GWFAST: a Fisher-matrix Python code for third-generation gravitational-wave detectors

Francesco Iacovelli

Based on *Astrophys. J.* 941 (2022) 2, 208 and *Astrophys. J. Supp.* 263 (2022) 1, 2.

In collaboration with: Michele Mancarella, Stefano Foffa, Michele Maggiore

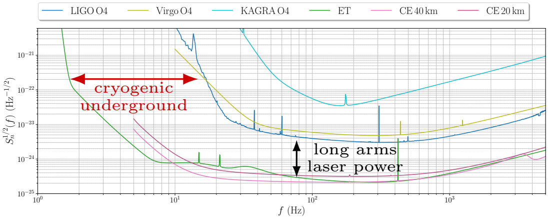
University of Geneva (UNIGE) – Department of Theoretical Physics

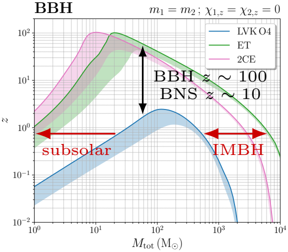
XIII ET Symposium – Cagliari 2023

Introduction: 3G GW detectors

2G detectors offer outstanding possibilities. . .
. . . but the potential of 3G detectors is unprecedented

Thanks to their technological advancements and the bigger facilities, ET and CE will have a broader frequency range and sensitivities improved more than 10 times compared to LVK





Assessing the capabilities of 3G detectors is fundamental to take informed decisions!

Introduction: challenged by the numbers

One of the key challenges when performing studies for ET and CE that emerged in recent years is the number of detectable sources

Network	BBH/yr	BNS/yr	NSBH/yr
LVK–O4	$\mathcal{O}(10^2)$	$\mathcal{O}(1-10)$	$\mathcal{O}(1-10)$
ET	$\mathcal{O}(10^4)$	$\mathcal{O}(10^3-10^5)$	$\mathcal{O}(10^3-10^4)$
ET+2CE	$\mathcal{O}(10^4-10^5)$	$\mathcal{O}(10^4-10^5)$	$\mathcal{O}(10^3-10^5)$

Currently used Bayesian parameter estimation codes, like BILBY, can take $\mathcal{O}(1\,\text{day/ev})$ to perform the analysis…

…and we do not have 10^5 days :'(

Introduction: Fisher codes

Various groups all across the world started to tackle the problem, and by now there are three public codes that can perform such a complex analysis exploiting the Fisher matrix formalism:

GWBENCH: a novel Fisher information package for gravitational-wave benchmarking

S. Borhanian[1,2]
[1] *Institute for Gravitation and the Cosmos, Department of Physics, Pennsylvania State University, University Park, PA 16802, USA*
[2] *Theoretisch-Physikalisches Institut, Friedrich-Schiller-Universität Jena, 07743, Jena, Germany**
(Dated: August 31, 2021)

GWFISH: A simulation software to evaluate parameter-estimation capabilities of gravitational-wave detector networks

Jan Harms[1,2], Ulyana Dupletsa[1,2], Biswajit Banerjee[1,2], Marica Branchesi[1,2], Boris Goncharov[1,2], Andrea Maselli[1,2], Ana Carolina Silva Oliveira[3], Samuele Ronchini[1,2], and Jacopo Tissino[1,2]
[1] *Gran Sasso Science Institute (GSSI), I-67100 L'Aquila, Italy*
[2] *INFN, Laboratori Nazionali del Gran Sasso, I-67100 Assergi, Italy and*
[3] *Department of Physics, Columbia University in the City of New York, New York, NY 10027, USA*
(Dated: May 6, 2022)

GWFAST: a Fisher information matrix Python code for third–generation gravitational–wave detectors

Francesco Iacovelli[1], Michele Mancarella[1], Stefano Foffa[1] and Michele Maggiore[1]
[1] *Département de Physique Théorique, Université de Genève, 24 quai Ernest Ansermet, 1211 Genève 4, Switzerland*

see also TiDoFM, Li et al. (2022) and Pieroni et al. (2022)

Introduction: Fisher codes

These independent codes, all featuring some peculiar implementations, have been cross–checked to assess their agreement in the context of the ET OSB Div9

`GWBENCH`, `GWFISH` and `GWFAST` have been used to produce different science cases for 3G detectors this year:

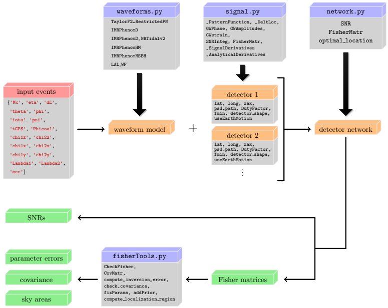
– Borhanian, Sathyaprakash (2022)
– Ronchini et al. (2022)
– FI, Mancarella, Foffa, Maggiore (2022)

Each paper focuses on some particular aspects, but all of them contribute to the blossoming future of GW science!

Introduction: GWFAST

GWFAST is particularly tuned towards high computational speed, user friendliness, and accuracy in derivative evaluation (which is the key element of the Fisher approximation), in particular:

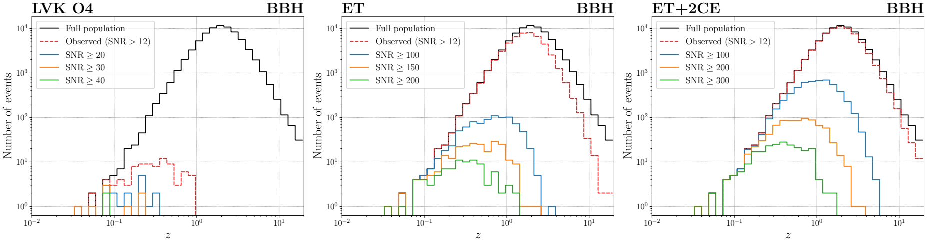
- $\Longrightarrow$ derivatives are computed using automatic differentiation with JAX
- $\Longrightarrow$ the code is written in pure Python (also the waveforms! See WF4Py)
- $\Longrightarrow$ vectorization is exploited to handle multiple events at a time, even on a single CPU



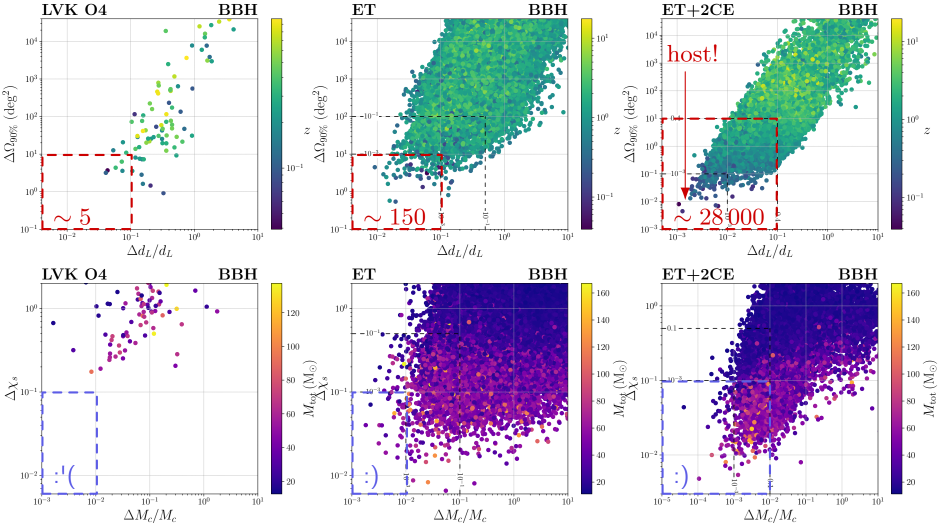
GWFAST needs $\lesssim 1$ day to run the PE on 10^5 events!

Forecasts with GWFAST: assessing the potential of 3G detectors

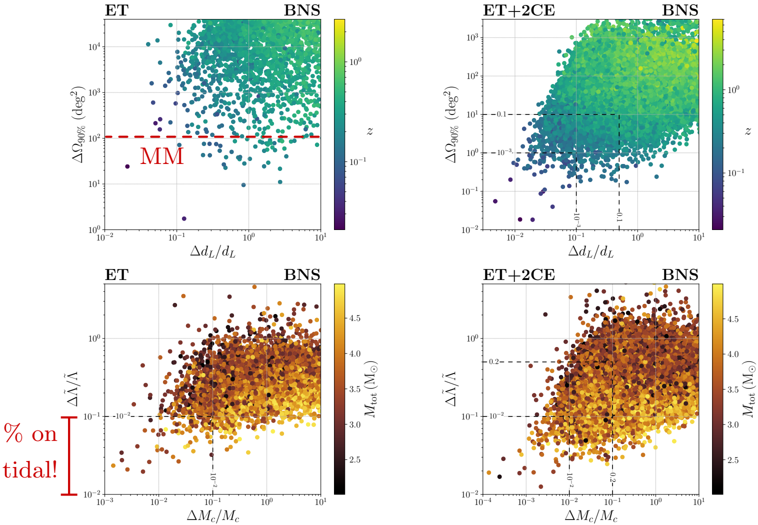
Simulating synthetic merger populations, based on the latest LVK results, through **GWFAST** it is possible to assess the capabilities of GW detectors, comparing among different networks and configurations and for different sources



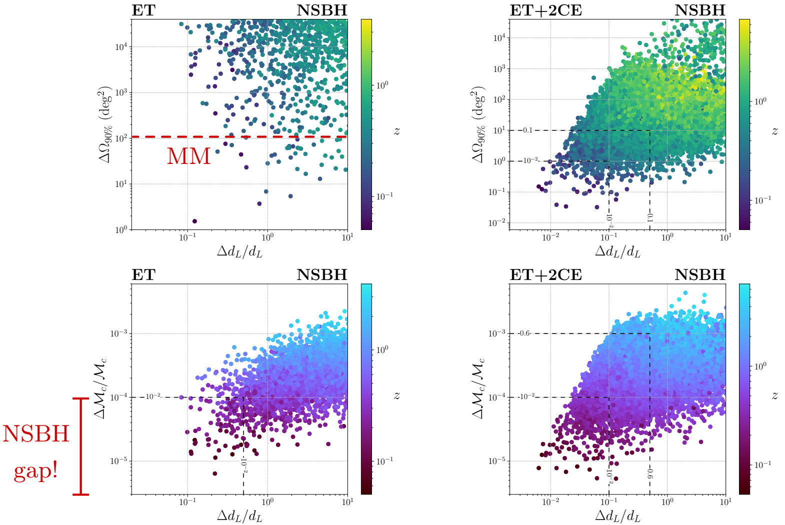
Forecasts with GWFAST: BBHs at 3G detectors



Forecasts with GWFAST: BNSs at 3G detectors



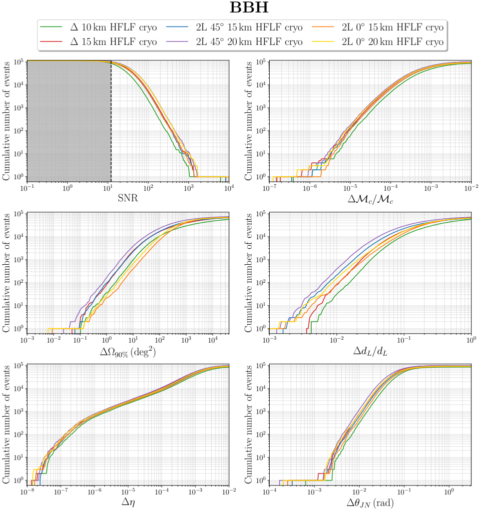
Forecasts with GWFAST: NSBHs at 3G detectors



Forecasts with GWFAST: comparing configurations

Tools like **GWFAST** allow to efficiently compare different detector configurations and designs, e.g. varying the geometry or PSD, as done for the "CoBA–science" study!

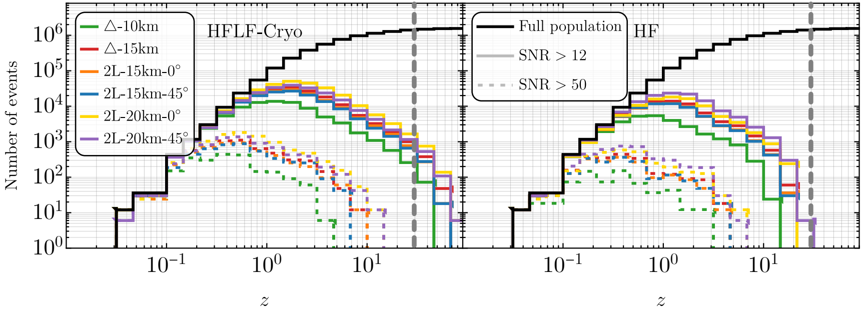
This can give us the possibility to push even further the extraordinary potential of 3G instruments, and understand which features can be more relevant for specific science cases.



Forecasts with GWFAST: comparing configurations

As an example, for potential primordial black hole binaries at high redshifts the low-frequency instrument of ET turns out to be crucial!

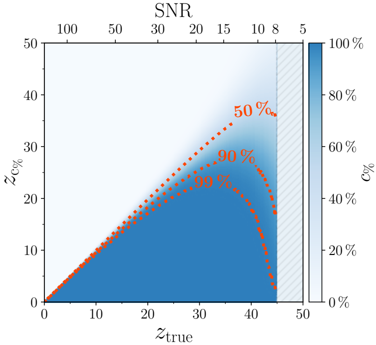
Franciolini, FI, et al. (2023)

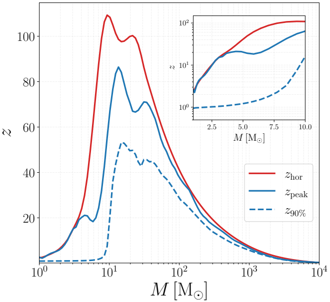


Forecasts with GWFAST: new metrics

Having tools like **GWFAST** can also allow us to easily test different metrics that can come up to our mind!

As an example, we might be interested in the accuracy we can obtain in the reconstruction of the luminosity distance of high redshift binaries, to understand if we will be able to tell they are out there

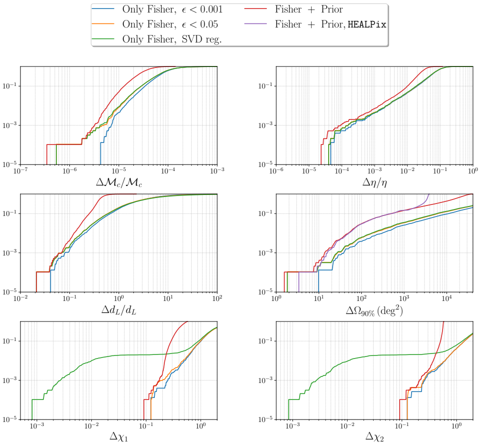




Mancarella, FI, Gerosa (2023)

Conclusions and future work

- Fisher matrix can provide a sensible approximation to forecast the capabilities of third generation detectors
- The existence of tools like GWFAST allows us to compare detector configurations and designs in a quick way
- We can try to push the Fisher approach further, e.g. imposing physical priors or finding more suitable parameters



Thanks for your attention. . . questions?

I am also available at Francesco.Iacovelli@unige.ch

GW parameter estimation: the GW likelihood

A GW signal as observed by a detector can be expressed as

$$s(t) = h_0(t) + n(t)$$

Defining the inner product for any two time–domain signals as

$$(a\,|\,b) = 4\,\mathrm{Re}\left\{\int_0^\infty \mathrm{d}f\,\frac{\tilde{a}^*(f)\,\tilde{b}(f)}{S_n(f)}\right\} \implies \mathrm{SNR} = (h_0\,|\,h_0)^{1/2}$$

we have for the GW likelihood, choosing a waveform model h,

$$\mathcal{L}(s\mid\boldsymbol{\theta}) \propto \exp\{-\left(s-h(\boldsymbol{\theta})\,|\,s-h(\boldsymbol{\theta})\right)/2\}$$

GW parameter estimation: MCMC timing for PE

Performing a full Bayesian PE for a GW signal via an MCMC sampling of the likelihood is computationally expensive

Signal	Sampler	n_ℓ	$n_{\rm samples}^{\rm eff}$
BBH	DYNESTY	2.2×10^8	15000
	BILBY-MCMC	3×10^8	5000
BNS	BILBY-MCMC	2.5×10^9	5000

Ashton, Talbot (2021)

With BILBY it can take $\gtrsim \mathcal{O}(1\ \mathrm{day/ev})$ to perform the estimation

Full PE is not feasible for 10^5 events

GW parameter estimation: Fisher matrix

In the linearized signal approximation / high–SNR limit, the GW likelihood can be approximated as a multivariate Gaussian with covariance

$$\mathrm{Cov}_{ij} = \Gamma^{-1}_{ij}\,, \quad \Gamma_{ij} \equiv -\left.\langle \partial_i \partial_j \log \mathcal{L}(s\,|\,\boldsymbol{\theta})\rangle_n\right|_{\boldsymbol{\theta}_0} = \left.(\partial_i h(\boldsymbol{\theta})\,|\,\partial_j h(\boldsymbol{\theta}))\right|_{\boldsymbol{\theta}_0}$$

Γ_{ij} being the Fisher matrix

The key ingredients are then computing derivatives and. . . speed!

GWFAST implementations: derivatives

Usually derivatives are computed using finite difference techniques, but this has some limitations, consider e.g.

$$f(x) = \sin\left(\ln\left(\sqrt{x}\right)\right) \implies f'(x) = \cos\left(\ln\left(\sqrt{x}\right)\right)/2x$$

```
eps = 1e-5
print((f(10.+eps) - f(10.))/eps - fp(10.))
```

```
0.003476493
```

Every function with a closed form expression, however complex, is built from simple operations $(+, -, \times, \div)$, and well–known functions $(\exp, \cos, \ln, \dots)$ whose derivative is trivial.

What a pity a machine cannot understand it. . . wait, it can!

GWFAST implementations: derivatives

Automatic differentiation is a technique to make a machine compute derivatives of any order in a pseudo–analytic way, iteratively applying the *chain rule* on a given function.

`GWFAST` uses the module `JAX` for automatic differentiation, that applied to our example function gives

```
JAXfp = jax.grad(f)
print(JAXfp(10.) - fp(10))
```

```
0.0
```

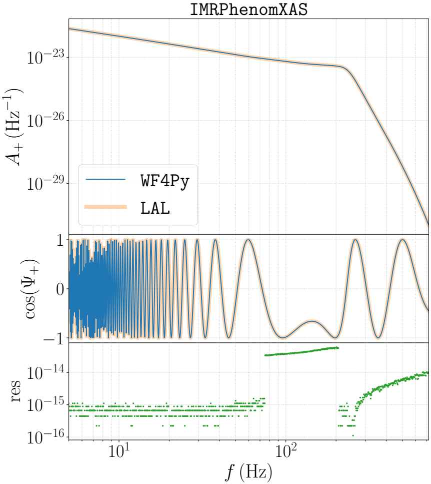
The only requirement is to write the function in a way the machine can understand, in our case pure `Python`... but `LAL` is written in `C`

GWFAST implementations: waveforms

To make JAX work we translated the waveform models in Python and carefully checked the adherence with their originals.

We released them also as a separate module, WF4Py, which features:

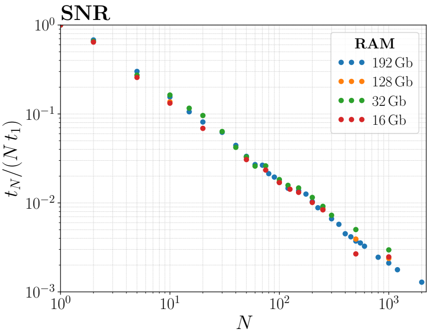
TaylorF2,
IMRPhenomD,
IMRPhenomD_NRTidalv2,
IMRPhenomHM,
IMRPhenomNSBH,
IMRPhenomXAS



GWFAST implementations: vectorization

Having a pure `Python` code and using `JAX`, it is possible to exploit what is called *vectorization*, i.e. the possibility to perform calculations for multiple events at a time even on a single CPU, not resorting to `for` loops.

This makes `GWFAST` ideal to handle large catalogs!



GWFAST testing

To asses the reliability of **GWFAST** we performed the PE analyses on the samples of real GW events with high SNR and good sky location, finding consistent results