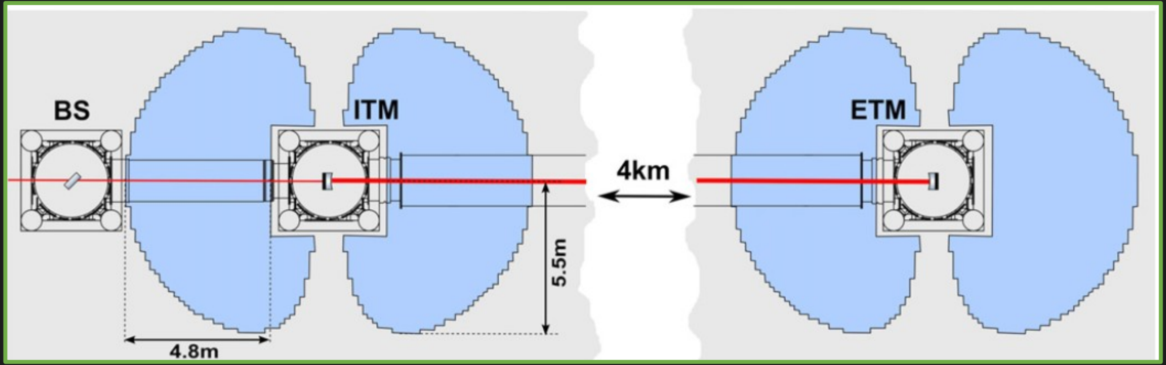


Strain noise [1/√Hz]
Frequency [Hz]
ET-D
Body wave
Surface

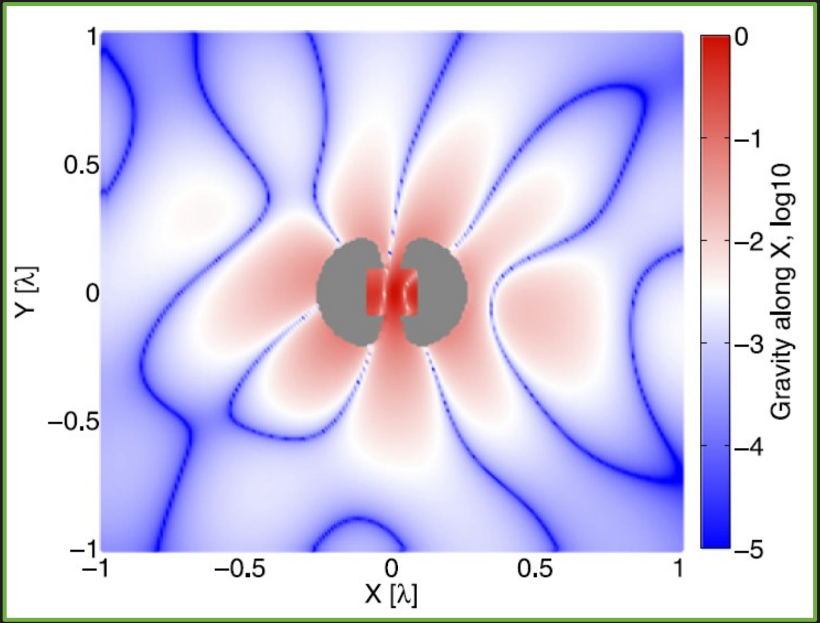
How can we reduce the Newtonian Noise?

- Excavating and removing material around the test masses (Recesses).
- Metamaterials.
- Active noise cancellation.

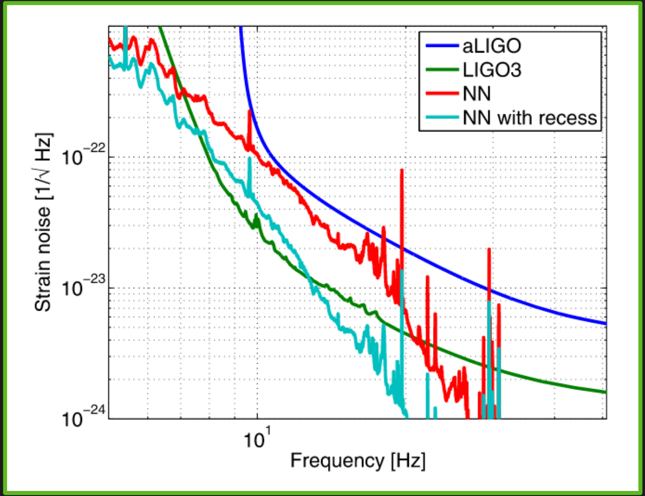
Recesses



J. Harms and S. Hild, Classical and Quantum Gravity31, 185011(2014), arXiv:1406.2253 [gr-qc]

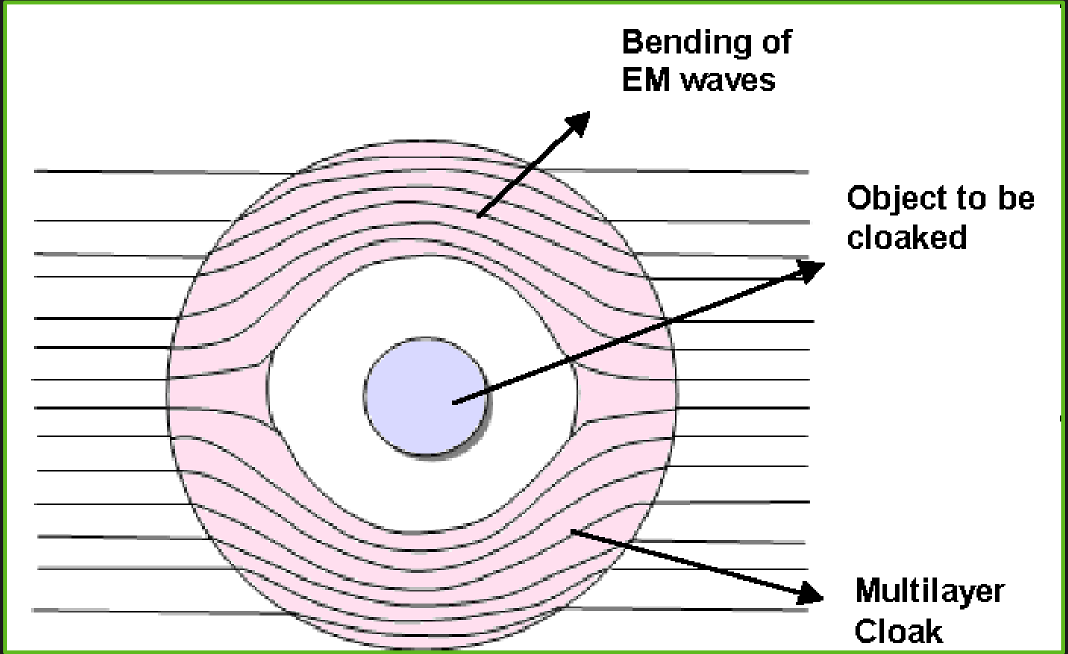


Suppression factors between 2 and 4 were obtained around 10 Hz with a recess 4 m deep and 11 m width on each side of a test mass.



Gravity perturbation of the test mass (normalized by its maximum value) at a specific frequency contributed by each point on the surface.

Metamaterials



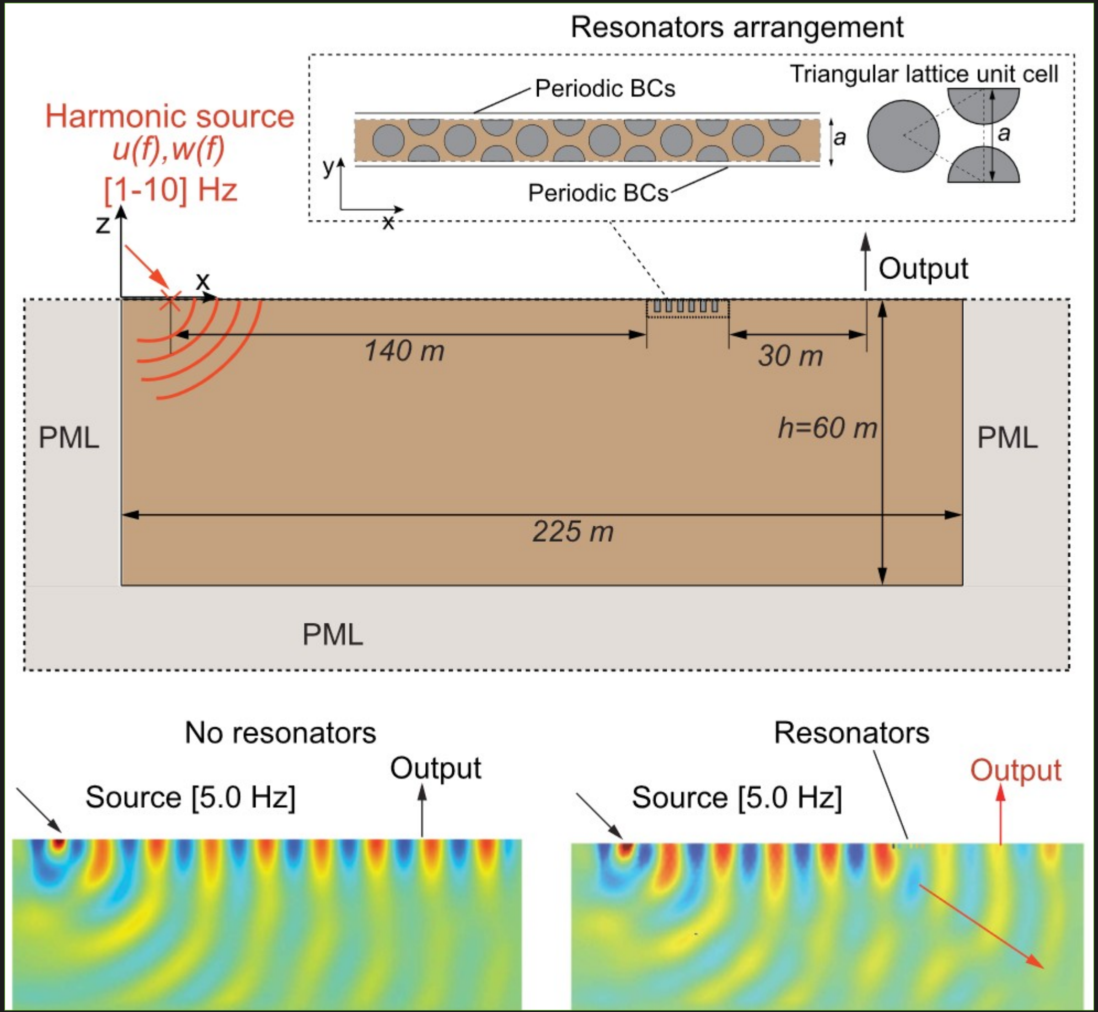
Choudhury, B., & Jha, R. (2013). A Review of Metamaterial Invisibility Cloaks. Cmc-computers Materials & Continua, 33, 277-310.

Metamaterials

Inspired by physical concepts well established in wave propagation control, like phononic crystals (also called acoustic metamaterials).

The longitudinal resonances of trees couple with the vertical component of the Rayleigh wave and attenuate the surface ground motion by redirecting part of the elastic energy into the bulk.

Soil-embedded resonators: the seismic metabarrier can attenuate surface ground motion within the 1–10 Hz range.

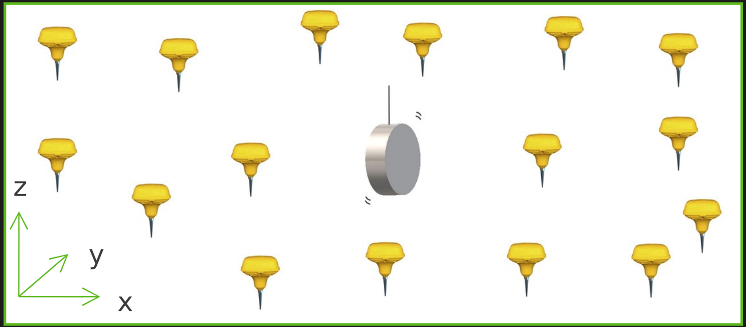


A. Palermo, S. Krödel, A. r. Marzani, and C. Daraio, ScientificReports6, 39356 (2016).

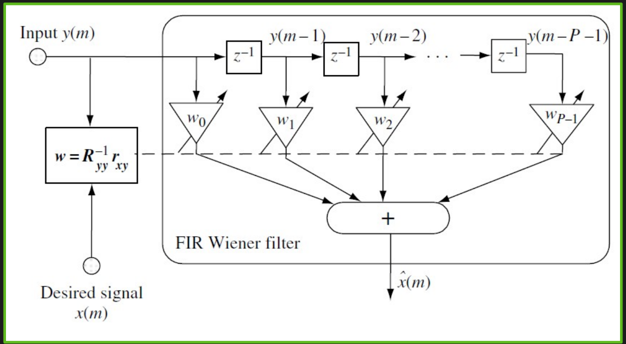
Active noise cancellation

- NN: it cannot be physically **shielded**
- We can perform an **active** noise cancellation
- Linear filter: **Wiener filter** (optimal filter)

Sensor array

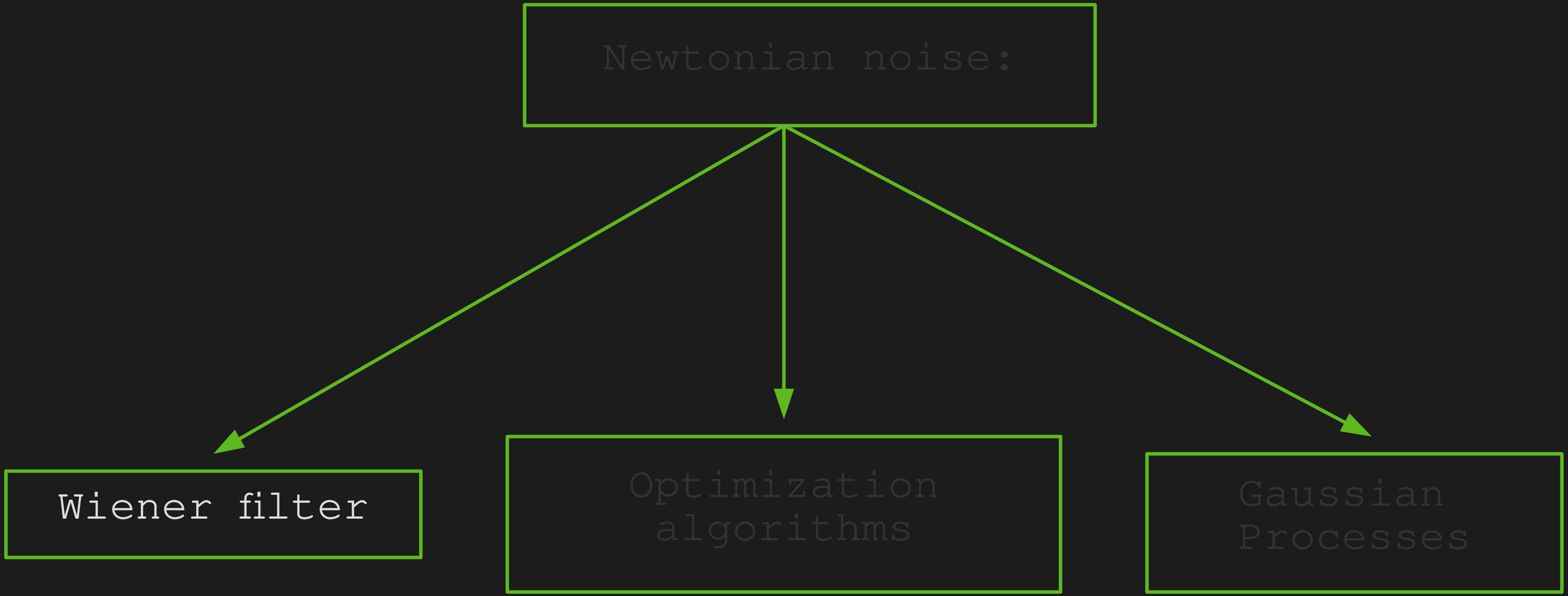


Wiener Filter



Newtonian **N**oise (NN) **cancellation**

Newtonian Noise Cancellation Strategies and Optimization Problems



Wiener Filter is the way:

Assumptions:

- **Stationary** signal
- Linear relationship

Wide stationary process

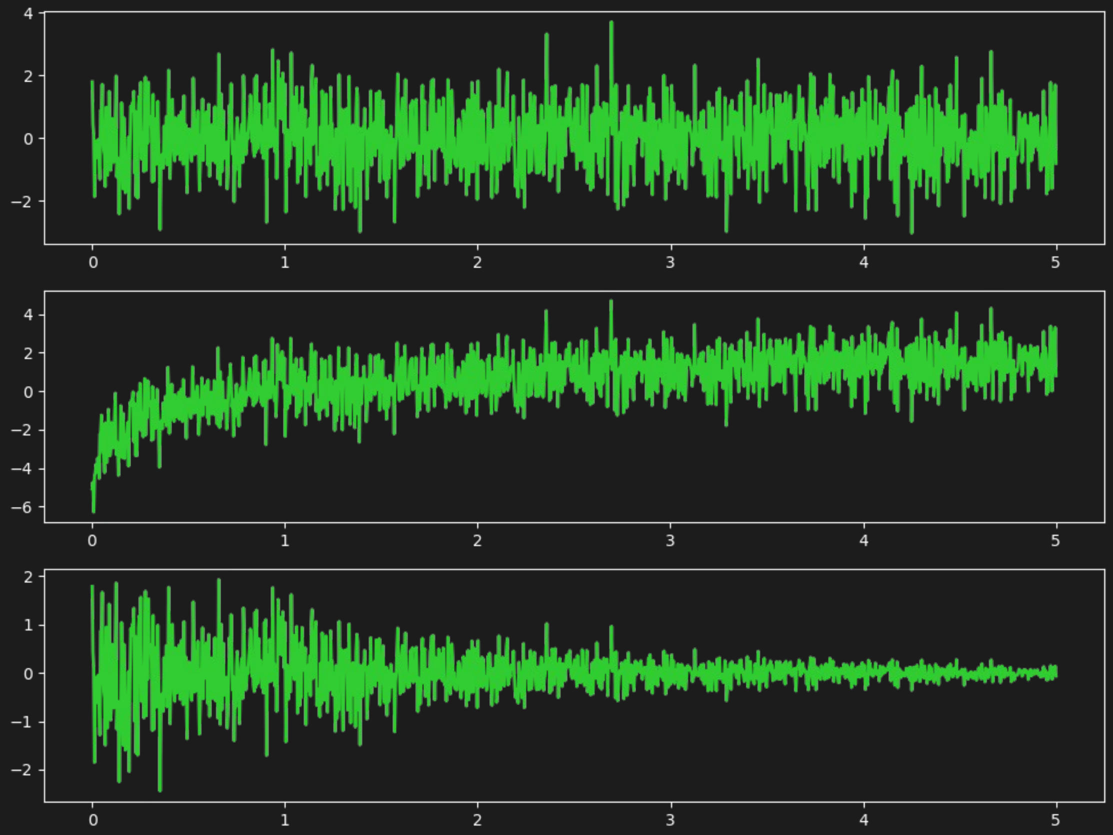
$$\mu(t) = E[X(t)] = const$$

$$C_{XX}(t_1, t_2) = E[(X(t_1) - \mu(t_1))(X(t_2) - \mu(t_2))] = C_{XX}(t_2 - t_1)$$

Wiener Filter is the way:

Assumptions:

- **Stationary** signal
- Linear relationship



stationary **mean**
and
stationary **variance**

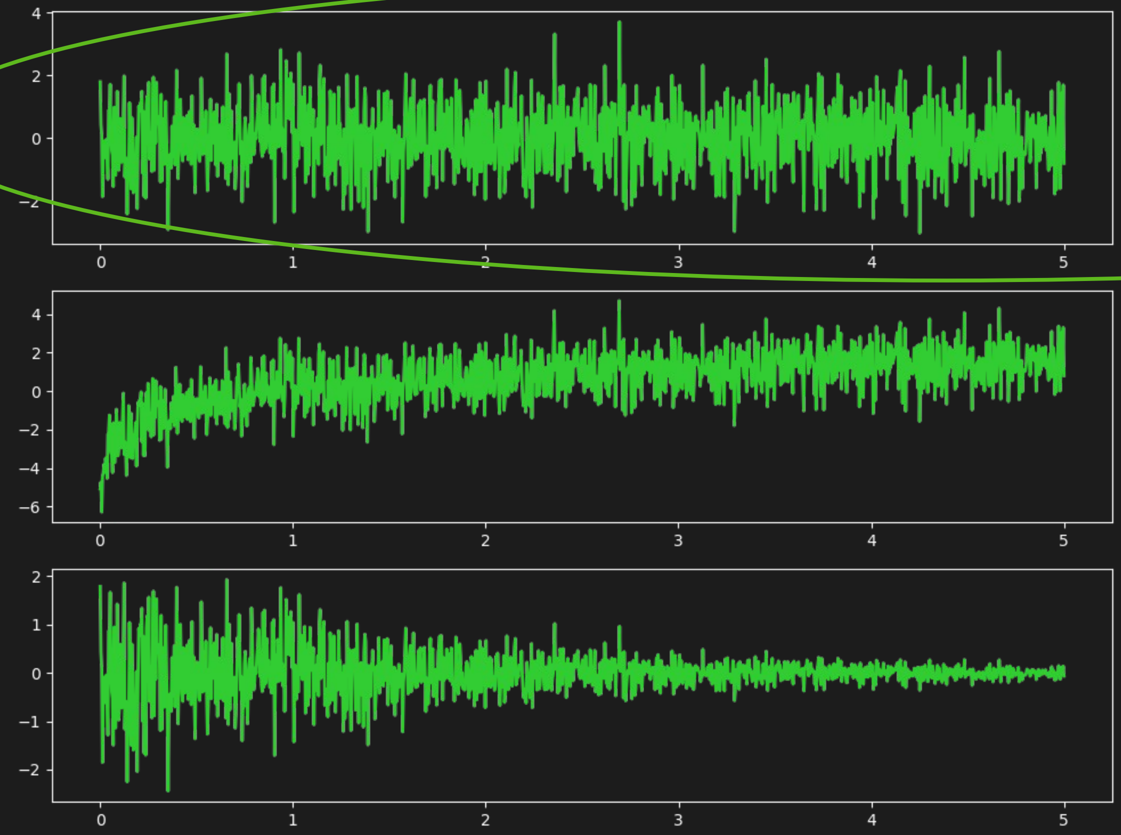
non-stationary **mean**
and
stationary **variance**

stationary **mean**
and
non-stationary **variance**

Wiener Filter is the way:

Assumptions:

- **Stationary** signal
- Linear relationship



stationary **mean**
and
stationary **variance**

non-stationary **mean**
and
stationary **variance**

stationary **mean**
and
non-stationary **variance**

Wiener Filter is the way:

Assumptions:

- Stationary signal
- **Linear** relationship

$$\delta\phi(\mathbf{r}_0, t) = -G \int dV \frac{\delta\rho(\mathbf{r}, t)}{|\mathbf{r}_0 - \mathbf{r}|} \qquad \delta\rho(\mathbf{r}, t) = -\nabla \cdot (\delta\rho(\mathbf{r})\xi(\mathbf{r}, t))$$

Linear relationship with the **seismic displacement**

$$\delta\mathbf{a}(\mathbf{r}_0, t) = -\nabla\delta\phi(\mathbf{r}_0, t) = G \int dV \rho(\mathbf{r})\xi(\mathbf{r}, t) \cdot \frac{\mathbf{r} - \mathbf{r}_0}{|\mathbf{r} - \mathbf{r}_0|^3}$$

Wiener Filter is the way:

Assumptions:

- Stationary signal
- **Linear** relationship

Filter order

$$\hat{x}(m) = \sum_{k=0}^{P-1} w_k \, y(m-k)$$

Estimated value of the Newtonian noise

Wiener filter coefficients

Measured signal (seismic displacement)

Discrete time

Wiener Filter is the way:

Assumptions:

- Stationary signal

GW data – Estimated NN

$$\hat{x}(m) = \sum_{k=0}^{P-1} w_k \, y(m-k)$$

Estimated value of the Newtonian noise

Discrete time

Wiener filter coefficients

Measured signal (seismic displacement)

Array optimization

Wiener filter to perform a NN cancellation (**time domain**):

$$\hat{x}(m) = \sum_{k=0}^{P-1} w_k \, y(m-k)$$

$$R(\omega) = 1 - \frac{\vec{C}_{sn}^{\dagger} \mathbf{C}_{ss}^{-1} \vec{C}_{sn}}{C_{nn}}$$

Wiener filter performances (**frequency domain**):

REMEMBER!!!

Residual in frequency domain

$$R(\omega) = 1 - \frac{\vec{C}_{sn}^{\dagger}\mathbf{C}_{ss}^{-1}\vec{C}_{sn}}{C_{nn}}$$

$$C_{sn_i}(\omega) = E[s_i^*(\omega)n(\omega)]$$

i^{th} element of the **vector** containing all the cross power spectral densities of all the seismic sensors with the test mass (containing also the NN)

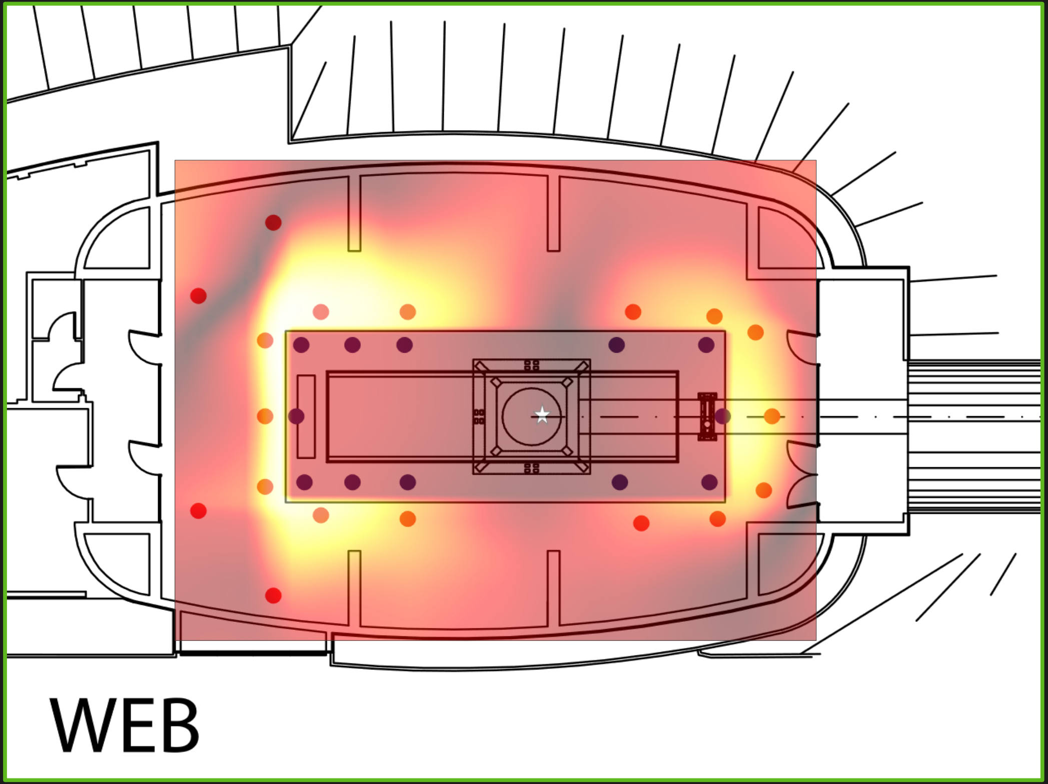
$$C_{nn}(\omega) = E[n^*(\omega)n(\omega)]$$

Power Spectral Density of the target signal (test mass). It's a **scalar**

$$C_{ss_{ij}}(\omega) = E[s_i^*(\omega)s_j(\omega)]$$

i^{th} element of the **matrix** containing all the cross power spectral densities between all the seismic sensors

The residual will depend on the frequency, the number of sensors and on their positions:
In 2D we have **2N coordinates**, where N is the number of the sensors.



Update Wiener filter every hours: LINK

Array optimization

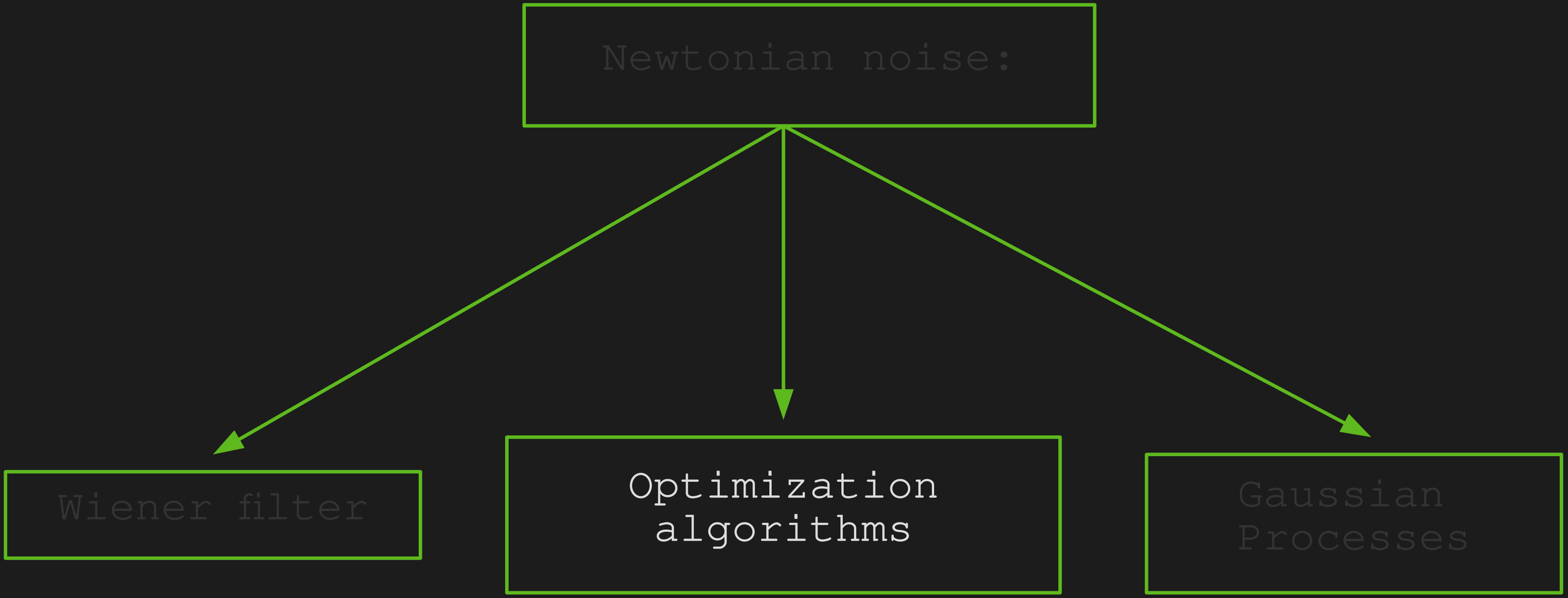
↓

Array deployment

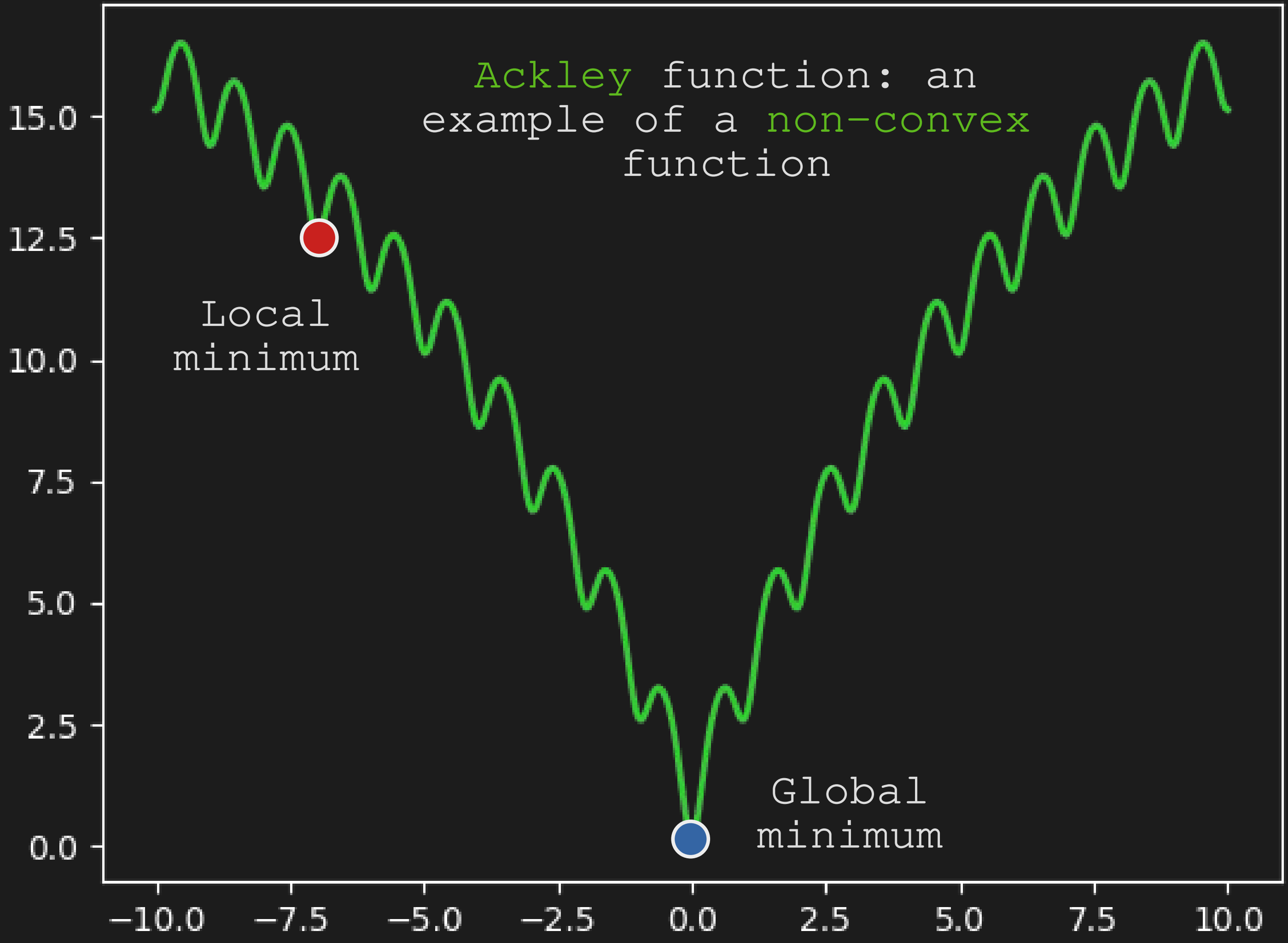
↓

Subtraction pipeline (applying wiener filter)

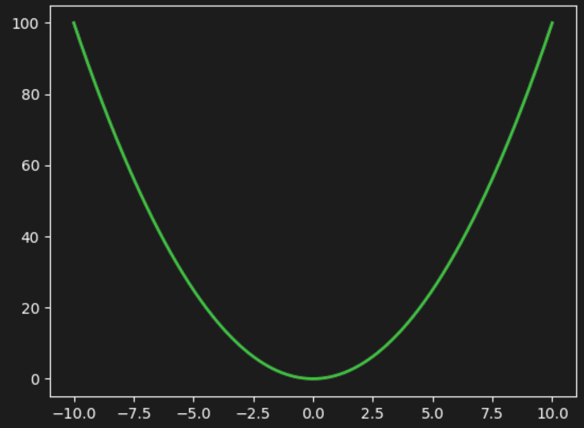
Newtonian Noise Cancellation Strategies and Optimization Problems



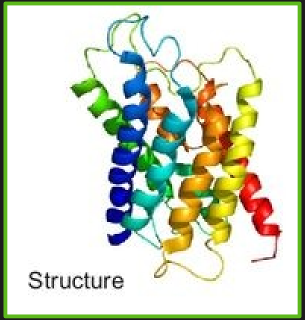
Global Optimization



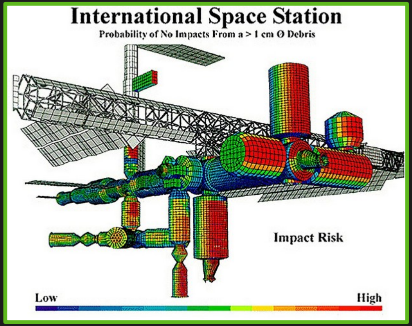
example of a convex function



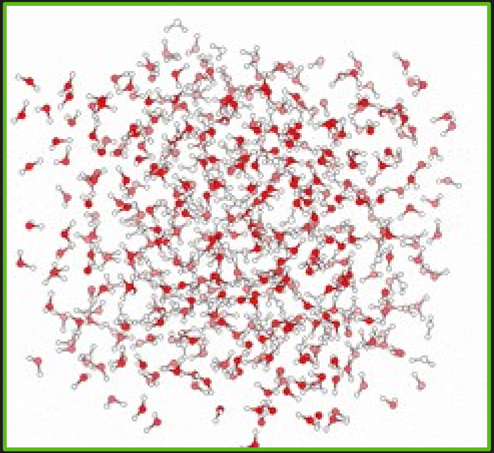
Applications:



Protein structure prediction (minimizing energy)



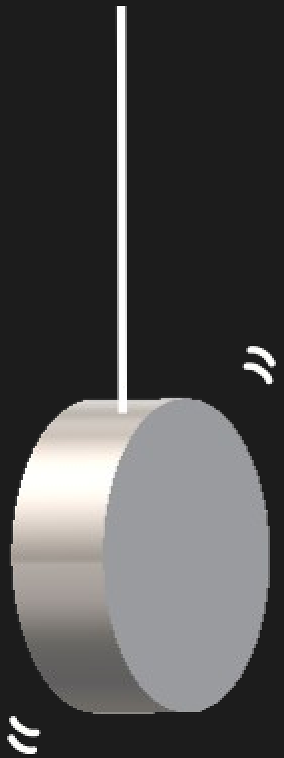
Safety engineering (provide acceptable levels of safety)



Molecular dynamics (initial optimization of the energy of the system to be simulated)

radiation therapy planning

...and much more!



...like Newtonian Noise!!!

… or GW detector physics: LINK

Global Optimization

Deterministic

You can have theoretical **guarantees** that the solution is indeed the **global minimum**

Stochastic

They involve **randomness** in the algorithm. They **cannot** provides any **guarantees** if the minimum is actually the global one.

Multiple runs

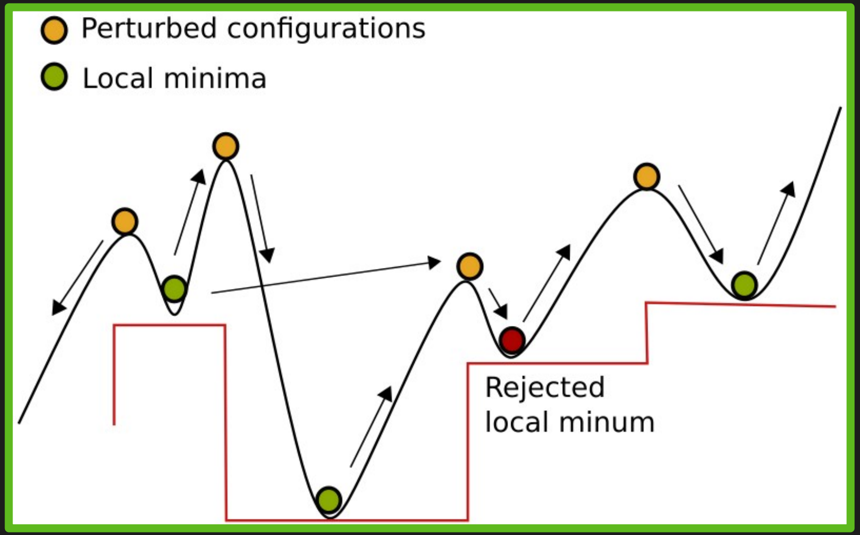
To escape local minima

1) Basin Hopping

2) Differential Evolution

3) Particle Swarm Optimization

Basin Hopping

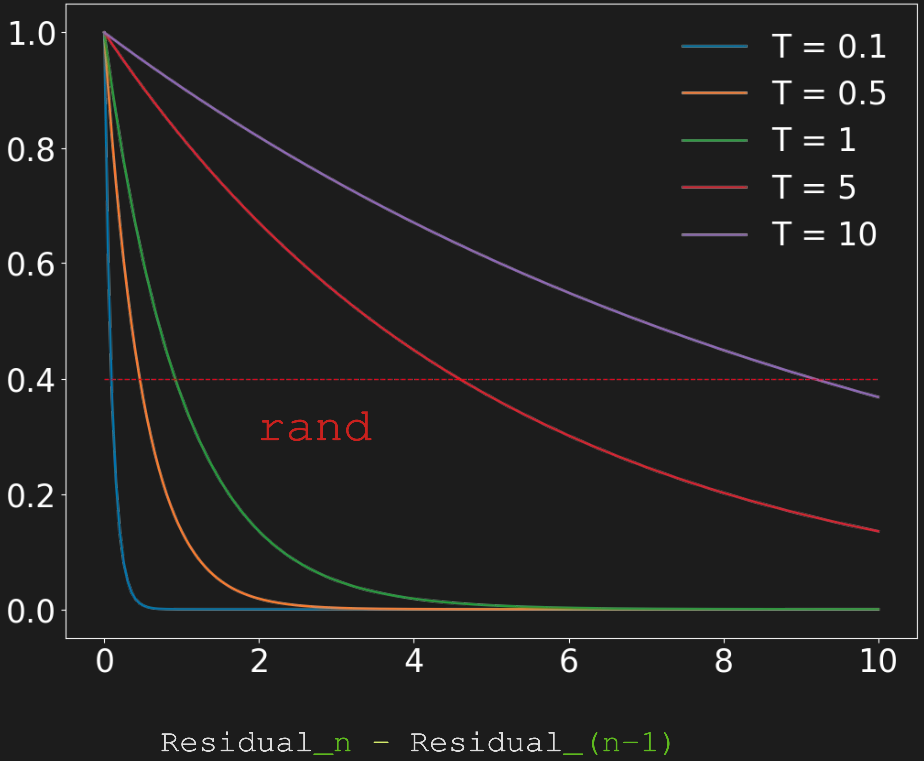


1) Perturbation
2) Local minimization
3) Acceptance/Rejection

Metropolis criterion:

Metropolis criterion:

Higher T → larger jumps
in the Residual will be
accepted



n^{th} step:

→ Residual_n

if: Residual_n < Residual_(n−1)
→ **accept** Residual_n
else:
if: $e^{-(\text{Residual\_n} - \text{Residual\_(n-1)})/T}$ >= rand:
→ **accept** Residual_n
else:
→ **reject** Residual_n

T → 0 => greedy algorithm

Curiosity fact:

Why T in

$$e^{-(Residual\_n\ -\ Residual\_(n-1))/T}\ ?$$

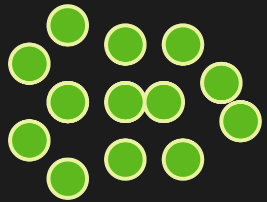
Annealing: It involves heating a material above its recrystallization temperature, maintaining a suitable temperature for an appropriate amount of time and then cooling.

In annealing, atoms migrate in the crystal lattice and the number of dislocations decreases, leading to a change in ductility and hardness. As the material cools it recrystallizes.

https://en.wikipedia.org/wiki/Annealing_(materials_science)

Differential Evolution

Evolutionary algorithm. They are inspired by the mechanisms of **biological evolution**: production, mutation, recombination and selection



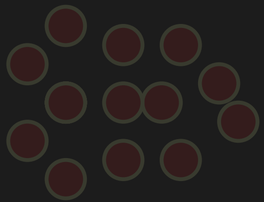
Generation G

Random starting population

(called **first generation**: it should cover the many possible points of the domain)

Mutation:

$i = j1 + \mathbf{F}(j2 - j3)$



F = mutation parameter

CR = crossover parameter

Crossover:

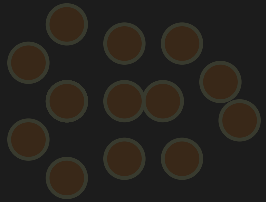
In order to increase the diversity of the perturbed parameter vectors, crossover is introduced.

If rand_n°$_j$ >= **CR** →

If rand_n°$_j$ < **CR** →

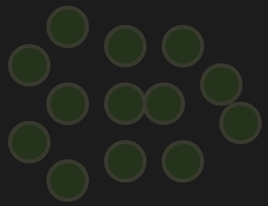
j = 1,…,D

D = dimensions of the individual (point in the D-dimensional domain)



Differential Evolution

Evolutionary algorithm. They are inspired by the mechanisms of **biological evolution**: production, mutation, recombination and selection

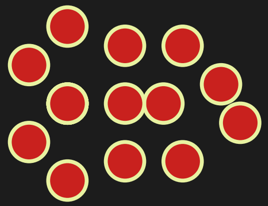


Generation G

Random starting population
(called **first generation**: it should cover the many possible points of the domain)

Mutation:

$$i = j1 + \mathbf{F}(j2 - j3)$$

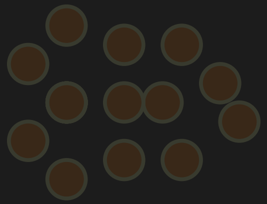


F = mutation parameter
CR = crossover parameter

Crossover:
In order to increase the diversity of the perturbed parameter vectors, crossover is introduced.

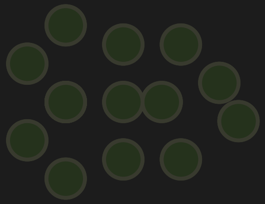
If rand_n°$_j$ >= **CR** →
If rand_n°$_j$ < **CR** →

j = 1,…,D
D = dimensions of the individual (point in the D-dimensional domain)



Differential Evolution

Evolutionary algorithm. They are inspired by the mechanisms of **biological evolution**: production, mutation, recombination and selection

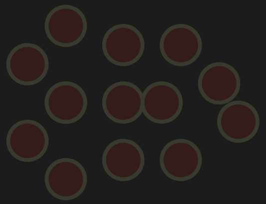


Generation G

Random starting population
(called **first generation**: it should cover the many possible points of the domain)

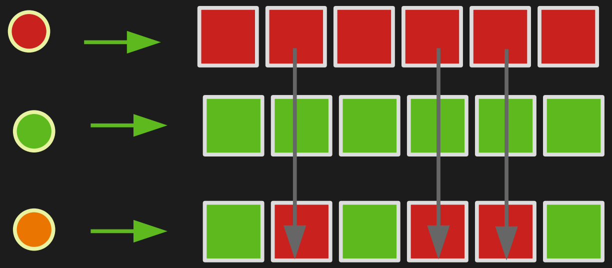
Mutation:

$i = j1 + F(j2 - j3)$



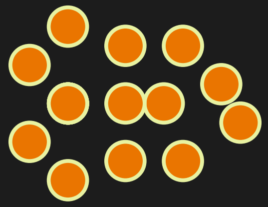
F = mutation parameter
CR = crossover parameter

Crossover:
In order to increase the diversity of the perturbed parameter vectors, crossover is introduced.



If rand_n°$_j$ >= **CR** →
If rand_n°$_j$ < **CR** →

j = 1,…,D
D = dimensions of the individual (point in the D-dimensional domain)

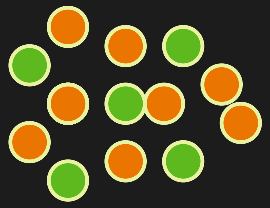


Differential Evolution

Selection:

To decide if can be part of the generation G+1 we use a **greedy criterion:**

If Residual(i) < Residual(i) → we can keep it, otherwise it will rejected and i will be kept instead.



Generation G+1

...then the loop start again with G+1 and so on, for a defined number of steps (or it can stop before if it reaches some stopping criterion: |Residual_n - Residual_(n-1)| <= min_error).

Particle Swarm

Not genes… but bird flocks

"Tra le rossastre nubi
stormi d'uccelli neri,
com'esuli pensieri,
nel vespero migrar".

G. Carducci, San Martino

"Between reddish clouds
black bird flocks,
like exiled thoughts,
in the eventide migrate".

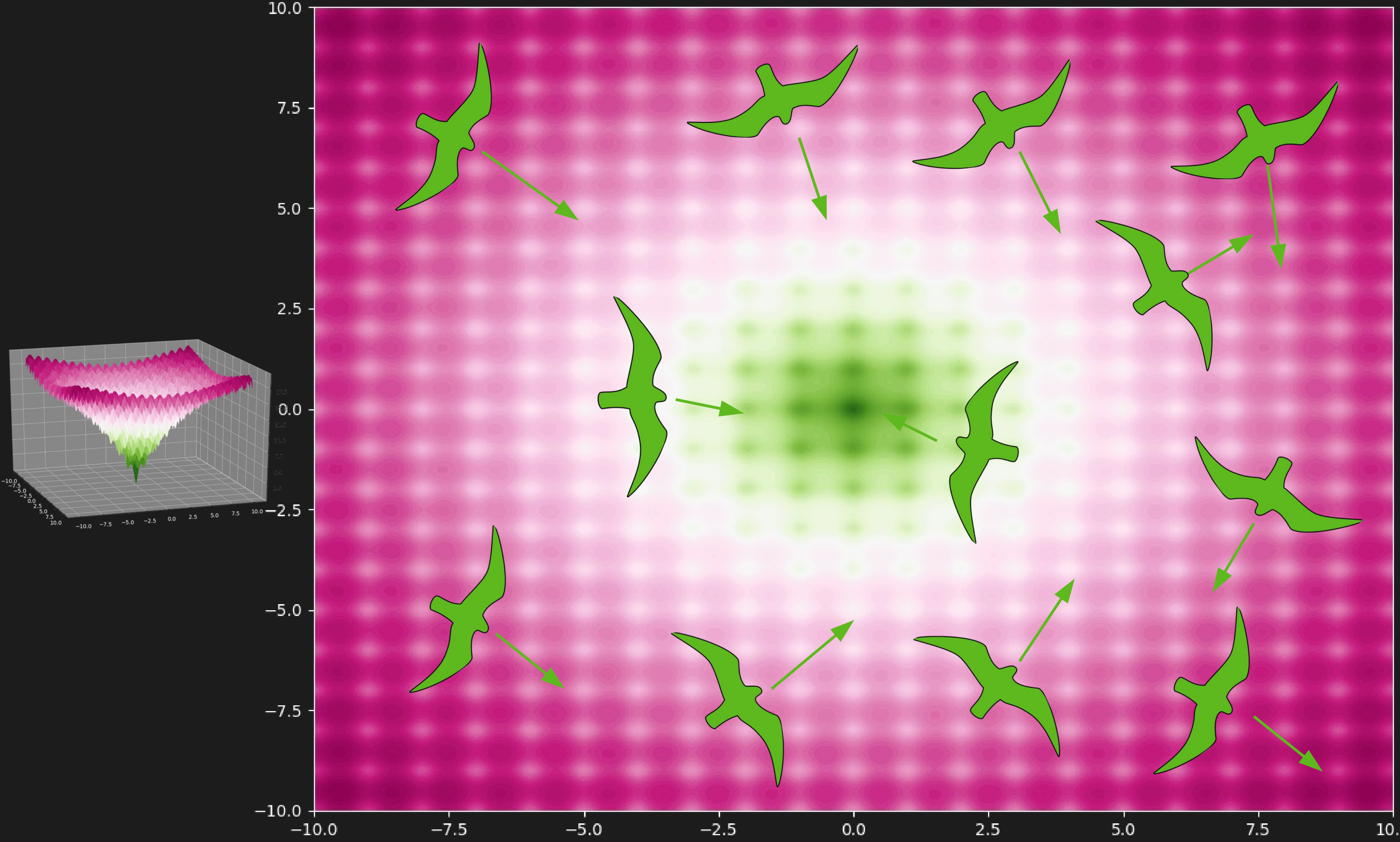
G. Carducci, San Martino

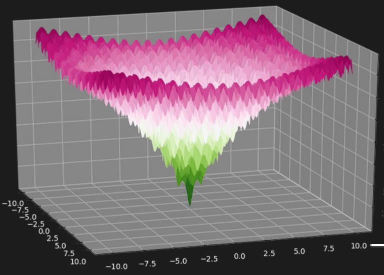
Particle Swarm

Curiosity fact: Particle swarm optimization arose in the context of simulating the ability of **human society** to **improve** its **knowledge**.

Psychological assumption:

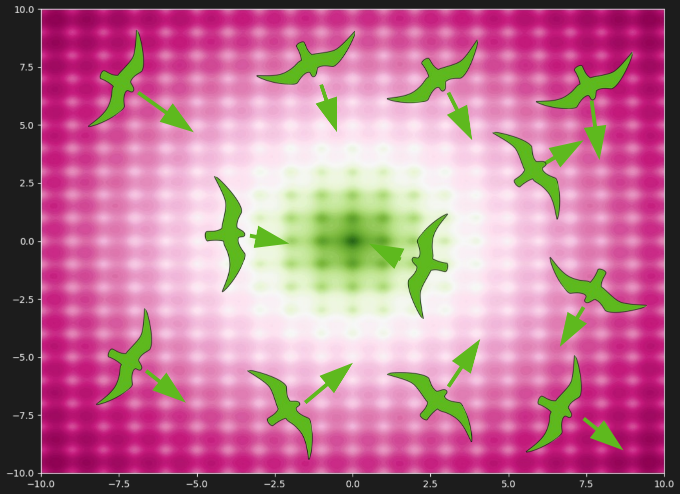
1) individual behaviour:
   individuals → follow the best beliefs in their experience.
2) social behaviour:
   individuals → also consider beliefs of others (if these are proved to be better than their own beliefs).


10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0




10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

Position of the i^{th} bird (particle) at the n^{th} step:



$$P^i_n = (x_0, x_1, \ldots, x_D)$$

and its velocity:

$$V^i_n = (v_0, v_1, \ldots, v_D)$$

The next position will be:

$$P^i_{n+1} = P^i_n + V^i_{n+1}$$

each particle memorizes its best **personal solution** and the **best global solution** (as if they were able to **communicate**)

$$V^i_{n+1} = \boxed{\mathbf{I}\ V^i_n} + \boxed{\mathbf{C}\,(P^i_{best} - P^i_n)} + \boxed{\mathbf{S}\,(P^{global}_{best} - P^i_n)}$$

Inertia **Individual behaviour (cognitive)** **Social behaviour**

Let's go back to the optimization for the Newtonian noise:

Exercise

link

Don't hesitate to contact me:
francesca.badaracco@uclouvain.be

Newtonian Noise Cancellation Strategies and Optimization Problems

Newtonian noise:

- Wiener filter
- Optimization algorithms
- Gaussian Processes

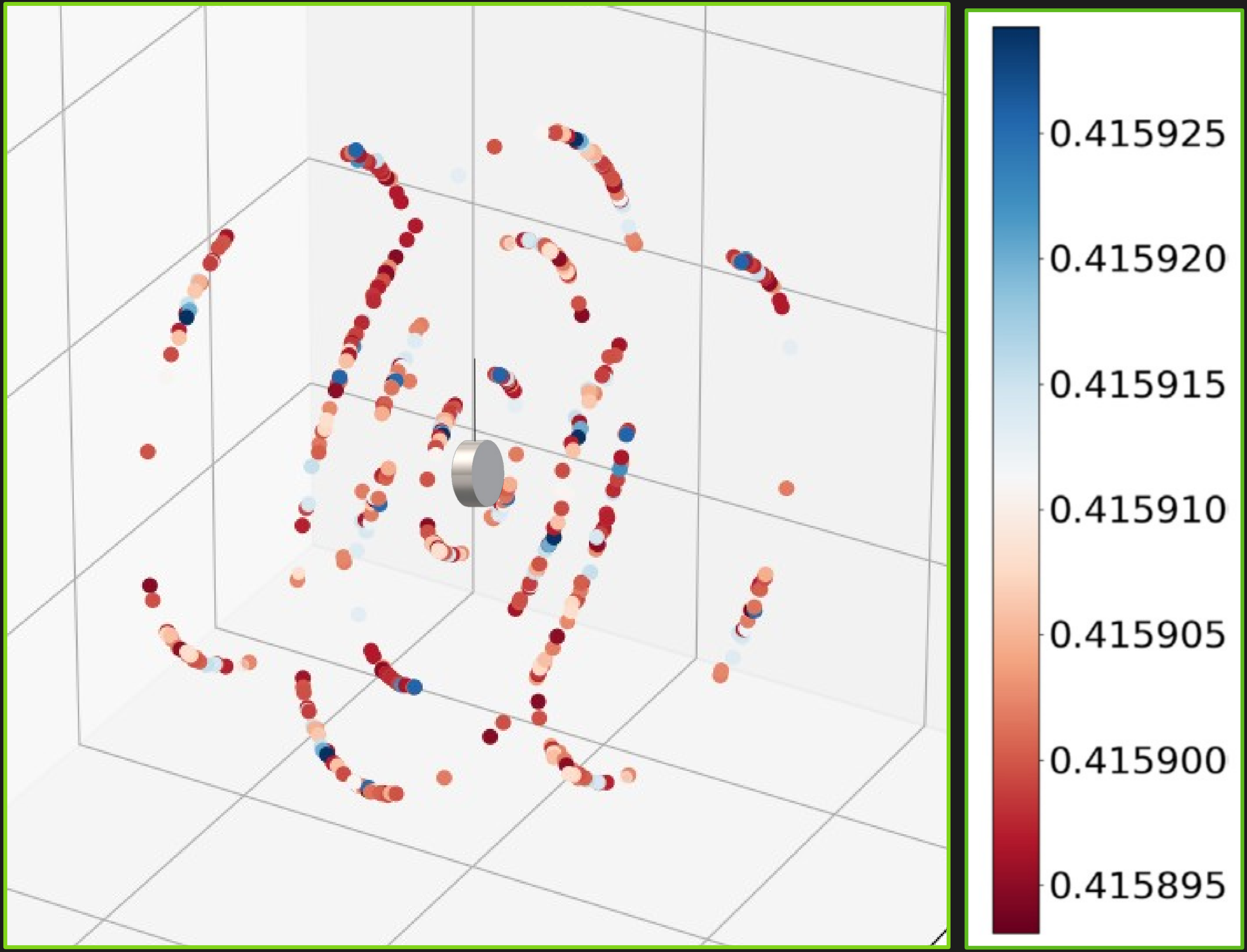
Let's go back to the optimization for the Newtonian noise:

$$R(\omega) = 1 - \frac{\vec{C}_{sn}^{\dagger}\mathbf{C}_{ss}^{-1}\vec{C}_{sn}}{C_{nn}}$$

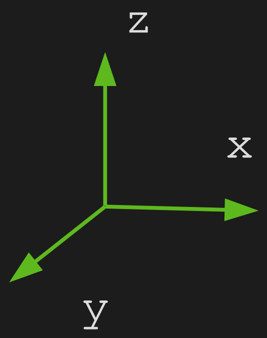
Isotropic and homogeneous seismic field for underground detectors.

All the **100 optimizations**

For arrays with **N = 6** seismometers each.



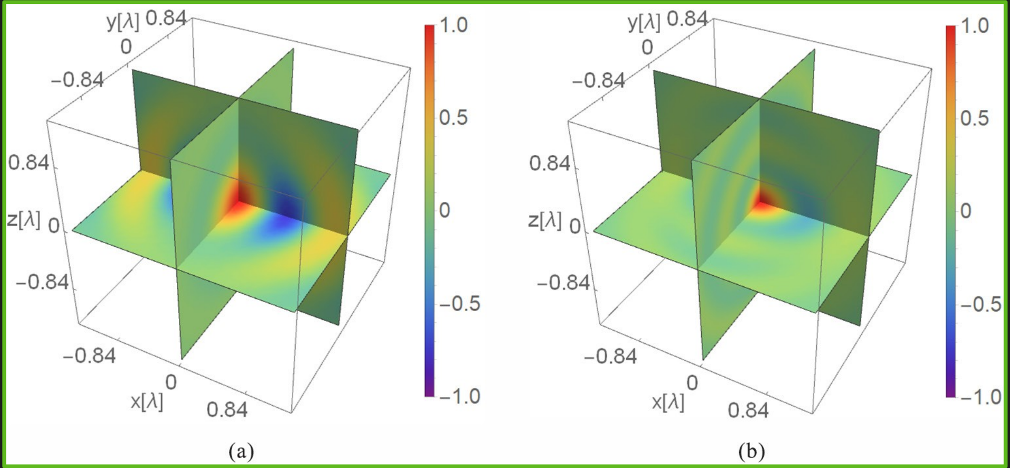
Underground case: we need to consider all the 3 directions of the seismic displacement:



Limited by P and S waves mixing:

Only P waves

Mixed: P and S



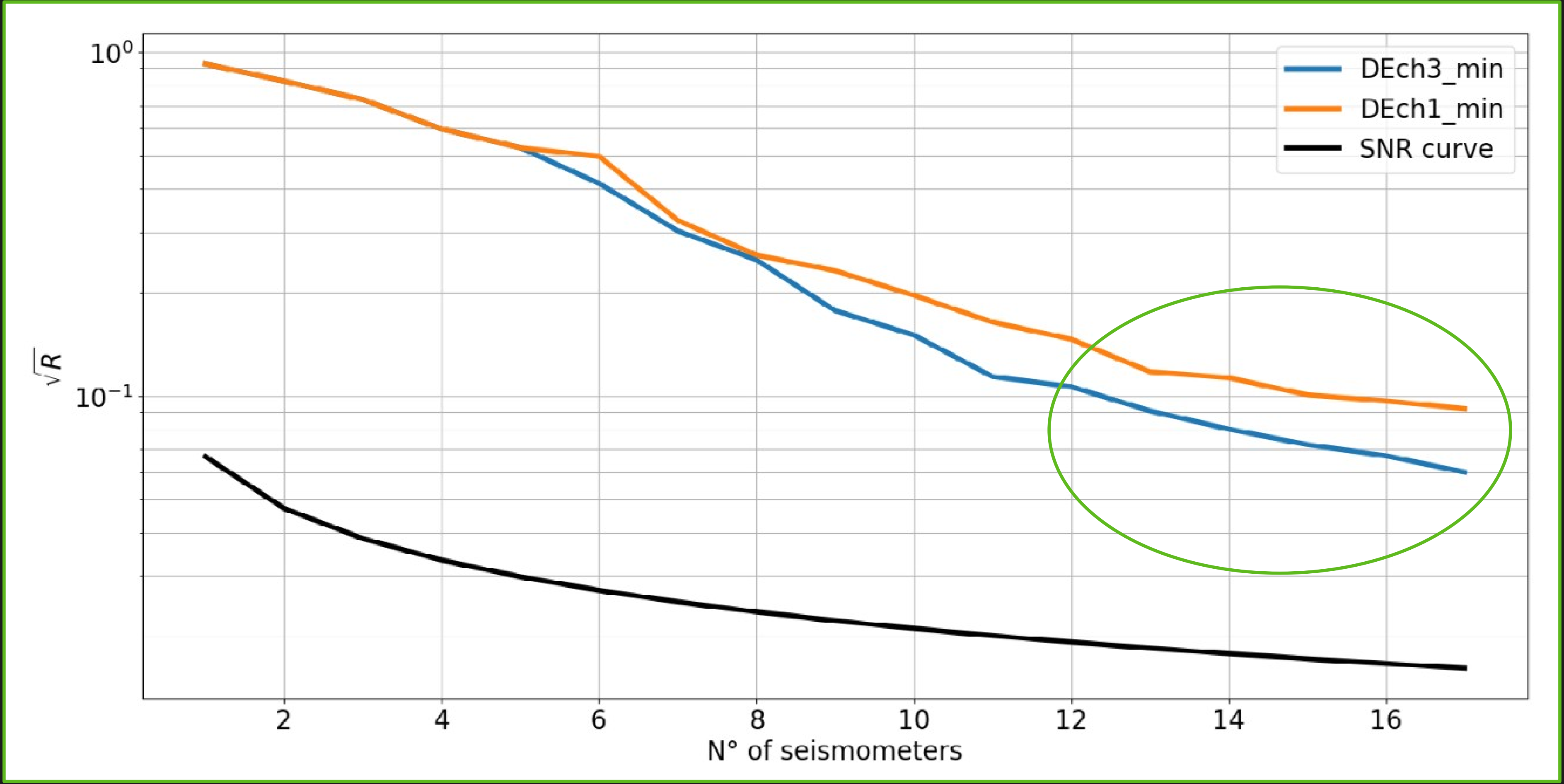
(a) (b)

Correlation of the seismometer in the origin with all the other points in a homogeneous and isotropic field.

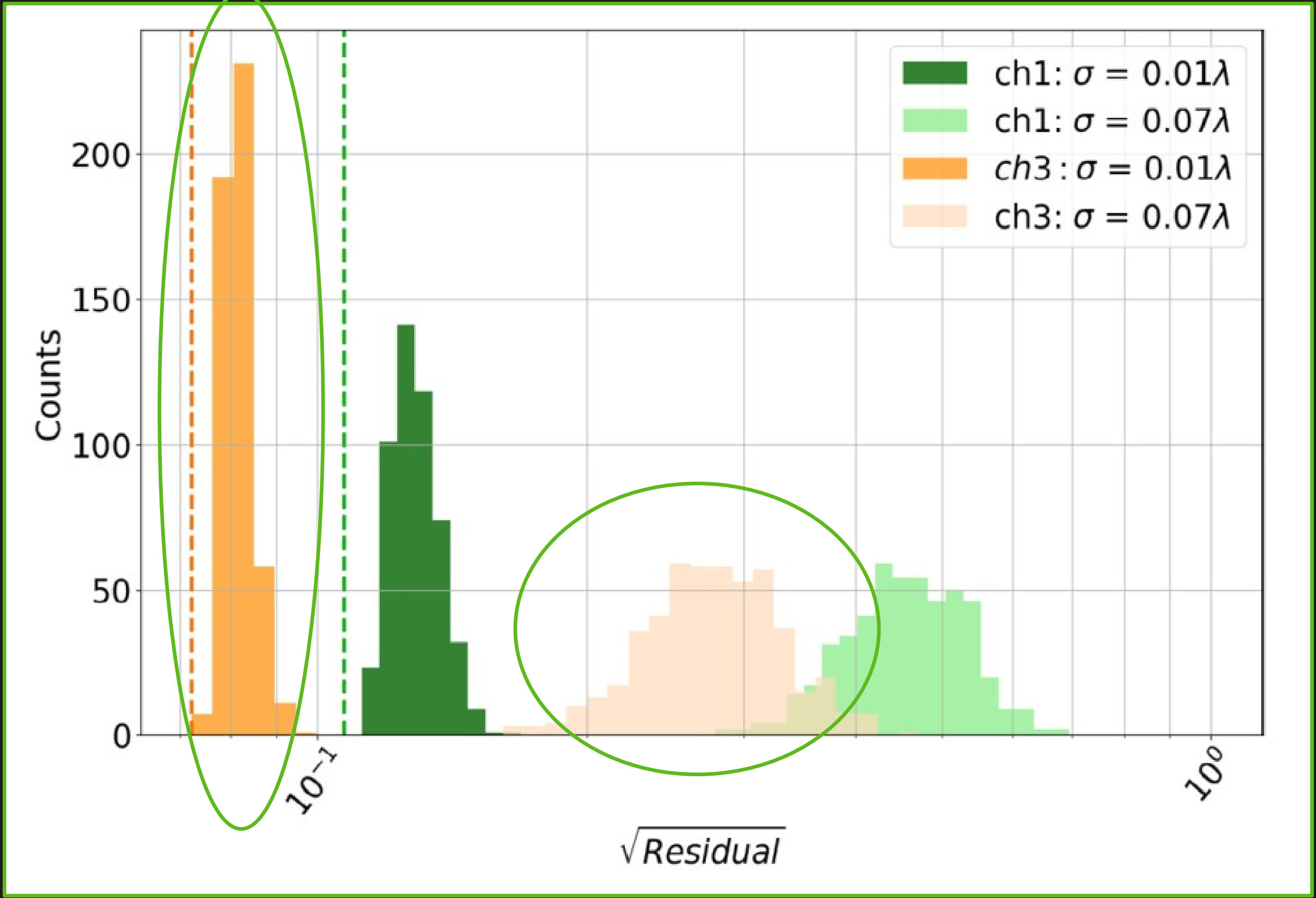
Remember:
P = compressional waves (always generate NN)
S = shear waves (**usually** don't generate NN)

Because of their **different propagation velocity** in the ground, P and S waves produce two-point correlations that are out of phase, thus affecting the configuration of the optimal array.

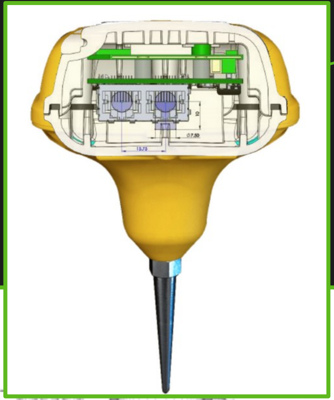
The more, the better:

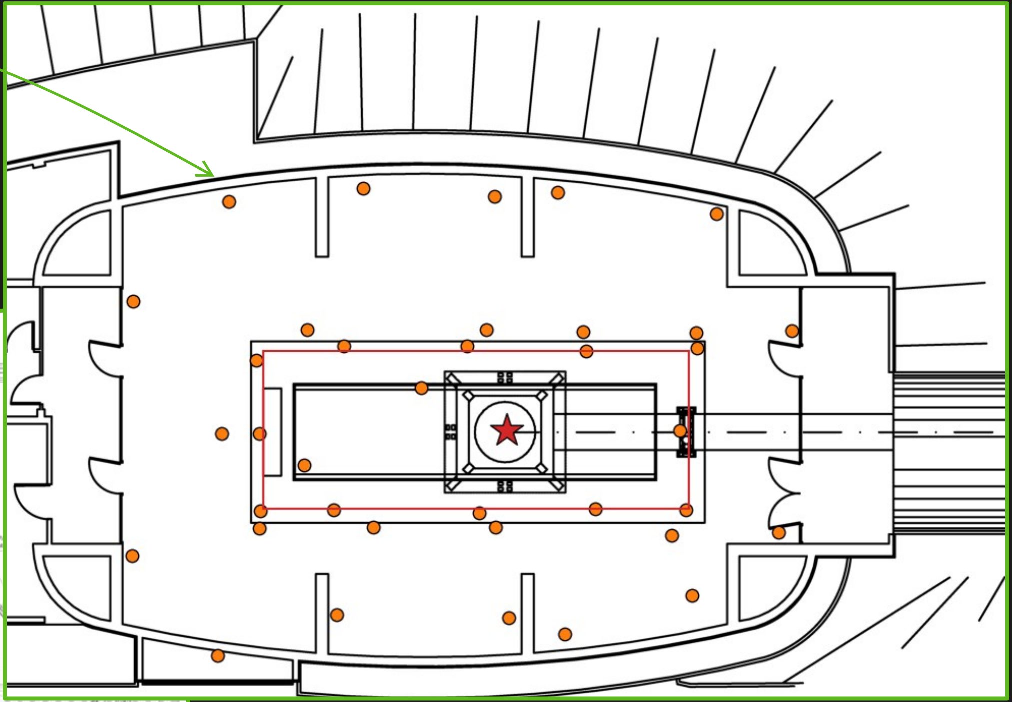


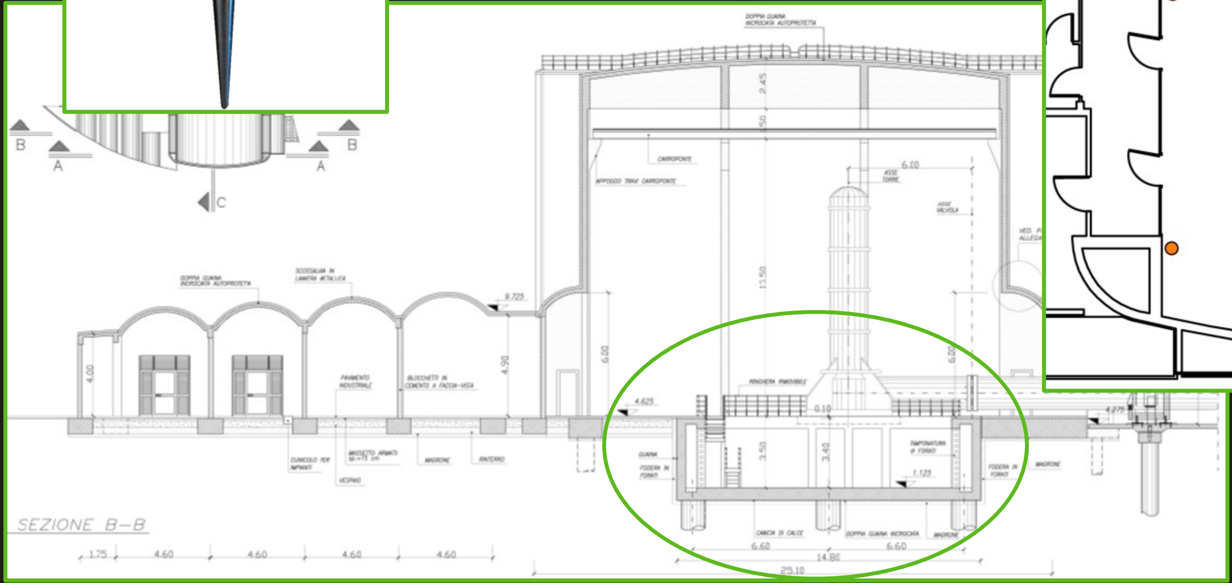
We might misplace the sensors, then what…?



What if the seismic field is not homogeneous and isotropic?







What if the seismic field is not homogeneous and isotropic?

Residual in frequency domain

$$R(\omega) = 1 - \frac{\vec{C}_{sn}^{\dagger} \mathbf{C}_{ss}^{-1} \vec{C}_{sn}}{C_{nn}}$$

$$C_{sn_i}(\omega) = E[s_i^*(\omega) n(\omega)]$$

We can use a model (next slide)

$$C_{nn}(\omega) = E[n^*(\omega) n(\omega)]$$

We treat it just as a **unknown** constant

$$C_{ss_{ij}}(\omega) = E[s_i^*(\omega) s_j(\omega)]$$

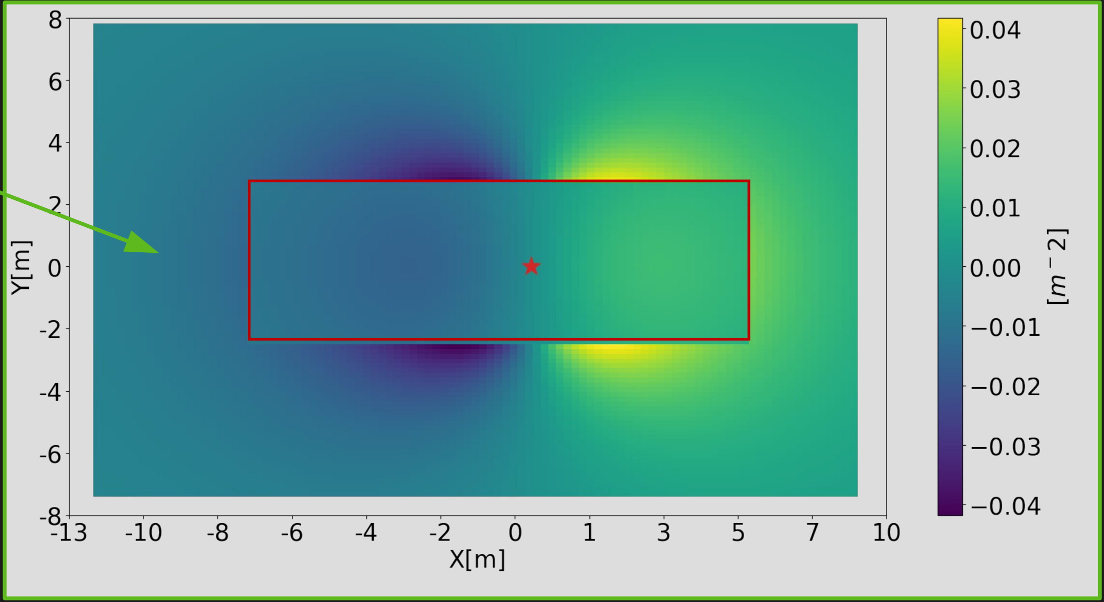
This is easy: we just need to **collect data**!

What if the seismic field is not homogeneous and isotropic?

$$C_{sn}(\mathbf{r},\mathbf{r}_0) = \mathcal{C}\int C_{ss}(\mathbf{r},\mathbf{r}_1)\frac{x_0 - x}{(h(\mathbf{r}_1)^2 + |\mathbf{r}_1 - \mathbf{r}_0|^2)^{3/2}}d\mathbf{r}_1$$

$$R(\omega) = 1 - \frac{\vec{C}_{sn}^{\dagger}\mathbf{C}_{ss}^{-1}\vec{C}_{sn}}{C_{nn}}$$

$$C_{sn}(\mathbf{r},\mathbf{r}_0) = \mathcal{C}\int C_{ss}(\mathbf{r},\mathbf{r}_1)\mathcal{K}(\mathbf{r}_1,\mathbf{r}_0)d\mathbf{r}_1$$



In the end, we only need to know this (and we can have it from data)

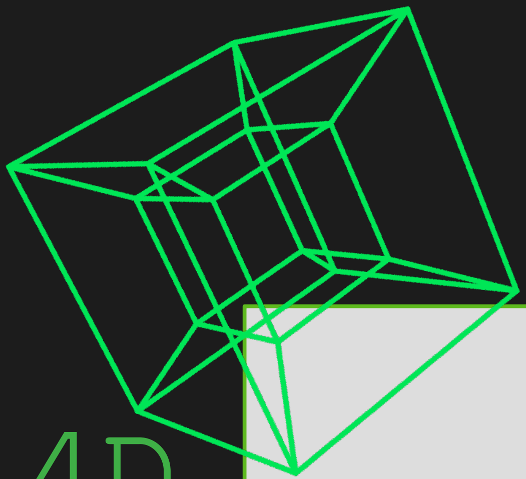
$C_{ss}(x_1, y_1, x_2, y_2)$ is a 4D function!

Curse of dimensionality:

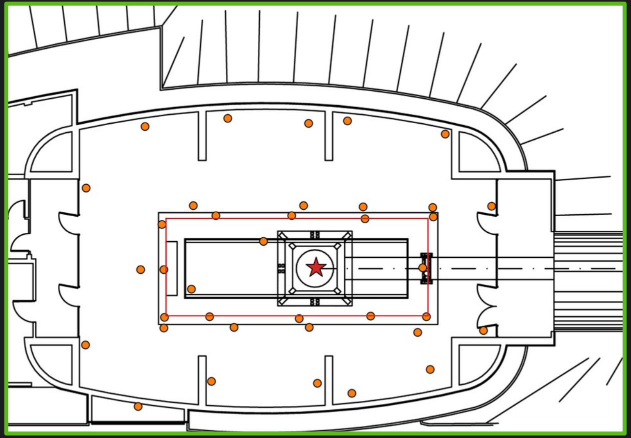
$\rho_{2D} = 0.30$

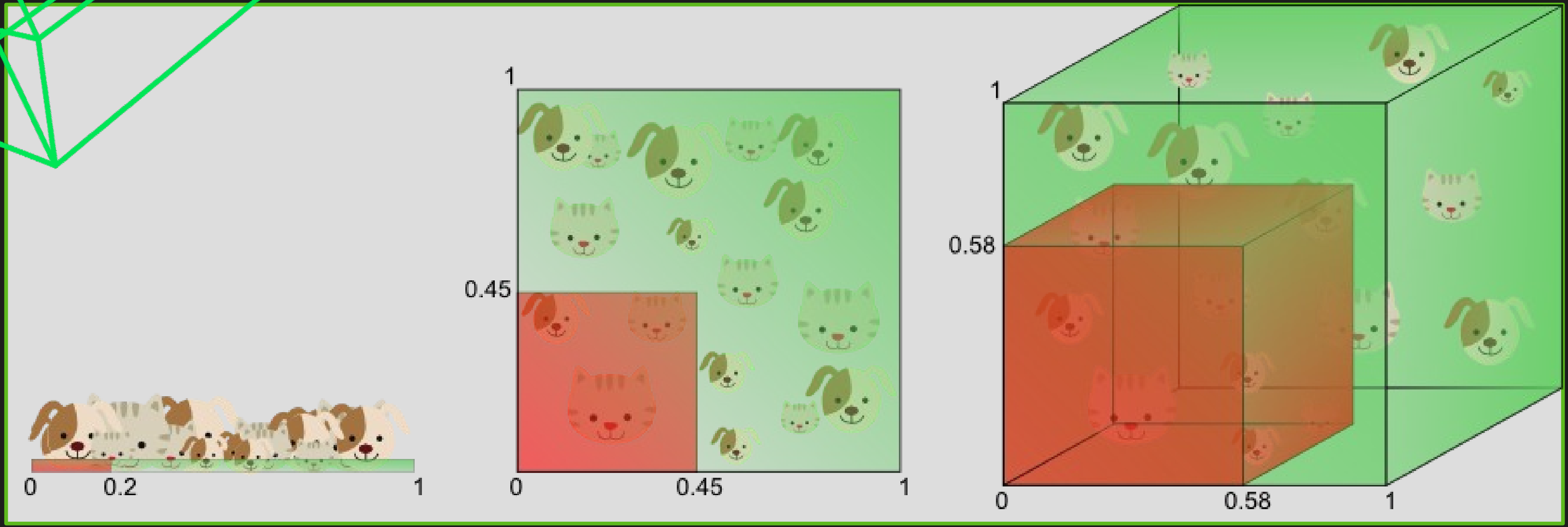
$\rho_{4D} = 0.05$

$\rho_{4D\_Regular\_grid} = 29$



4D





https://www.visiondummy.com/2014/04/curse-dimensionality-affect-classification/

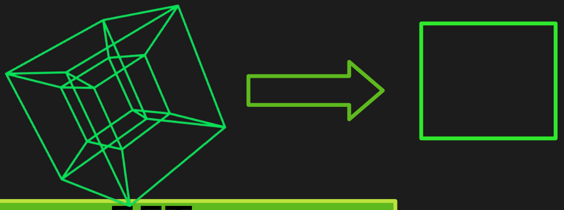
$$C_{ss}(x_1, y_1, x_2, y_2) = \langle (FFT^*\{s(x_1, y_1)(\omega)\}\ FFT\{s(x_2, y_2)(\omega)\}) \rangle$$

i^{th} seismometer's data stream (1 hour, for example)

FFT_1

FFT_2

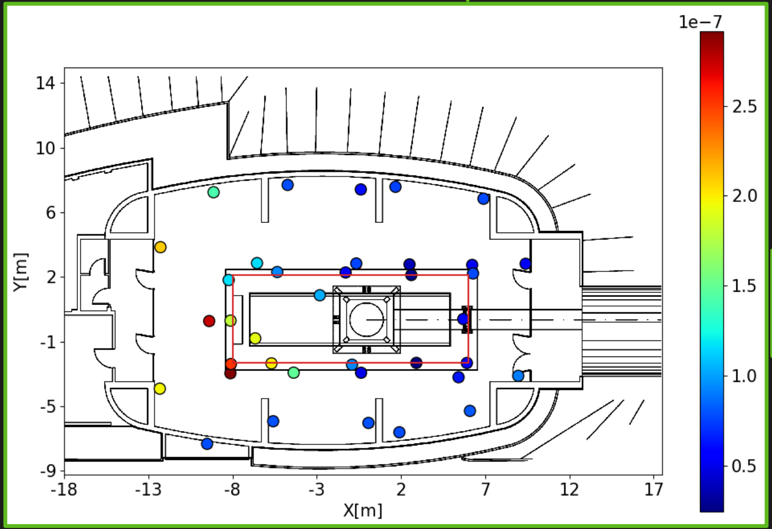
N segments with
50% overlapping



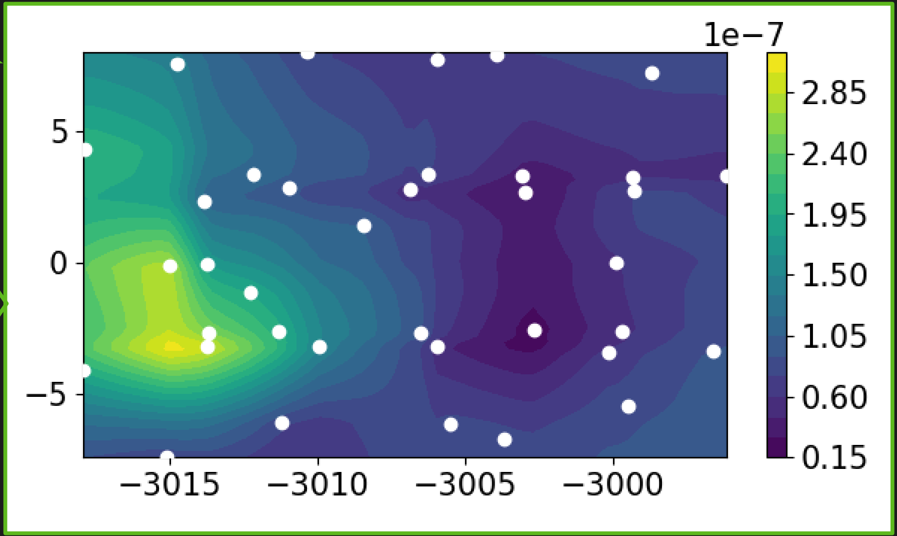
. . .

FFT_{N-1}

FFT_N



Gaussian Process

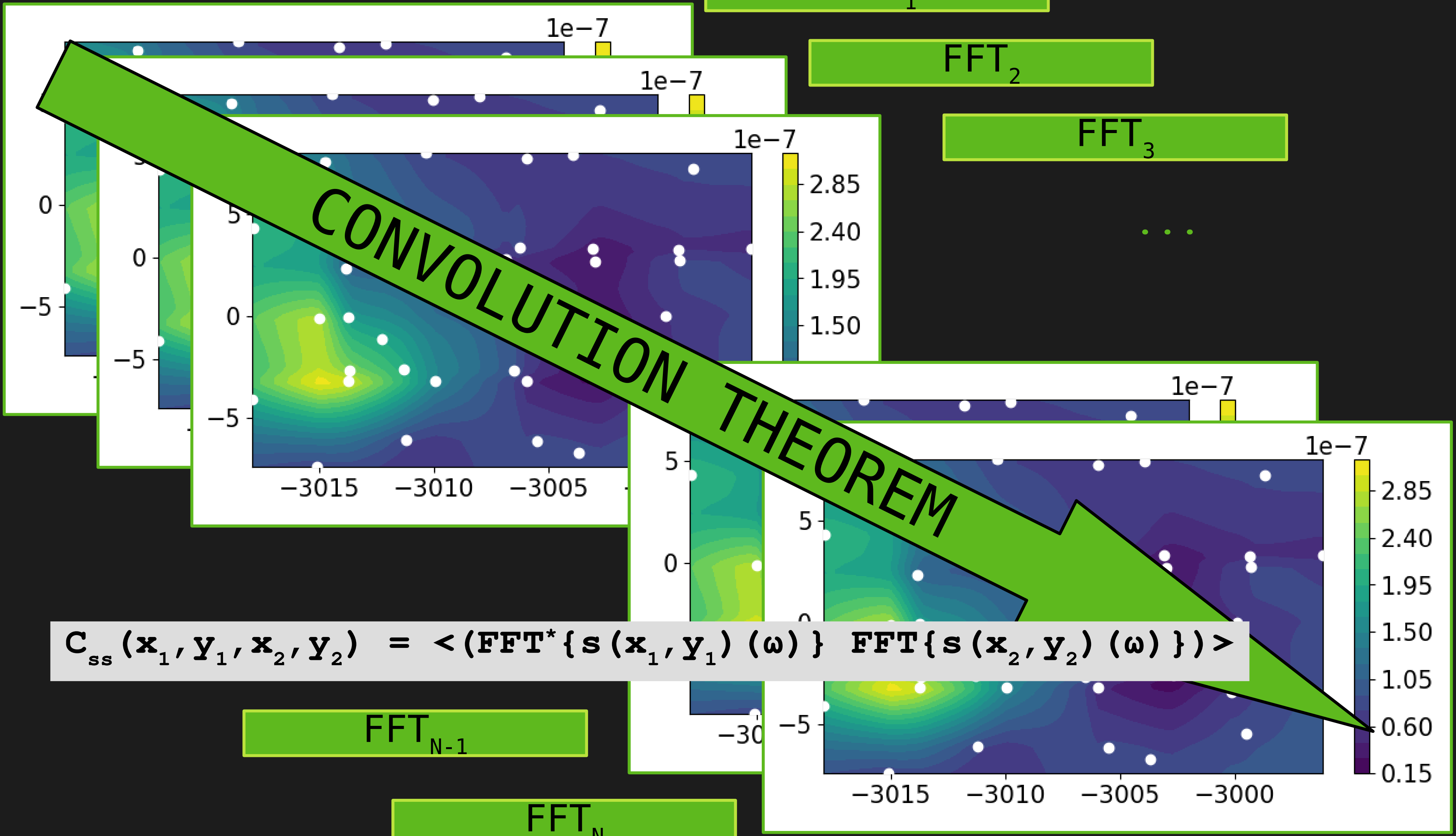


FFT$_1$

FFT$_2$

FFT$_3$

. . .

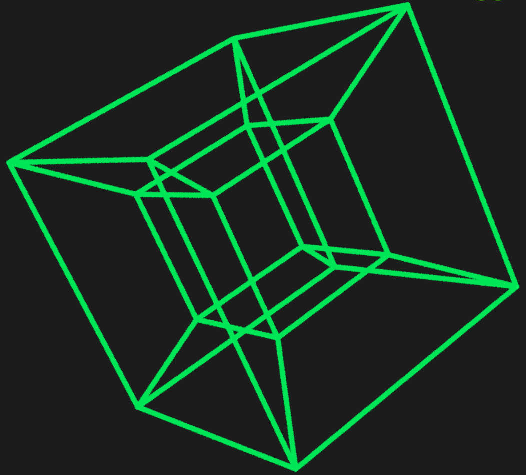


$$C_{ss}(x_1,y_1,x_2,y_2) = \langle (FFT^*\{s(x_1,y_1)(\omega)\}\ FFT\{s(x_2,y_2)(\omega)\}) \rangle$$

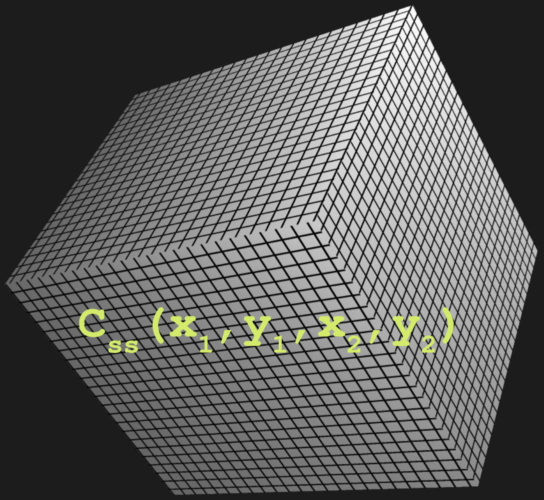
FFT$_{N-1}$

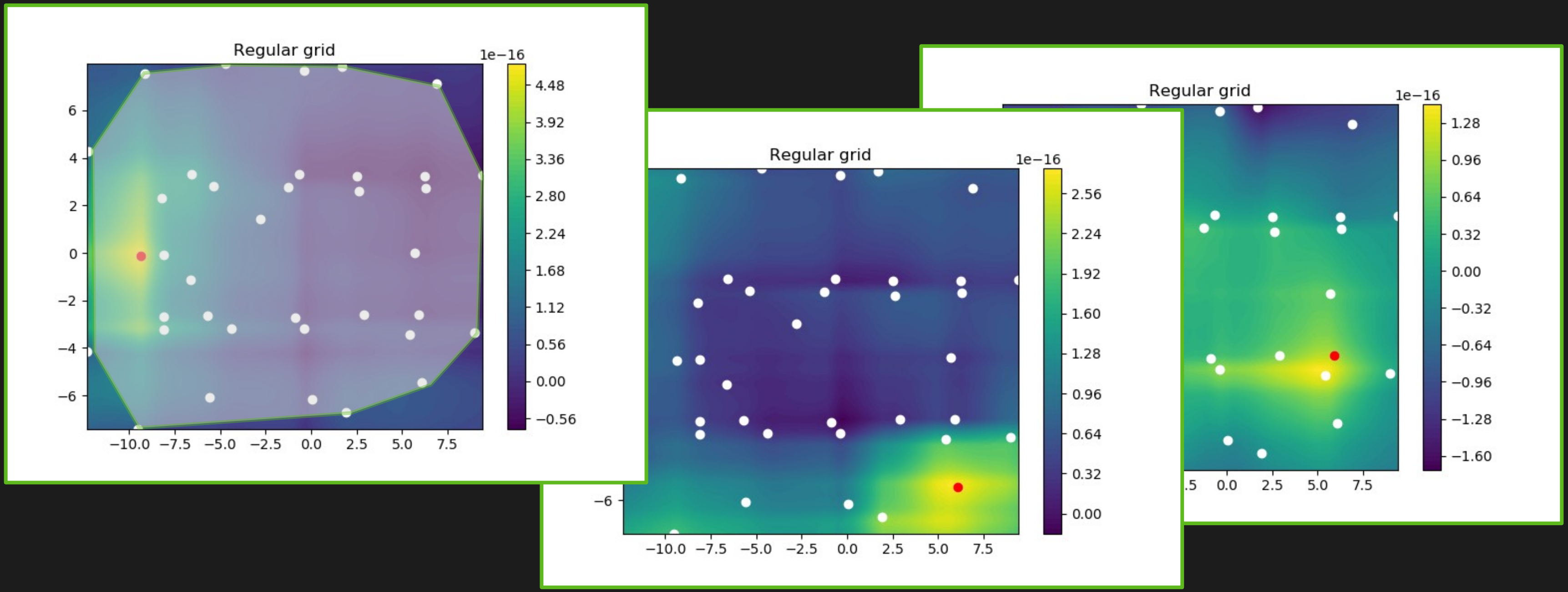
FFT$_N$

Every point of $C_{ss}(x_1,y_1,x_2,y_2)$ in the 4D space is calculated as before → We can virtually sample as many values of $C_{ss}(x_1,y_1,x_2,y_2)$ as we want, **wherever** we want.



Virtual Sampling +
Linear interpolation:
we created a **surrogate model** of $C_{ss}(x_1,y_1,x_2,y_2)$

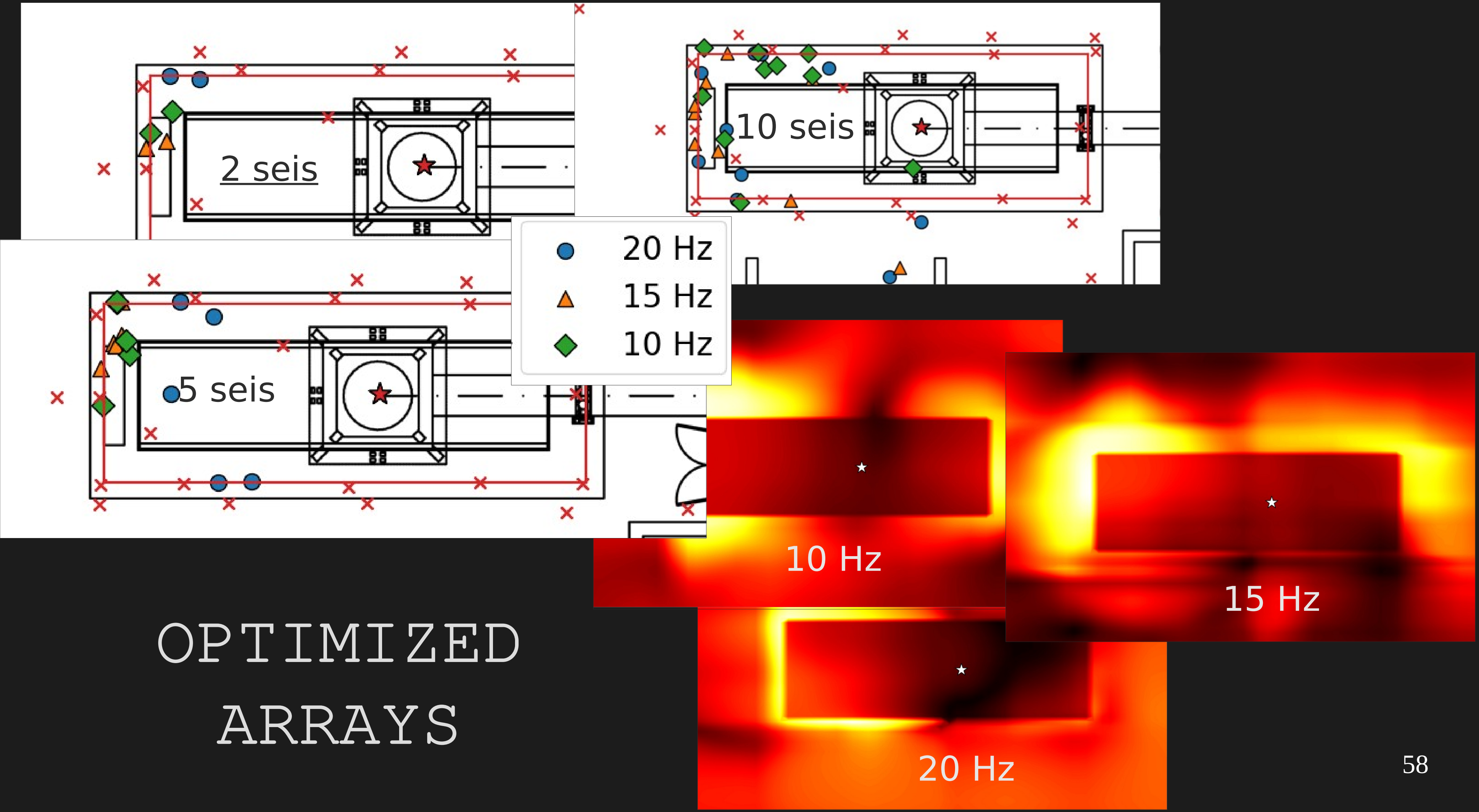




1) FFT of 37 seismometers' data (seismic displacement) → **2D gaussian process** at a frequency f_0: **Convolution theorem** → **surrogate model** of Css:

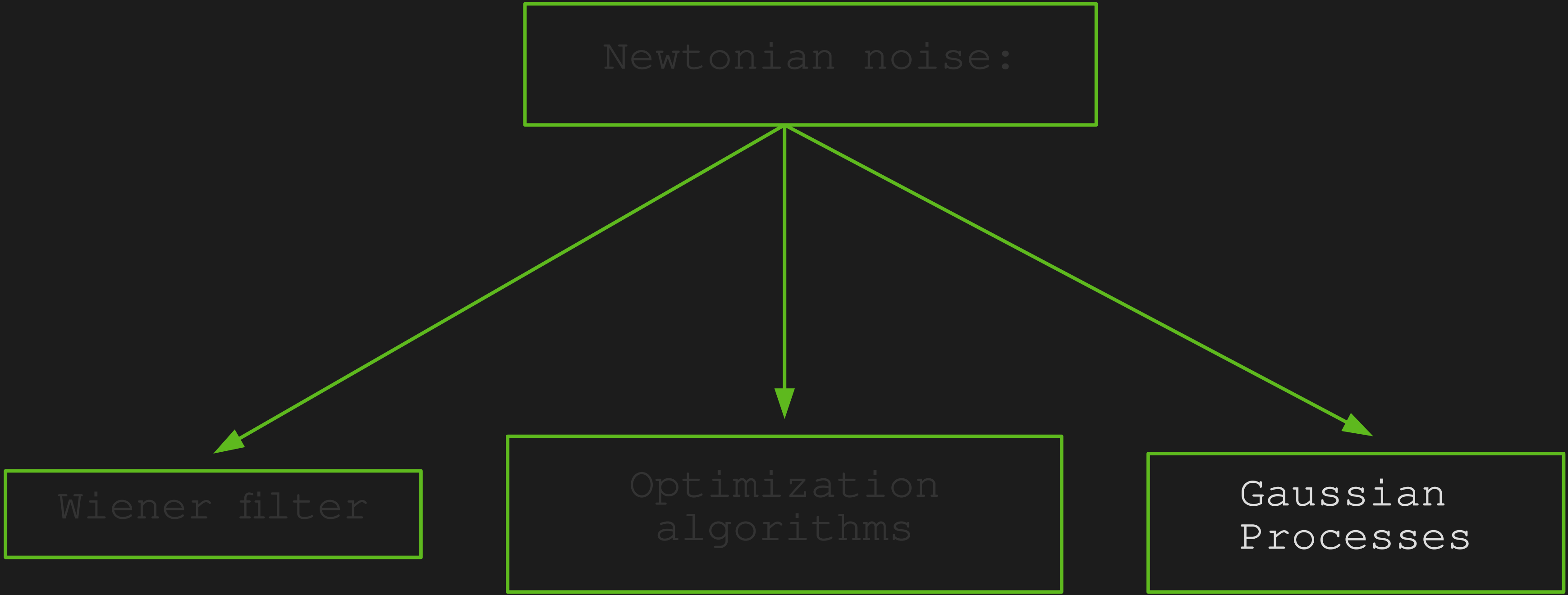
$$\mathbf{C_{ss}(x_1,y_1,x_2,y_2) = \langle (FFT^*\{s(x_1,y_1)(\omega)\}\ FFT\{s(x_2,y_2)(\omega)\}) \rangle}$$

2) Css Sampling → **4D Linear Interpolation on a Regular grid** (faster) → **Css** & **Csn** (integrated with Simpson method)



OPTIMIZED ARRAYS

Newtonian Noise Cancellation Strategies and Optimization Problems



What is a Gaussian Process?

Statistic →
Inferring models from data →
Interpretation

Machine Learning →
Learn algorithms to **predict** new data →
Black boxes

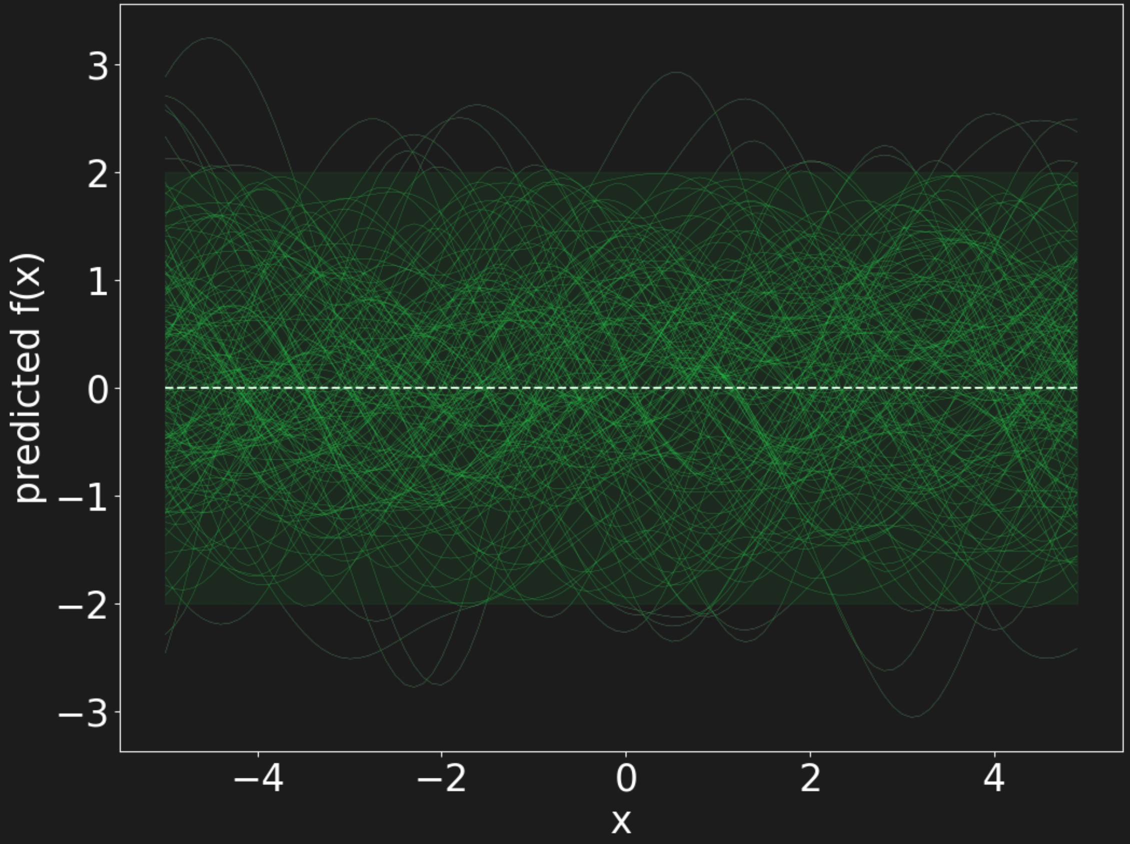
Gaussian Processes →
mathematically equivalent to **known models**
+
Learn from data and can **predict** new values

regression

Kriging
(geophysics)

classification

What is a Gaussian Process?



σ_f = signal variance, l = scale

σ_ε= noise variance

Gaussian process = a collection of random variables with a joint Gaussian distribution.

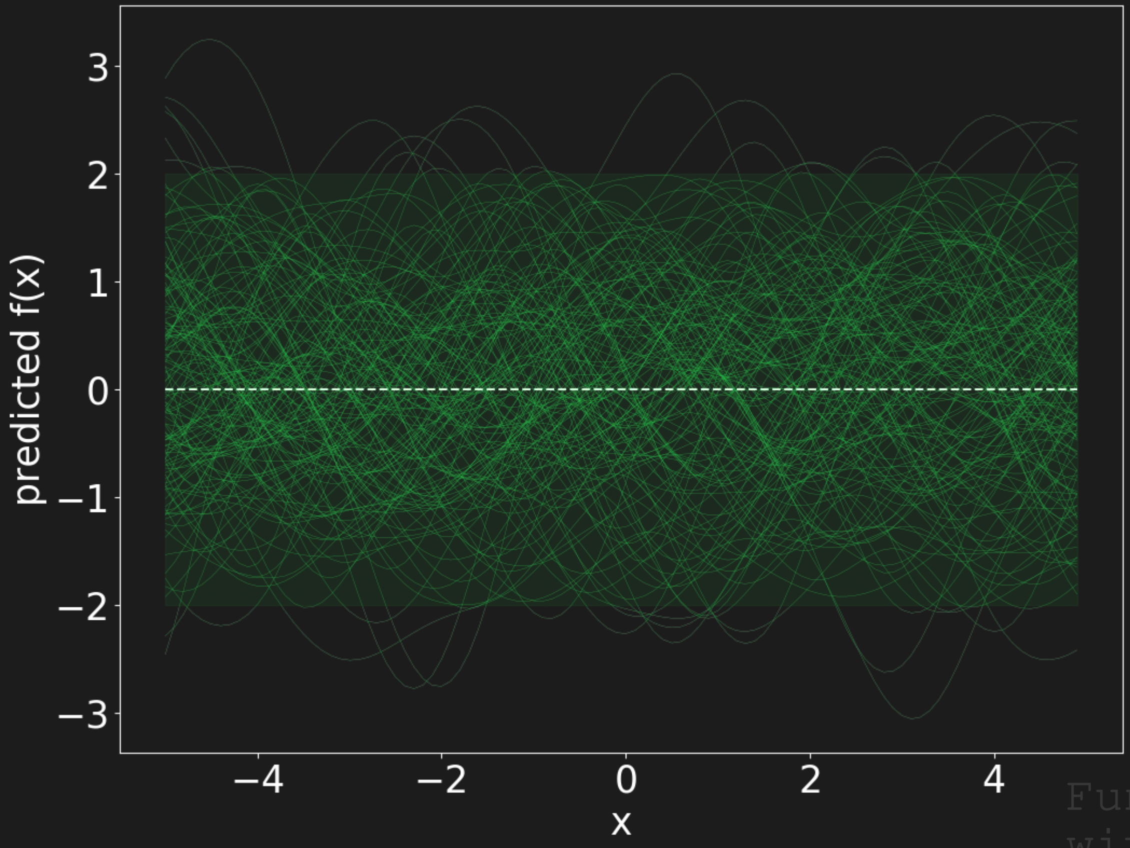
Gaussian process over functions = the values taken by a function in a point x_i: $f(x_i) = f_i$ are random variables.

$x_1, x_2, \ldots, x_N \rightarrow f_1, f_2, \ldots, f_N$ with a gaussian joint distribution with mean and covariance:

$$m(x) = E[f(x)]$$

$$k(x, x') = E[(f(x) - m(x))(f(x') - m(x'))]$$

What is a Gaussian Process?



We can draw functions from a multivariate normal distribution:

$$f(x) \sim \mathcal{GP}(m(x), k(x, x'))$$

GP regression takes the form of a Bayesian inference over a "latent function", f(x):

$$y = f(x) + \varepsilon$$

Gaussian distributed noise

$$f(x) \sim \mathcal{N}(0, k(x, x'))$$

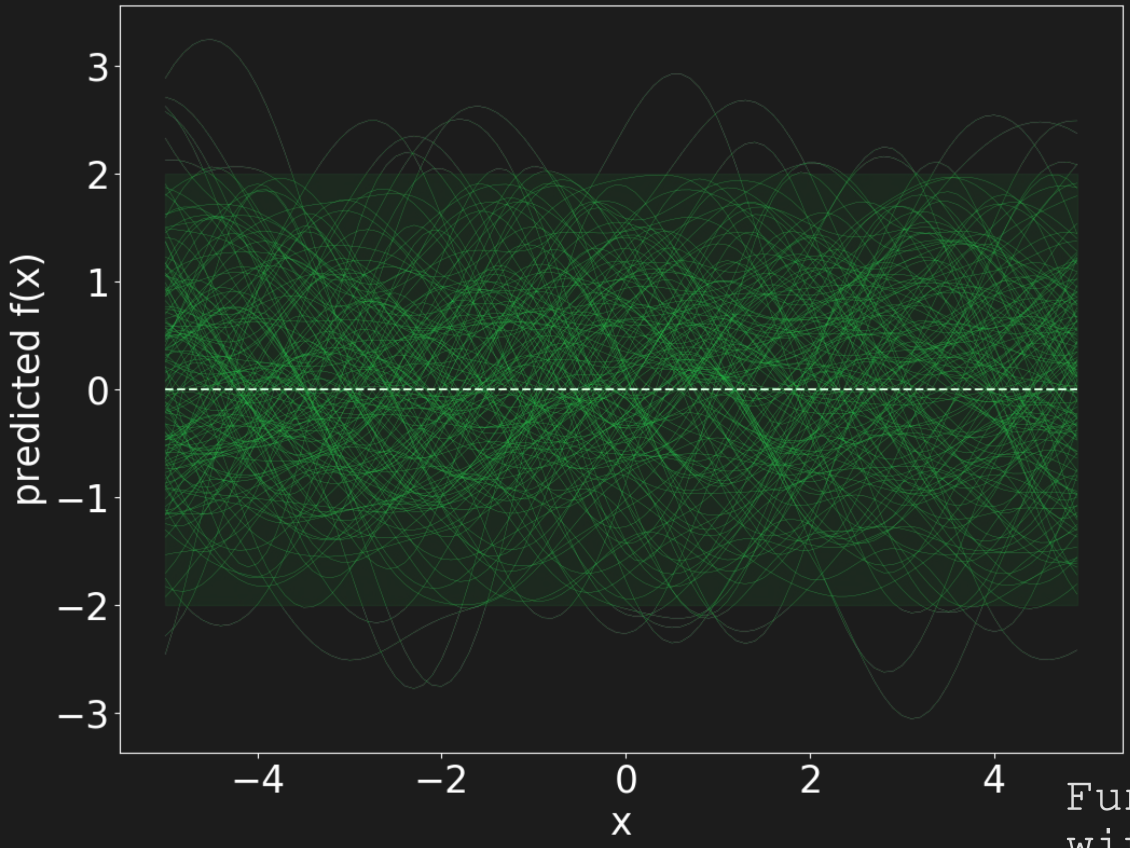
$$k(x_i, x_j) = \sigma_f^2 e^{-\frac{(x_i - x_j)^2}{2l}} - \sigma_\varepsilon \delta_{ij}$$

σ_f = signal variance, l = scale

σ_ε= noise variance

Functions f(x) sampled by a prior with fixed hyper-parameters: σ_f = 1, l = 0 and σ_ε= 0 and zero mean.

What is a Gaussian Process?



We can draw functions from a multivariate normal distribution:

$$f(x) \sim \mathcal{GP}(m(x), k(x, x'))$$

GP regression takes the form of a Bayesian inference over a ”latent function”, f(x):

$$y = f(x) + \varepsilon$$

Gaussian distributed noise

$$f(x) \sim \mathcal{N}(0, k(x, x'))$$

$$k(x_i, x_j) = \sigma_f^2 e^{-\frac{(x_i - x_j)^2}{2l}} - \sigma_\varepsilon \delta_{ij}$$

σ_f = signal variance, l = scale

σ_ε= noise variance

Functions f(x) sampled by a prior with fixed hyper-parameters: σ_f = 1, l = 0 and σ_ε= 0 and zero mean.

Prior for the distribution of values f(x):

$$f(x) \sim \mathcal{GP}(m(x), k(x, x'))$$

Conditioning with observed data:

$$f_* | x_*, \mathbf{x}_0, \mathbf{y}_0 \sim \mathcal{N}(\mu_*, \sigma_*)$$

Predicted value in x_*

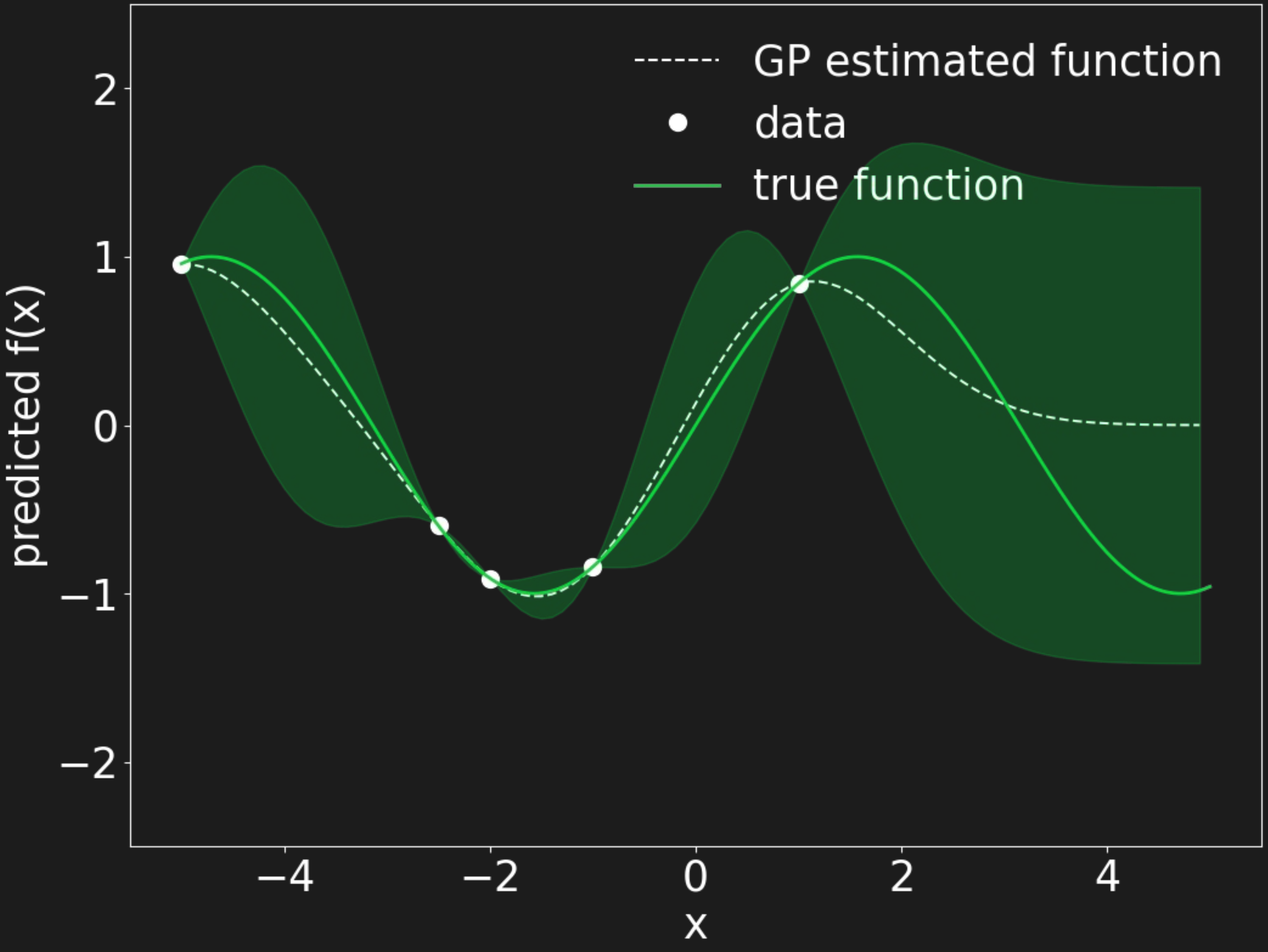
$$\mu_* = k(x_*, \mathbf{x}_0)^T (k(\mathbf{x}_0, \mathbf{x}_0) + \sigma_\varepsilon^2 \mathcal{I})^{-1} \mathbf{y}_0$$

$$\sigma_* = k(x_*, x_*) - k(x_*, \mathbf{x}_0)^T (k(\mathbf{x}_0, \mathbf{x}_0) + \sigma_\varepsilon^2 \mathcal{I})^{-1} k(x_*, \mathbf{x}_0)$$

Prior covariance

Info that observation gives us about the function

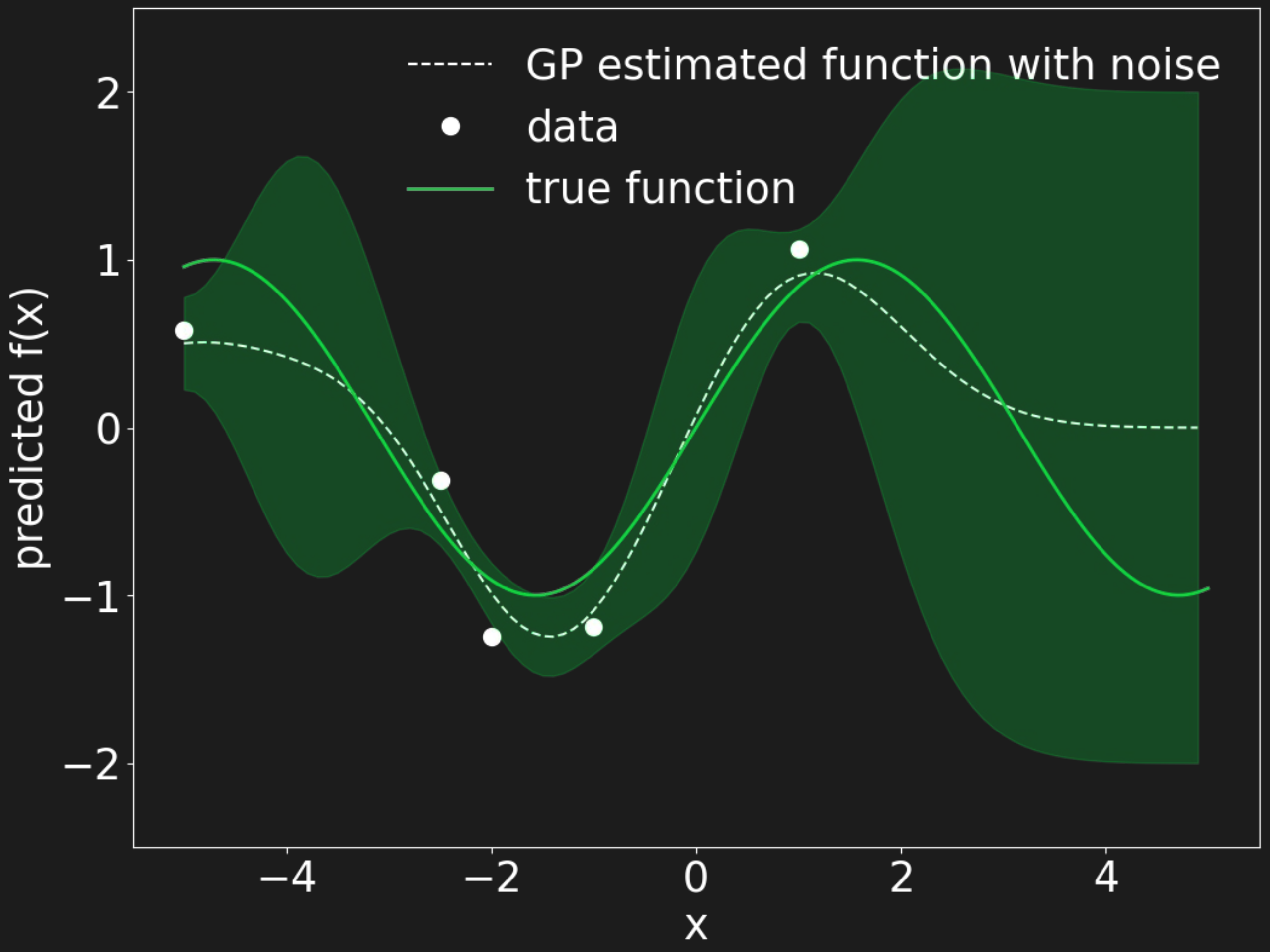
Free noise signal



Posterior obtained from the conditioning of the prior over the white data point. The white dashed curve represents $\mu*(x_*)$ and the shaded area $\pm 2\sigma_*(x_*)$.

The hyper-parameters were fixed: $\sigma_f=1$, $l=0$, $\sigma_\varepsilon=0$.

Noisy signal



Posterior obtained from the conditioning of the prior over the white data point. The white dashed curve represents $\mu_*(x_*)$ and the shaded area $\pm 2\sigma_*(x_*)$.

The hyper-parameters were fixed: $\sigma_f=1$, $l=0$ and $\sigma_\varepsilon=0.4$

Which are the best hyper-parameters?

Parameters: they **define the model** and can be learned from the data (e.g. coefficients of a linear model or the weights in a neural network).

Hyper-parameters: they are **external to the model** and cannot be estimated from the data (like the learning rate for neural networks). However, they can be **optimized** in 2 ways:

Fully Bayesian framework:

→ non-gaussian likelihood → rely on **Monte Carlo methods** (computationally expensive)

or

Maximizing the log-likelihood:

Optimization + matrix inversion

Gaussian Processes are non-parametric models.

Likelihood: given some parameters, the higer it is, the more likely it will be that we sample that observed data.

$$\log p(\mathbf{y}|\mathbf{x}_0) = -\frac{1}{2}\mathbf{y}^T(k(\mathbf{x}_0,\mathbf{x}_0)+\sigma_\varepsilon^2\bar{\mathcal{I}})^{-1}\mathbf{y} - \frac{1}{2}\log|k(\mathbf{x}_0,\mathbf{x}_0)+\sigma_\varepsilon^2\bar{\mathcal{I}}| - \frac{N}{2}\log 2\pi$$

Data fit:
It decreases monotonically with the length scale (l) → less flexible model → worse fit

Minus complexity penalty:
The simpler the model (big l scale) the bigger it becomes

N=number of training points

Likelihood: try to favour the least complex model able to explain the data (automatic Occam Razor).

Summary

- **Newtonian noise** (NN) affects the **low frequency band** of GW detectors
- We can reduce it with an **active noise cancellation**
- The **Wiener filter** can be employed to estimate the NN
- To maximize the noise estimation we need to find the **optimized seismic array**
- We need a **global optimizer** (3 examples: PSO, DE, BH)
- When the seismic **field is complicated**, calculating the cost function for the optimizer is not an easy task
- We can make use of **Gaussian Processes** and the **convolution theorem**

Let's go back to the optimization for the Newtonian noise:

Exercise

link

Don't hesitate to contact me:
francesca.badaracco@uclouvain.be

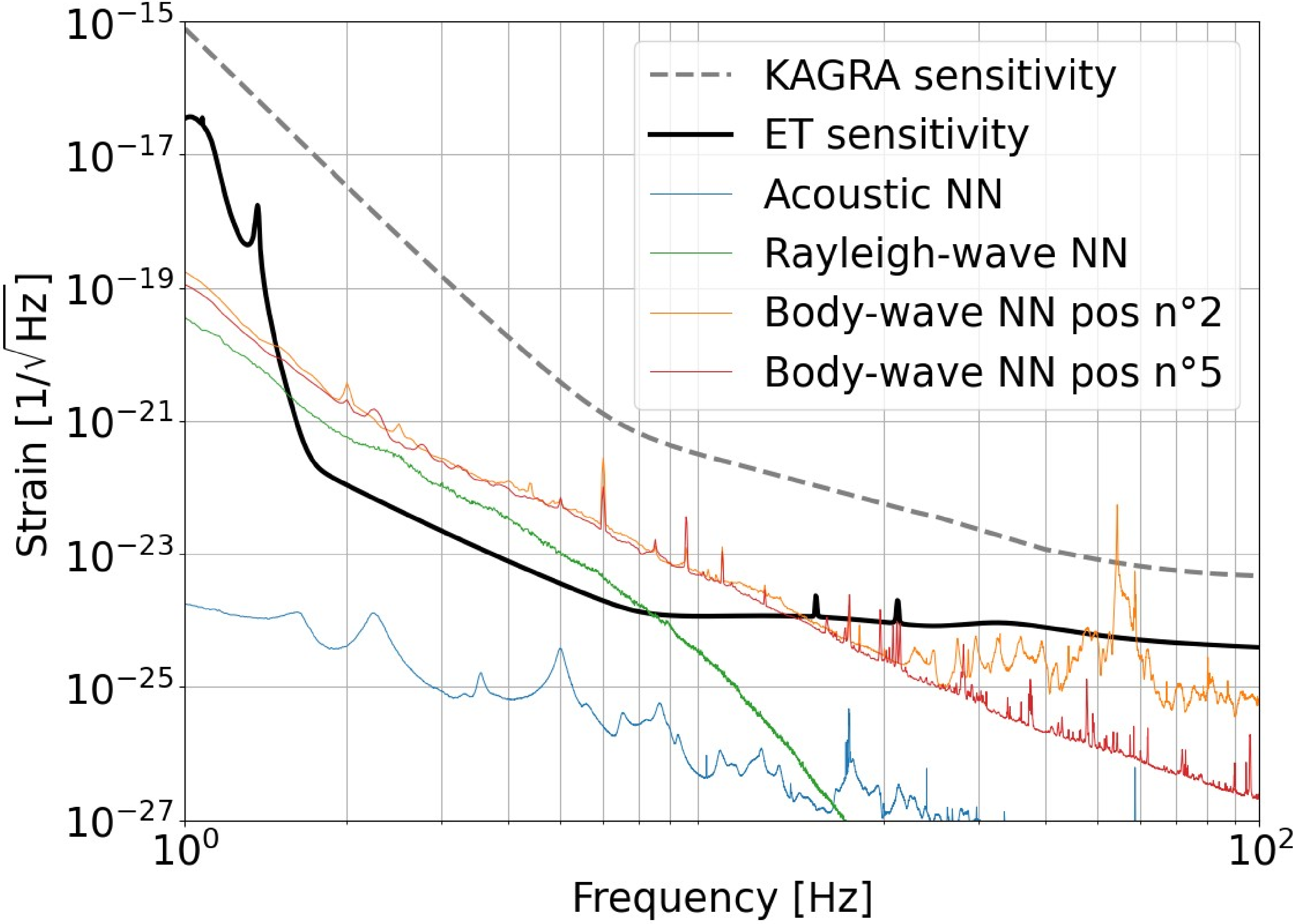
Useful references:

- **Newtonian Noise:**
  - Harms J., Terrestrial gravity fluctuations. Living Rev Relativ 22, 6 (2019). LINK
  - Badaracco, F., Newtonian noise studies in 2nd and 3^rd gravitational-wave interferometric detectors. LINK
  - Bader M., Seismic and Newtonian noise modeling for Advanced Virgo and Einstein Telescope. LINK
- **Stationarity:**
  - Shynk J., Probability, Random Variables, and Random Processes: Theory and Signal Processing Applications. John Wiley & Sons, 2012. Section 6.5

- **Wiener filter:**
  - Saeed V. Vaseghi, Advanced Digital Signal Processing and Noise Reduction. Third Edition, John Wiley & Sons, 2006. Chapter 6.
- **Particle Swarm:**
  - Kennedy J. et al., Particle Swarm Optimization
  - LINK
- **Differential Evolution:**
  - Storn R. et al., Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces, 1997.
- **Basin Hopping:**
  - Wales D., et al., Global Optimization by Basin-Hopping and the Lowest Energy Structures of Lennard-Jones Clusters Containing up to 110 Atoms, 1997.
- **Gaussian Processes:**
  - Rasmussen C, William C., Gaussian Processes for Machine Learning LINK FREE BOOK
  - LINK1 (notebook)
  - LINK2 (visual exploration of GP)

If you survived awake until now: **thank you for your attention,**

otherwise… I am sorry I made you sleep!


KAGRA sensitivity
ET sensitivity
Acoustic NN
Rayleigh-wave NN
Body-wave NN pos n°2
Body-wave NN pos n°5
Strain [1/√Hz]
Frequency [Hz]
10^{-15}
10^{-17}
10^{-19}
10^{-21}
10^{-23}
10^{-25}
10^{-27}
10^{0}
10^{2}

$$\mathbf{w}_k = \begin{pmatrix} w_{k_0} \\ ... \\ w_{k_N} \end{pmatrix} \qquad \mathbf{y}[m] = \begin{pmatrix} y_0 \\ ... \\ y_N \end{pmatrix} \qquad \hat{X}(\omega) = \mathbf{W}^T(\omega)\mathbf{Y}(\omega) = \mathbf{Y}^T(\omega)\mathbf{W}(\omega)$$

$$\begin{aligned}
E\left[e^*e\right] &= E\left[\left(X - \mathbf{Y}^T\mathbf{W}\right)^*\left(X - \mathbf{Y}^T\mathbf{W}\right)\right] \\
&= E\left[XX^* - X(\mathbf{Y}^T\mathbf{W})^* - X^*\mathbf{Y}^T\mathbf{W} + (\mathbf{Y}^T\mathbf{W})(\mathbf{Y}^T\mathbf{W})^*\right] = \\
&= E\left[XX^* - X\sum_{\mathrm{i}} Y_{\mathrm{i}}^* W_{\mathrm{i}}^* - X^*\sum_{\mathrm{i}} Y_{\mathrm{i}} W_{\mathrm{i}} + (\sum_{\mathrm{i}} Y_{\mathrm{i}} W_{\mathrm{i}})(\sum_{\mathrm{j}} Y_{\mathrm{j}}^* W_{\mathrm{j}}^*)\right] = \\
&= P_{XX} - \sum_{\mathrm{i}} W_{\mathrm{i}} P_{XY_{\mathrm{i}}}^* - \sum_{\mathrm{i}} W_{\mathrm{i}}^* P_{XY_{\mathrm{i}}} + \sum_{\mathrm{i,j}} W_{\mathrm{i}} W_{\mathrm{j}}^* P_{YY_{\mathrm{ij}}}
\end{aligned}$$

$$\frac{\partial}{\partial \mathbf{W}^*} E[e^*e] = \left(\partial_{W_1^*}, ..., \partial_{W_N^*}\right) E[e^*e] = 0 \qquad \mathbf{P}_{XY} = \bar{\mathbf{P}}_{YY}\mathbf{W} \rightarrow \mathbf{W} = \left(\bar{\mathbf{P}}_{YY}\right)^{-1}\mathbf{P}_{XY}$$

$$
\begin{aligned}
E\left[e^{*}e\right] &= P_{XX} - 2\sum_{i,j}(P_{YY}^{-1})_{\mathrm{ij}} P_{XY_{\mathrm{j}}} P_{XY_{\mathrm{i}}}^{*} + \sum_{i,j}\sum_{m,l}(P_{YY}^{-1})_{im} P_{XY_{m}} (P_{YY}^{-1})_{jl}^{*} P_{XY_{l}} (P_{YY})_{\mathrm{ij}} = \\
&= P_{XX} - \sum_{i,j}(P_{YY}^{-1})_{\mathrm{ij}} P_{XY_{\mathrm{j}}} P_{XY_{\mathrm{i}}}^{*} = \\
&= P_{XX} - \mathbf{P}_{XY}^{\dagger}\bar{\mathbf{P}}_{YY}^{-1}\mathbf{P}_{XY}
\end{aligned}
$$

$$
R(\omega) = 1 - \frac{\mathbf{P}_{XY}^{\dagger}\bar{\mathbf{P}}_{YY}^{-1}\mathbf{P}_{XY}}{P_{XX}}
$$