



# CHORUS

This is the accepted manuscript made available via CHORUS. The article has been published as:

## Very fast stochastic gravitational wave background map making using folded data

Anirban Ain, Jishnu Suresh, and Sanjit Mitra

Phys. Rev. D **98**, 024001 — Published 2 July 2018

DOI: [10.1103/PhysRevD.98.024001](https://doi.org/10.1103/PhysRevD.98.024001)

# Very fast stochastic gravitational wave background map-making using folded data

Anirban Ain,<sup>\*</sup> Jishnu Suresh,<sup>†</sup> and Sanjit Mitra<sup>‡</sup>

*Inter-University Centre for Astronomy and Astrophysics (IUCAA), Pune 411007, India*

A stochastic gravitational-wave background (SGWB) is expected from the superposition of a wide variety of independent and unresolved astrophysical and cosmological sources from different stages in the evolution of the Universe. Radiometric techniques are used to make sky maps of anisotropies in the SGWB by cross-correlating data from pairs of detectors. The conventional searches can be made hundreds of times faster through the folding mechanism introduced recently. Here we present a newly developed algorithm to perform the SGWB searches in a highly efficient way. Taking advantage of the compactness of the folded data we replaced the loops in the pipeline with matrix multiplications. We also incorporated well-known HEALPix pixelization tools for further standardization and optimization. Our Python-based implementation of the algorithm is available as an open source package `PyStoch`. Folding and `PyStoch` together has made the radiometer analysis *a few thousand times* faster; it is now possible to make all-sky maps of a stochastic background in just a few minutes on an ordinary laptop. Moreover, `PyStoch` generates a skymap at every frequency bin as an intermediate data product. These techniques have made SGWB searches very convenient and will make computationally challenging analyses like blind all-sky narrowband search feasible.

PACS numbers: 04.80.Nn, 95.55.Ym, 98.70.Vc

## I. INTRODUCTION

A new era in astronomy began with the detection of gravitational waves (GW) [1]. As was anticipated, the first detected sources are all compact binary coalescence [2–6]. GW astronomy, however, promises much more excitement. A vigorous global effort is underway to observe GW signals in widely separated frequency bands. This includes ground-based interferometric detectors aLIGO, aVIRGO, GEO, TAMA, KAGRA [7–11], pulsar timing arrays [12], and the planned space-based detectors such as LISA, DECIGO, BBO [13–15]. Different kinds of sources [16–18] are expected to be seen in the current and future generation network of detectors, the stochastic background being one of the most interesting ones. A stochastic background of gravitational waves (SGWB) [19, 20] can be generated by the superposition of a wide variety of independent and unresolved astrophysical and cosmological sources from different stages in the evolution of the universe. The unresolved compact binary coalescence consisting of black holes and neutron stars, spinning neutron stars, supernovae, cosmic strings, inflationary models, phase transitions, and the pre-Big-Bang scenario are some of these sources which contribute towards this background [21–27]. The background is likely to be anisotropic if it is dominated by the nearby universe [25, 28]. The detection of an anisotropic SGWB will offer novel opportunities to study the origin, distribution, and properties of various astrophysical objects, which are still not accessible to the current observational windows of astronomy [21, 29, 30].

Various techniques have been developed to search for isotropic and anisotropic SGWB in data from detectors in different bands [31]. For GW detectors, since noise is geographically far away detectors are likely to be uncorrelated, data from pairs of detectors are cross-correlated to look for a stochastic signal [28, 32–36]. To probe anisotropies of the SGWB using ground-based detectors, the standard technique presently is the GW radiometer algorithm [37–39], which is analogous to aperture synthesis often used in radio astronomy [40, 41]. The algorithm has been thoroughly studied and implemented in a pixel [38, 39] and spherical harmonic basis [42], where appropriate time-varying phase delays are applied to probe various spatial scales, and is routinely applied to LIGO data [43–47]. GW can also cause a considerable amount of perturbations in the time of arrival of pulses from millisecond pulsars. Efforts are on to measure these perturbations for some of the well modelled pulsars [48–50] to constrain stochastic background in the nano-Hertz band, as well as, to probe the anisotropy of the background [51].

In a recent work, the efficiency of the GW radiometer algorithm was dramatically improved through the mechanism of data folding [52]. A temporal symmetry in the GW radiometry algebra was utilized to fold the entire data into one sidereal day, thereby reducing the computational cost by a factor equal to the total number of days of observation. The enormous efficiency brought by folded data allows one to perform new kinds of analyses on long duration signals observed by ground based detectors using much lower computational resources, e. g., a blind all-sky narrowband search [53]. A parallel pipeline developed to implement folding on LIGO data, and the advantages gained from that are well documented [52].

However, folding does not exhaust the possibilities for computational improvement of the radiometer analysis. Here we present steps that can boost the efficiency of the algorithm by another factor of few tens. Moreover,

---

<sup>\*</sup>Electronic address: [ainz@iucaa.in](mailto:ainz@iucaa.in)

<sup>†</sup>Electronic address: [jishnus@iucaa.in](mailto:jishnus@iucaa.in)

<sup>‡</sup>Electronic address: [sanjit@iucaa.in](mailto:sanjit@iucaa.in)

the existing implementation of the stochastic pipeline is not equipped to take full advantage of folded data. Since the folded data volume is generally between 1-10 GB, the whole dataset can be loaded into a computer's RAM, thereby dramatically reducing disk access and providing huge I/O advantage.

The scope for improvement is not limited to computational cost and associated conveniences. The current searches for an anisotropic SGWB in LIGO-Virgo data either use a spherical harmonic basis [46], which is appropriate for smooth and diffuse sources, but has limited sensitivity for localized sources like a galaxy cluster, or a pixel space radiometer search on a Cartesian grid. However, analysis of an equal area pixelization scheme is highly desirable for an all-sky map-making application like ours, primarily for better handling of noise in every pixel. Here we present the first implementation of radiometer search that is fully integrated with the Hierarchical Equal Area isoLatitude Pixelization of the sphere (HEALPix) scheme [54], which is perhaps the most widely-used pixelization scheme in astronomy at present for describing all-sky maps and has been rigorously tested by the cosmology community. Apart from the advantages of equal area pixels, HEALPix offers highly efficient tools for Fourier transforms on the sky, making it easy to transform a map from pixel to spherical harmonic bases and vice-versa, thus making the analysis suitable for both localised and diffused sources<sup>1</sup>.

Finally, the existing LIGO-Virgo stochastic pipeline is suitable for broadband searches or targeted narrowband searches. The directional upper limits on GW intensity using data from the first Advanced LIGO observing run (O1) has been calculated [55]. With the fast tools in hand, we wanted to design the implementation in such a way that the intermediate results, the observed maps at every frequency bin, should be derivable in a straightforward way, which can then be combined to get the broadband result. This would also alleviate the need to perform separate searches for different spectral shapes of the modeled power spectral densities of the sources.

This paper is organized as follows. We briefly review the GW radiometer formalism as well as the folding method in Sec. II. Algorithms used in `PyStoch` and the narrowband map-making process are explained in Sec. III. Implementation of `PyStoch` and the results are presented in Sec. IV. We conclude the paper with a summary and discussion of algorithm performance in Sec. V along with results from simulated data, whose noise characteristics are statistically identical to that of O1 data.

<sup>1</sup> However it is important to note that in practice, the final results (e.g., clean map, upper limit map, etc.) do depend on the chosen basis because of the different kinds of numerical errors associated with the inversion of the Fisher information matrices in different bases.

## II. MAPPING THE STOCHASTIC GRAVITATIONAL WAVE BACKGROUND

### A. Gravitational wave radiometer

Detectors which are geographically separated by vast distances are expected to have nearly statistically independent noise. Moreover, since a Gaussian stochastic background is characterized by its second moment, weighted cross-correlation of data from two independent detectors turns out to be the optimal statistic for detecting and mapping a SGWB [35, 39]. The GW radiometer algorithm is based on this fundamental principle, which is commonly used to search for an isotropic or anisotropic SGWB in data from the ground-based interferometric detectors.

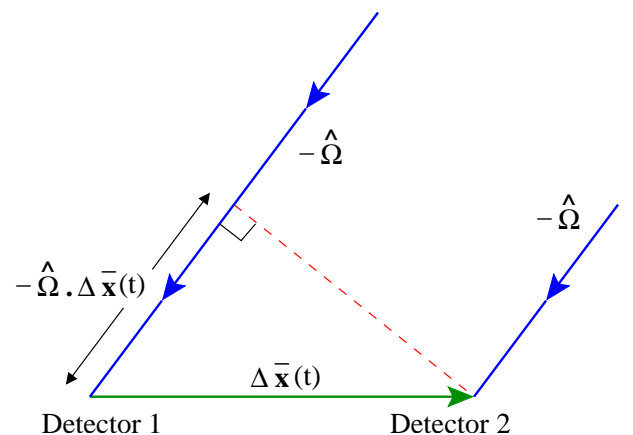


FIG. 1: Geometry of baseline: schematic diagram of the radiometer. To measure the signal coming from a direction  $\hat{\Omega}$ , data from the detectors are correlated with a phase delay of  $\hat{\Omega} \cdot \Delta \mathbf{x}_I(t)/c$ .

The GW radiometer algorithm considers the delay in the GW signal arrival at detectors which are at different locations. For a given direction in the sky, this delay changes as the baseline orientation changes due to the Earth's rotation. If the time delayed data from two detectors are cross-correlated, the potential GW signals arriving from the given direction interfere constructively while the noises do not. When integrated over a long observational time, the signal cross-correlation grows faster than the noise variance, making the detection statistic more and more significant. The changes mentioned above in baseline orientation due to the Earth's rotation allows one to construct an SGWB map by performing the synthesis for each direction in the sky, with an appropriate choice of the size of the pixel, even with only two detectors.

The working principle can be explained with the help of Fig 1. Consider two gravitational wave detectors denoted by  $\mathcal{I}_1$  and  $\mathcal{I}_2$ . The output data from the detectors

can be written as,

$$s_{\mathcal{I}_1}(t) = h_{\mathcal{I}_1}(t) + n_{\mathcal{I}_1}(t), \quad (1)$$

$$s_{\mathcal{I}_2}(t) = h_{\mathcal{I}_2}(t) + n_{\mathcal{I}_2}(t), \quad (2)$$

where  $h_{\mathcal{I}_1}(t)$  and  $h_{\mathcal{I}_2}(t)$  denote the gravitational wave strain in the two detectors due to the SGWB, and  $n_{\mathcal{I}_1}(t)$  and  $n_{\mathcal{I}_2}(t)$  denote the noises intrinsic to the first and second detectors respectively. Since the response of a detector (i.e., its antenna pattern with respect to the sky) changes with time, it is convenient to split the data taken from the detectors (time series data) into chunks of duration  $\tau$ , such that  $\tau$  is much greater than the light-travel time between the detectors, but small enough so that the detector response functions do not change considerably over that period. The typically acceptable chunk sizes are a few tens to a few hundreds of seconds. The Fourier transform of such a data segment, often called a *short-term Fourier transforms* (SFTs), is defined as,

$$\tilde{s}_I(t; f) := \int_{t-\tau/2}^{t+\tau/2} dt' s_I(t') e^{-i2\pi f t'}, \quad (3)$$

where a tilde above a variable denotes its Fourier transform of a time series. As mentioned before, the noise in different geographically distant detectors can be assumed to be mutually uncorrelated and also uncorrelated with the signal. Thus we have,

$$\langle \tilde{h}_{\mathcal{I}_{1,2}}^*(t; f) \tilde{n}_{\mathcal{I}_{1,2}}(t; f) \rangle = 0, \quad (4)$$

$$\langle \tilde{n}_{\mathcal{I}_1}^*(t; f) \tilde{n}_{\mathcal{I}_2}(t; f) \rangle = 0. \quad (5)$$

The cross-power spectral density (CSD) of data,  $\mathbf{C}^I$ , from a baseline  $I$  constituted by two detectors  $\mathcal{I}_1$  and  $\mathcal{I}_2$  and its noise component  $\mathbf{n}^I$  are given by,

$$\mathbf{C}^I \equiv C_{ft}^I := \tilde{s}_{\mathcal{I}_1}^*(t; f) \tilde{s}_{\mathcal{I}_2}(t; f), \quad (6)$$

$$\mathbf{n}^I \equiv n_{ft}^I := \tilde{n}_{\mathcal{I}_1}^*(t; f) \tilde{n}_{\mathcal{I}_2}(t; f). \quad (7)$$

In the small signal limit,  $\langle |\tilde{h}(t; f)|^2 \rangle \ll \langle |\tilde{n}(t; f)|^2 \rangle$ , the instantaneous cross-power noise variance becomes,

$$\sigma_{Ift}^2 := \langle n_{ft}^{I*} n_{ft}^I \rangle = \frac{\tau^2}{4} P_{\mathcal{I}_1}(t; f) P_{\mathcal{I}_2}(t; f), \quad (8)$$

where  $P_{\mathcal{I}_{1,2}}(t; f)$  are the one-sided power spectral density (PSD) of noise  $n_{\mathcal{I}_{1,2}}$  for a segment at time  $t$  and  $\tau$  is the duration of a segment.

The SGWB is characterized by the CSD of the signal. A search for anisotropic SGWB for a specific spectral distribution  $H(f)$  (assuming that the frequency spectral shape is the same in every direction of the sky) boils down to estimation of the SGWB skymap  $\mathcal{P}(\hat{\Omega})$  that is proportional to the flux coming from different directions on the sky [39]. One can however perform the search in different set of basis  $e_p(\hat{\Omega})$  on the two-sphere, in which the anisotropy map can be expanded as,

$$\mathcal{P}(\hat{\Omega}) := \sum_p \mathcal{P}_p e_p(\hat{\Omega}). \quad (9)$$

Then the expectation value of the CSD for a baseline can be written as [42]

$$\langle C_{ft}^I \rangle := \tau H(f) \sum_p \mathcal{P}_p \gamma_{ft,p}^I, \quad (10)$$

where  $\gamma_{ft,p}^I$  is a geometric factor usually known as the overlap reduction function (ORF) defined as [33, 56],

$$\gamma_{ft,p}^I := \sum_A \int_{S^2} d\hat{\Omega} F_{\mathcal{I}_1}^A(\hat{\Omega}, t) F_{\mathcal{I}_2}^A(\hat{\Omega}, t) e^{2\pi i f \frac{\hat{\Omega} \cdot \Delta \mathbf{x}_I(t)}{c}} e_p(\hat{\Omega}), \quad (11)$$

where  $\Delta \mathbf{x}_I(t)$  is the separation vector between the two detectors and  $F_{\mathcal{I}_{1,2}}^A(\hat{\Omega}, t)$  denotes the antenna pattern functions of the detectors. Thus the observed CSD is a convolution of the SGWB skymap with the kernel  $\mathbf{K}$ , plus an additive noise term  $\mathbf{n}$ ,

$$\mathbf{C}^I = \mathbf{K}^I \cdot \mathcal{P} + \mathbf{n}^I. \quad (12)$$

Here  $\mathbf{K}^I$ , the kernel or the beam function is defined as,

$$\mathbf{K}^I \equiv K_{ft,p}^I := \tau H(f) \gamma_{ft,p}^I. \quad (13)$$

The convolution equation has a standard maximum-likelihood (ML) solution  $\hat{\mathcal{P}}_p$  which produces the estimates for the SGWB skymaps [39],

$$\hat{\mathcal{P}}_p \equiv \hat{\mathcal{P}} = \mathbf{\Gamma}^{-1} \cdot \mathbf{X}, \quad (14)$$

where  $\mathbf{X}$ , the dirty map, is given as,

$$\mathbf{X} = \frac{4}{\tau} \sum_{Ift} \frac{H(f) \gamma_{ft,p}^{I*}}{P_{\mathcal{I}_1}(t; f) P_{\mathcal{I}_2}(t; f)} \tilde{s}_{\mathcal{I}_1}^*(t; f) \tilde{s}_{\mathcal{I}_2}(t; f), \quad (15)$$

and  $\mathbf{\Gamma}$ , the Fisher information matrix, as,

$$\mathbf{\Gamma} = 4 \sum_{Ift} \frac{H^2(f)}{P_{\mathcal{I}_1}(t; f) P_{\mathcal{I}_2}(t; f)} \gamma_{ft,p}^{I*} \gamma_{ft,p'}^I. \quad (16)$$

ML estimation of the convolution equation [Eq. (12)] takes the simple form given in Eq. (14) only when the inverse of the beam matrix exists. So directly inverting the beam matrix is non-trivial. Hence we prefer to solve the linear algebraic equation

$$\mathbf{\Gamma} \cdot \hat{\mathcal{P}} = \mathbf{X}, \quad (17)$$

to find  $\hat{\mathcal{P}}$ , the ML estimate for the GWB sky map<sup>2</sup>.

<sup>2</sup> Fisher information matrix  $\mathbf{\Gamma}$  in most cases is not easily invertible. And, due to the noise term in Eq. (12), the solution of Eq. (17) is noisy. A conjugate gradient or least square method is used to find the best 'clean' map  $\hat{\mathcal{P}}$ .

## B. Folding

As we mentioned previously, gravitational wave radiometry relies on the basic principles of Earth rotation image synthesis. Characteristic properties of the SGWB and the algebra of analysis technique reveal that there exists a temporal symmetry. In the expressions for the dirty map  $\mathbf{X}$  in Eq. (15) and the Fisher information matrix  $\mathbf{\Gamma}$  in Eq. (16), the only two quantities needed for ML estimation of  $\mathcal{P}$ , the geometric part has a period of one sidereal day (i.e. 23 hr 56 min 4 sec). One can use this property to fold the entire detector data of several hundreds of days to only one sidereal day [52]. This can be done by splitting the time segment  $t$  into multiples of sidereal day plus remainder within that sidereal day:  $t = i_{\text{day}} \times T_s + t_s$  and, correspondingly we can rearrange the sum over segments into a sum over sidereal time and a sum over days,  $\Sigma_t \rightarrow \Sigma_{i_{\text{day}}} \Sigma_{t_s}$ , where  $i_{\text{day}}$  is an integer representing the sidereal day number in which a given  $t$  lies and  $T_s$  is the duration of one sidereal day. Then one can rewrite (15) and (16) as,

$$X_p = \sum_{Ift_s} K_{ft_s,p}^{I*} \sum_{i_{\text{day}}} \sigma_{If(i_{\text{day}}T_s+t_s)}^{-2} C_{f(i_{\text{day}}T_s+t_s)}^I, \quad (18)$$

$$\Gamma_{pp'} = \sum_{Ift_s} K_{ft_s,p}^{I*} K_{ft_s,p'}^I \sum_{i_{\text{day}}} \sigma_{If(i_{\text{day}}T_s+t_s)}^{-2}. \quad (19)$$

The summation over  $i_{\text{day}}$  part of the above equations correspond to the folded data. It is evident that the folded objects are independent of the search basis indexed with  $p$ . That is, once the data are folded the same data can be used for fast searches in any basis.

The algebra gets more involved in practice due to the application of overlapping window functions to reduce spectral leakage. Noise in the neighboring segments no longer remains statistically independent.

A window function is a mathematical function of time,  $\mathcal{W}_{\mathcal{I}}(t)$ , that is zero-valued outside of some chosen interval. In GW radiometry, usually a smooth window function is applied to the time series data. Such windows would lead to loss of detector data. In order to prevent that the windows are made to overlap. The usual practice is to use a Hanning window [57] with 50% overlapping segments. Overlapping windows makes the noise in the neighboring segments partly correlated and the algebra gets more involved. After incorporating the correction in the algebra to account for the effect of overlapping windows, one can express the dirty map and the Fisher information matrix respectively as [52],

$$X_p = \sum_{Ift_s} K_{ft_s,p}^{I*} x_{ft_s}^I, \quad (20)$$

$$\Gamma_{pp'} = \sum_{Ift_s} K_{ft_s,p}^{I*} [K_{ft_s,p'}^I v_{ft_s}^I - K_{f(t_s-1),p'}^I u_{ft_s}^I - K_{f(t_s+1),p'}^I w_{ft_s}^I]. \quad (21)$$

Here the folded data are given by three real frequency series,

$$\begin{aligned} v_{ft_s}^I &= \sum_{i_{\text{day}}} \sigma_{If(i_{\text{day}}T_s+t_s)}^{-2}, \\ u_{ft_s}^I &= \sum_{i_{\text{day}}} \frac{1}{2} \varepsilon_{i_{\text{day}}}^I \sigma_{If(i_{\text{day}}T_s+t_s-1)}^{-2} \times \\ &\quad \left[ \sigma_{If(i_{\text{day}}T_s+t_s)}^{-2} + \sigma_{If(i_{\text{day}}T_s+t_s-1)}^{-2} \right], \\ w_{ft_s}^I &= \sum_{i_{\text{day}}} \frac{1}{2} \varepsilon_{i_{\text{day}}}^I \sigma_{If(i_{\text{day}}T_s+t_s+1)}^{-2} \times \\ &\quad \left[ \sigma_{If(i_{\text{day}}T_s+t_s)}^{-2} + \sigma_{If(i_{\text{day}}T_s+t_s+1)}^{-2} \right], \end{aligned} \quad (22)$$

---

and *one complex* folded time-frequency map

$$\begin{aligned} x_{ft_s}^I &= \sum_{i_{\text{day}}} \left[ \sigma_{If(i_{\text{day}}T_s+t_s)}^{-2} C_{f(i_{\text{day}}T_s+t_s)}^I - \frac{1}{2} \varepsilon_{i_{\text{day}}}^I \left\{ \sigma_{If(i_{\text{day}}T_s+t_s)}^{-2} + \sigma_{If(i_{\text{day}}T_s+t_s-1)}^{-2} \right\} C_{f(i_{\text{day}}T_s+t_s-1)}^I \right. \\ &\quad \left. - \frac{1}{2} \varepsilon_{i_{\text{day}}}^I \left\{ \sigma_{If(i_{\text{day}}T_s+t_s)}^{-2} + \sigma_{If(i_{\text{day}}T_s+t_s+1)}^{-2} \right\} C_{f(i_{\text{day}}T_s+t_s+1)}^I \right], \end{aligned} \quad (23)$$

expressed in terms of the redefined CSD,

$$\mathbf{C}^I \equiv C_{ft}^I := \frac{1}{\overline{\mathcal{W}_{\mathcal{I}_1}(t)\mathcal{W}_{\mathcal{I}_2}(t)}} \tilde{s}_{\mathcal{I}_1}^*(t; f) \tilde{s}_{\mathcal{I}_2}(t; f), \quad (24)$$

and its variance,

$$\sigma_{Ift}^2 = \frac{\overline{\mathcal{W}_{\mathcal{I}_1}^2(t)\mathcal{W}_{\mathcal{I}_2}^2(t)}}{\left[\overline{\mathcal{W}_{\mathcal{I}_1}(t)\mathcal{W}_{\mathcal{I}_2}(t)}\right]^2} \frac{\tau^2}{4} P_{\mathcal{I}_1}(t; f) P_{\mathcal{I}_2}(t; f). \quad (25)$$

---

The quantity  $\mathcal{W}_I$  is usually a small fraction that depends on the window functions. For 50% overlapping Hanning windows, which is often the standard choice in the current analyses,  $\mathcal{W}_I = 3/70$ . Here  $\tilde{s}_{\mathcal{I}_{1,2}}(t; f)$  are windowed SFTs and a line over a quantity, e.g.,  $\overline{\mathcal{W}_{\mathcal{I}_1}(t)\mathcal{W}_{\mathcal{I}_2}(t)}$ , denotes average over time [58].

$$\varepsilon_{t\pm 1}^I = \begin{cases} \mathcal{W}_I & \text{if segment } t \pm 1 \text{ exists for baseline } I \\ 0 & \text{otherwise} \end{cases}.$$

Using the deconvolution technique discussed previ-

ously, we will then produce the clean map out of the dirty map with the help of Fisher matrix. The science is carried out by producing the clean sky maps. In [52] we have discussed in detail about the advantages of the folding algorithm. The advantages include *efficiency*: computational resources reduces by a factor of few 100; *portability*: folded data size is  $\sim 1.3\text{GB}$  for standard stochastic search; *management*: irrespective of the observation time computation time is fixed; *modularity*: one can do intensive folding part in a low-level language like C and complex algebra of filtering for different searches in MATLAB or Python; and *convenience*: possible to analyse in a portable computer. In this scenario, we are aiming towards developing a new pipeline, which is capable of using the folded data to produce the sky map in pixel basis with a significant improvements.

### III. EFFICIENT MAPMAKING ALGORITHM: PyStoch

A variety of data analysis techniques have been proposed and implemented in the past for the SGWB searches. In the recent times, the searches for anisotropic backgrounds are being performed either in spherical harmonic basis, by measuring the first few spherical harmonic multipoles of the sky, or in the pixel basis with an equispaced grid in latitude and longitude (i.e., the pixel are not of equal area). The PyStoch pipeline is an attempt to develop a new implementation incorporating the Hierarchical Equal Area isoLatitude Pixelization (HEALPix) scheme [54], which makes it possible to trivially obtain the spherical harmonic moments as well. PyStoch also generates maps at every frequency bin as intermediate results. PyStoch uses some existing PyCBC modules [59, 60] and is designed to take full advantage of folded data, which, in addition to the speed-up resulted from folding, leading to nearly a factor of hundred boosts in the computational efficiency.

#### A. Narrowband maps

From Eq. (6), Eq. (8) and Eq. (15), it is evident that one can split the expression for dirty map into a frequency sum and a time dependent sum,

$$X_p = \tau \sum_f H(f) \sum_{I_t} \gamma_{ft,p}^{I_t*} \sigma_{I_t}^{-2} C_{ft}^I, \quad (26)$$

such that, the broadband map  $X_p$  becomes a source spectrum weighted sum of narrowband maps  $X_{p,f}$ ,

$$X_p = \sum_f H(f) X_{p,f}, \quad (27)$$

where the expression of the narrowband maps at each frequency is given by,

$$X_{p,f} = \tau \sum_{I_t} \gamma_{ft,p}^{I_t*} \sigma_{I_t}^{-2} C_{ft}^I. \quad (28)$$

In the existing pipeline, the summation over time follows the summation over frequency because the length of data is arbitrary. This order is non-trivial to change once the pipeline is fully developed. PyStoch is designed to do the time summation first and hence the narrowband maps are automatically produced as an intermediate product. Computation of the Fisher information matrix for each frequency bin can also be split in this manner, indicating that it will be possible to deconvolve the narrowband maps or to produce upper limit maps at each frequency saving the extra computation power that was necessary for making these maps.

#### B. HEALPix

HEALPix is arguably the most popular equal area pixelization scheme in modern Astronomy. Equal area pixelization makes it easier to track pixel noise and its covariances. HEALPix has been primarily developed by the Cosmic Microwave Background community for two decades [54]. In HEALPix, the 2-sphere is tessellated into  $12n_{\text{side}}^2$  pixels where  $n_{\text{side}}$  is an integer power of 2 which defines the number of divisions along the side of a base-resolution pixel that is needed to reach a desired high-resolution partition. By considering the optimal resolution required for the radiometer analysis for the two LIGO detectors in the US, we choose  $n_{\text{side}} = 16$  which corresponds to a pixel width of  $\sim 3^\circ$  and 3072 pixels for the full sky. The advantage of using pixel basis is that one need not worry about the loss of information as compared with the analysis in spherical harmonics basis.

Figure 2 shows how a spherical harmonics basis suppresses pixels with extreme values. The top left image has more information than the spherical harmonic in the bottom right. This is obvious because the HEALPix map ( $n_{\text{side}} = 16$ ) has 3072 pixels with unique information in each pixel, but the spherical harmonics up to  $l_{\text{max}} = 15$  has only 256 components.

Furthermore, HEALPix offers ready-made tools to provide spherical harmonic transform of a pixel-space map and vice-versa, via Fast Fourier Transform (FFT) by taking advantage of isoLatitude pixelization. Hence a separate search for anisotropic SGWB in a spherical harmonic domain may become redundant. We have used here the `healpy` package, the Python implementation of HEALPix.

#### C. Posing mapmaking as a matrix multiplication

Taking advantage of the compression achieved by folding, PyStoch is explicitly calculating  $K_{fts,p}^{I_t*}$ ,  $x_{fts}^I$  for all time segments and performing Eqs. (20,21) as a matrix multiplication for a fixed frequency  $f$ . This provides much more efficiency compared to the usual practice of looping over time segments and computing each component in place. When using folded data for the sky

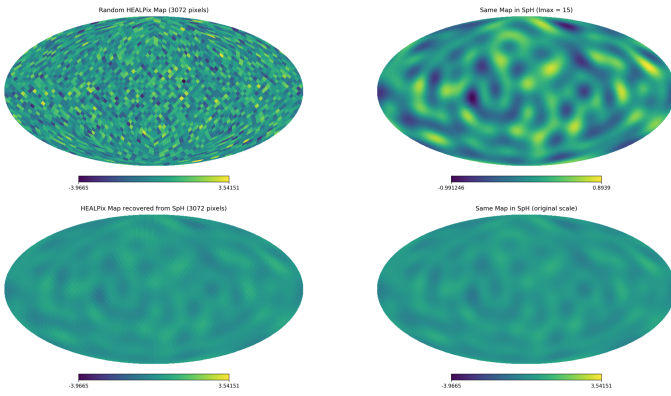


FIG. 2: Illustration of the fact that HEALPix is more efficient in detecting localized sources. The top left image is a random HEALPix map ( $n_{\text{side}} = 16$ ), the top right is the same map in spherical harmonics ( $n_{\text{side}} = 16$ ). The bottom two maps are produced from the top right map but the ‘colour scale’ is same as the top left map. The bottom left map is in HEALPix (converted from the top right) the bottom right one is same as the top right but with different colour scale. It is evident that spherical harmonics cannot retain the localized pixel information.

map making, the segment times and data lengths are predictable. Every data-segment can have previously determined start times, as the segment division is now in sidereal times. This allows pre-calculation of the overlap reduction function. Hence it reduces computational time significantly compared to the usual method of calculating the overlap reduction function (ORF) on-the-go for all available segments. This approach makes it easy to perform the analysis for a network of detectors by calculating the dirty map and Fisher matrix ( $X_p, \Gamma_{pp'}$ ) for all detector pairs in the network and adding them.

#### D. Efficient computation of overlap reduction function

Calculation and storage of ORF for a sidereal day with a segment duration of 52 sec and a bin size of 0.25 Hz requires almost 292 GB of RAM for a resolution corresponding to  $n_{\text{side}} = 16$ . PyStoch alleviates this high RAM requirement issue by introducing seed matrices for the ORF, which can be used to compute the ORF in a fast manner, alleviating the demand for an unusually large amount of memory.

Let us consider the algebraic structure of ORF. It can be seen in Eq. (11) that the ORF depends on two components, the combined antenna pattern function,  $F_{\mathcal{I}_1}^A(\hat{\Omega}, t)F_{\mathcal{I}_2}^A(\hat{\Omega}, t)$ , and the time delay,  $\hat{\Omega} \cdot \Delta \mathbf{x}_I(t)/c$ , which have no frequency dependence and they are sky maps for a particular time segment. We call these two quantities the ORF seeds, which are easy to store in the RAM and can be used to calculate the ORF for each time segment and frequency bin. The combined antenna

pattern function provides all possible combination of antenna pattern function corresponding to different detectors (in this demonstration we have used two), whereas from the phase factor expressed in the overlap reduction function one can find the time delay corresponding to each pixel. The quantity  $\hat{\Omega} \cdot \Delta \mathbf{x}_I(t)/c$  gives the time delay between two detectors in receiving a signal from a certain direction  $\hat{\Omega}$ . The process is pictorially shown in Fig. 3.

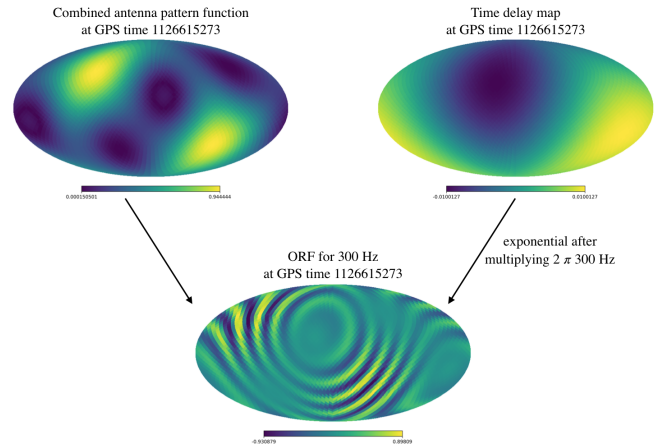


FIG. 3: The top left map is essentially the sensitivity map for a baseline for a particular time. The top right map is the time delay map for the same time. Multiplying the top right map with a frequency dependent term ( $2\pi if$ ) and taking the exponential then multiplying it pixel by pixel with the top left map we get the ORF for that time and that frequency. Here only the real part of the ORF is mapped in the bottom map. We call the top two maps ORF seeds. The ORF seeds for all available time segments are called ORF seed matrices (each row of the ORF seed matrices corresponds to ORF seeds for a particular segment).

#### E. Putting the pieces together

In this pipeline, our first step is to go through the folded data, produced by folding the entire data of LIGO’s observational run to a single sidereal day, and produce the combined antenna pattern function map for every time segment in the folded data. One can store this combined antenna pattern function maps as a matrix which has a dimension equal to the number of pixels times the number of time segments. Similarly, for the time delay maps one can calculate and store the delay between two detectors for different baselines as maps. This can be considered as calculating and storing the time delay for different time segments in a matrix having a dimension of the number of time segments times number of pixels. One can calculate and store these seed maps within no time as matrices using a small amount of RAM. This entire process of calculating and storing these quantities took only 155 MB of RAM each. As a

result, the 3314 pairs of seed maps corresponding to one sidereal day's data with 52 sec segment duration could be produced on a laptop in just 20 seconds.

From these pairs of seed maps, by considering both the combined antenna pattern function map as well as the exponential of the time delay map, for each frequency one can calculate the overlap reduction function. This can also be stored as a matrix with dimension equal to the number of frequency segments times number of time segments.

Once we have the ORF, calculation of the dirty map is straightforward. The CSD  $C_{ft}^I$  and PSD  $\sigma_{Ift}$  are stored in two different channels in stochastic intermediate data (SID) or folded SID (FSID). The CSD and PSD are in the frequency domain, where the minimum and maximum frequencies and the frequency resolution of CSD and PSD determine how many frequency bins are there. We have to loop over all the frequency bins, and each iteration of the loop will produce one narrowband map for that frequency.

In each loop, the ORF is calculated from the ORF seeds for the particular frequency corresponding to that loop. Then the columns in CSD and PSD for that frequency are multiplied with the ORF. As per Eq. (28), the result is the narrowband map for that frequency. Looping over all the frequencies gives all the narrowband maps.

#### IV. IMPLEMENTATION AND RESULTS

We have implemented the `PyStoch` code on LIGO's first observational run (O1) data from the Hanford and Livingston detectors. We have also used it for simulated data. Raw data from the detectors are down-sampled to 4 kHz and cross-correlated. Cross-correlation is done in the frequency domain for faster calculation. We then used the folding code to fold the data into one sidereal day. The folding code can take care of the overlap of the segments and data quality cuts, but the data we used already had it done during the cross-correlation. The output of folding is 3314 frames of 52 sec segment duration which span a complete sidereal day.

`PyStoch` reads the parameters which are the same for all the frames (e.g., GPS start time, frequency cutoffs, segment durations, etc.) from the first frame. Then it reads the CSD and PSD from all other frames. We had our data in a spectrum where the lower and higher cutoff was 20 Hz and 1980 Hz respectively, and it had a resolution of 0.25 Hz. The HEALPix map resolution we choose corresponds to  $n_{\text{side}} = 16$ , which uses 3072 pixels for the entire sky (each pixel covers approximately 13 square degrees of the sky). The overlap reduction function had to be calculated for all the 3314 time segments. Each of the two sets of seed matrices was hence a matrix of dimension  $3072 \times 3314$  with each element a real number. The memory consumption of the seed matrices is 156MB. On a typical laptop (2.6 GHz processor, 4 CPU threads) it takes less than 20 seconds to calculate and save the seed

matrices.

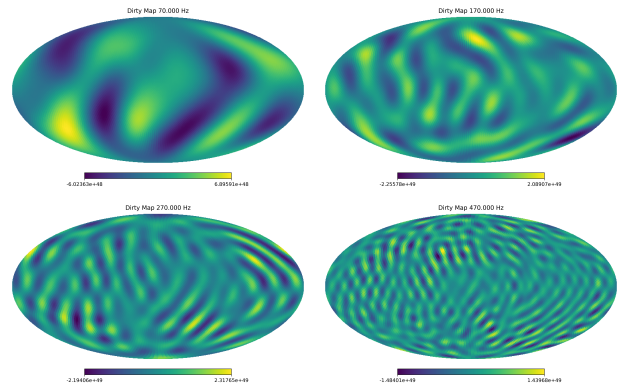


FIG. 4: Sample narrowband dirty maps from simulated data at four different frequencies 70, 170, 270 and 470 Hz are shown. The simulated data has the same statistical properties as O1 data. In these plots it can be seen that the spot sizes get smaller as the frequency goes up due to the diffraction limit.

While calculating the narrowband maps, we restricted ourselves to an upper cutoff of 500 Hz, so we had 1920 frequency bins. Then we load the CSD ( $C_{Ift}$ ) and PSD ( $\sigma_{Ift}$ ). The appropriate 1920 frequency bins are taken for the calculations so the data (CSD and PSD) are matrices of  $1920 \times 3314$ . We now have to loop over 1920 frequencies. In each loop for a particular frequency, ORF is calculated using the ORF seed matrices. Then we take one column from the CSD and PSD matrices which is of the size  $1 \times 3314$ . This corresponds to the data for that frequency and all time segments. When this data column is multiplied with the ORF, we get a  $3072 \times 1$  matrix. This matrix is the narrowband map for that frequency. In Fig. 4, 4 of the 1920 narrowband maps generated from simulated data are shown. This simulated data which we used is generated by adding a random phase factor to the O1 CSD for all frequency bins. This preserves the statistical properties of the data but renders the results unphysical.

To further validate `PyStoch`, we tested the code with some injections. The injection sources we used were broadband point sources with a flat spectrum at three different points of the sky, the strengths were two orders higher than the noise PSD. We calculated the CSD for these injected sources and added them to the CSD data. After running the code, the injected sources were properly recovered. The results from applying this injection on simulated data are shown in Fig. 5. This figure also includes a map in the bottom right which is obtained by deconvolution of the dirty map with a Fisher matrix. This was a check to see if the source strength and location in the clean map match the injected map. In this case, the point sources in the clean map appears at the same location as the injected map, thus validating the deconvolution method. Moreover, the point sources in the clean map are more localized than in the dirty map.



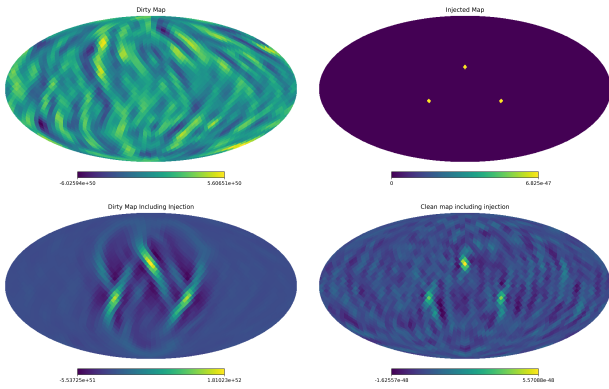


FIG. 5: The top left map is broadband dirty map from simulated data, the top right map shows the injected sources. The bottom left map is the dirty map made from simulated data including the injections. The bottom right map is the clean map obtained by deconvolution of the (bottom left) dirty map including injections.

## V. CONCLUSIONS

The primary advantage of `PyStoch` is the speed up and convenience. It makes the map calculation few hundred times faster<sup>3</sup>. Table I shows the scale of speed up by folding and `PyStoch`. It also gets rid of the requirement for storage to save intermediate results. With folding and `PyStoch` SGWB searches with LIGO data can be done on a laptop, in place of parallel computing on few hundred processors, which is very convenient.

	Conventional Pipeline	Folding Pipeline	Folding and <code>PyStoch</code>
Intermediate Data	450 GB	1.5 GB	1.5 GB
Processing Time	10 CPU years	10 CPU days	40 CPU minutes
Intermediate Results	800 TB	2.5 TB	not required
Final Results	500 MB	500 MB	500 MB

TABLE I: This table shows the estimated calculation time (on a single node of IUCAA computational facility) and storage required to calculate the narrowband maps using three pipelines: the standard pipeline, the standard pipeline with folded data and `PyStoch` with folded data.

Another advantage is that `PyStoch` produces results regarding narrowband maps. In the older pipeline one had to specify the expected SGWB spectrum  $H(f)$  before running the pipeline. But `PyStoch` does not require the

spectrum. From Eq. (28), the set of narrowband maps, the spectrum-specific result can be produced using the following equation,

$$X_p = \sum_{I_f} H(f) X_{f,p} \quad (29)$$

The above summation of narrowband maps can be done in one matrix multiplication. Also, `PyStoch` is a directed search for all directions in the sky, which means that if one wants to search a particular direction of the sky, e.g., Sco X-1 or Virgo cluster, it is straightforward. We only have to see which pixel(s) include the source and we can just add the pixels (since HEALPix pixels corresponds to the equal area in the sky one does not even have to worry about pixel weights). Similarly, `PyStoch` search results can be contracted into the result of isotropic search just by adding all the pixel values. All data quality cuts, removing bad frequencies (notch list) and correction for the windowing and the segment overlaps can be incorporated during or before running `PyStoch`.

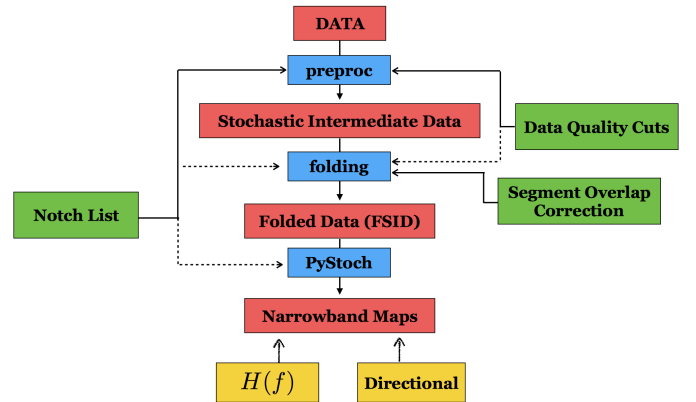


FIG. 6: Flowchart for a pipeline including folding and `PyStoch`. The interferometer data are cross-correlated in the first step `preproc` (short for pre-processing), turning the data into SID. The SID is then folded into FSID by the folding module. `PyStoch` takes the FSID and calculates the narrowband maps. Directional or spectrum specific searches can be performed on those narrowband maps. Corrections for bad frequencies (notch list), segment overlaps, data qualities can be applied at many points in this pipeline (solid arrows indicate where we applied it, dotted arrows indicate other modules where it can be applied).

The enormous efficiency achieved by folding is further enormously amplified by `PyStoch`. Folding makes the analysis few hundred times faster, `PyStoch` also makes the analysis few ten times faster. So our new algorithm for stochastic analysis (Fig. 6) is about *ten thousand* times faster than the standard pipeline, and the results are more general and versatile. `PyStoch` automatically uses multi-threading when utilized by a multi-core CPU. This can make the SGWB search even faster. Also, this new pipeline eliminate the need for intermediate data

<sup>3</sup> Tens of times faster on a single thread, hundred times faster when used with multi-threading.

storage. The results of `PyStoch` are more practical. In future, `PyStoch` will enable more in-depth searches and maps of SGWB.

`PyStoch` uses `PyCBC` routines to calculate the ORF. `PyCBC` has information about other detectors (aVIRGO, GEO600, KAGRA, etc. [8, 11, 61]) incorporated into it. This makes it very easy to tweak `PyStoch` for baselines other than the Livingston-Hanford baselines. We have tested it for many different baselines. As a toy model, let us consider the situation where VIRGO becomes as sensitive as AdvLIGO and when LIGO-India starts operating. In these scenarios, `PyStoch` can be easily used to map the SGWB using the global network of terrestrial detectors.

## VI. ACKNOWLEDGEMENT

We thank the LIGO Scientific Collaboration for access to the data and gratefully acknowledge the support of the

United States National Science Foundation (NSF) for the construction and operation of the LIGO Laboratory and Advanced LIGO as well as the Science and Technology Facilities Council (STFC) of the United Kingdom, and the Max-Planck-Society (MPS) for support of the construction of Advanced LIGO. Additional support for Advanced LIGO was provided by the Australian Research Council. We thank Dr. Andrew Matas and rest of the stochastic working group of LIGO for their help with the stochastic pipeline and their valuable comments. We acknowledge the use of IUCAA LDG cluster Sarathi for the computational/numerical work. This research benefited from a grant awarded to IUCAA by the Navajbai Ratan Tata Trust (NRTT). A. A. acknowledges the support of Council of Scientific and Industrial Research (CSIR), India. S. M. acknowledges support from the Department of Science & Technology (DST), India provided under the Swarna Jayanti Fellowships scheme.

- 
- [1] B. P. Abbott et al. (Virgo, LIGO Scientific), *Phys. Rev. Lett.* **116**, 061102 (2016), 1602.03837. [I](#)
- [2] B. P. Abbott et al. (Virgo, LIGO Scientific), *Phys. Rev. X* **6**, 041015 (2016), 1606.04856. [I](#)
- [3] B. P. Abbott et al. (VIRGO, LIGO Scientific), *Phys. Rev. Lett.* **118**, 221101 (2017), 1706.01812.
- [4] B. P. Abbott et al. (Virgo, LIGO Scientific), *Astrophys. J.* **851**, L35 (2017), 1711.05578.
- [5] B. P. Abbott et al. (Virgo, LIGO Scientific), *Phys. Rev. Lett.* **119**, 141101 (2017), 1709.09660.
- [6] B. Abbott et al. (Virgo, LIGO Scientific), *Phys. Rev. Lett.* **119**, 161101 (2017), 1710.05832. [I](#)
- [7] T. L. S. Collaboration, J. Aasi, B. P. Abbott, R. Abbott, T. Abbott, M. R. Abernathy, K. Ackley, C. Adams, T. Adams, P. Addesso, et al., *Classical and Quantum Gravity* **32**, 074001 (2015). [I](#)
- [8] F. Acernese et al. (VIRGO), *Tech. Rep. 2* (2015), 1408.3978. [V](#)
- [9] C. Affeldt et al., *Class. Quant. Grav.* **31**, 224002 (2014).
- [10] M. Ando and the TAMA collaboration, *Classical and Quantum Gravity* **19**, 1409 (2002).
- [11] K. Somiya (KAGRA), *Class. Quant. Grav.* **29**, 124007 (2012), 1111.7185. [I](#), [V](#)
- [12] G. Hobbs et al., *Classical and Quantum Gravity* **27**, 084013 (2010), 0911.5206. [I](#)
- [13] P. Amaro-Seoane, S. Aoudia, S. Babak, P. Binétruy, E. Berti, A. Bohé, C. Caprini, M. Colpi, N. J. Cornish, K. Danzmann, et al., *GW Notes*, Vol. 6, p. 4-110 **6**, 4 (2013), 1201.3621. [I](#)
- [14] Seiji Kawamura et al., *Classical and Quantum Gravity* **28**, 094011 (2011).
- [15] G. M. Harry, P. Fritschel, D. A. Shaddock, W. Folkner, and E. S. Phinney, *Classical and Quantum Gravity* **23**, 4887 (2006). [I](#)
- [16] A. Einstein, *Sitzungsberichte der Königlich Preussischen Akademie der Wissenschaften* p. 154 (1918). [I](#)
- [17] C. W. Misner, K. S. Thorne, and J. A. Wheeler, *Gravitation* (Freeman, New York, USA, 1973).
- [18] K. S. Thorne, in *300 Years of Gravitation* (Cambridge University Press, Cambridge, UK, 1987). [I](#)
- [19] T. Regimbau and V. Mandic, *Classical and Quantum Gravity* **25**, 184018 (2008), 0806.2794. [I](#)
- [20] D. Coward and T. Regimbau, *New Astron.Rev.* **50**, 461 (2006), astro-ph/0607043. [I](#)
- [21] B. Allen (1996), gr-qc/9604033. [I](#)
- [22] L. P. Grishchuk, in *Gyros, Clocks, Interferometers: Testing Relativistic Gravity in Space*, edited by C. Lämmerzahl, C. W. F. Everitt, and F. W. Hehl (2001), vol. 562 of *Lecture Notes in Physics*, Berlin Springer Verlag, p. 167, gr-qc/0002035.
- [23] M. S. Turner, *Phys. Rev. D* **55**, 435 (1997), astro-ph/9607066.
- [24] S. Dhurandhar, H. Tagoshi, Y. Okada, N. Kanda, and H. Takahashi, *Phys. Rev. D* **84**, 083007 (2011).
- [25] N. Mazumder, S. Mitra, and S. Dhurandhar, *Phys. Rev. D* **89**, 084076 (2014), 1401.5898. [I](#)
- [26] M. Geller, A. Hook, R. Sundrum, and Y. Tsai, *ArXiv e-prints* (2018), 1803.10780.
- [27] A. Jenkins and M. Sakellariadou, *ArXiv e-prints* (2018), 1802.06046. [I](#)
- [28] B. Allen and A. C. Ottewill, *Phys. Rev. D* **56**, 545 (1997), gr-qc/9607068. [I](#)
- [29] B. Allen, *Phys. Rev. D* **37**, 2078 (1988). [I](#)
- [30] A. Sesana, A. Vecchio, and C. N. Colacino, *Mon. Not. Roy. Astron. Soc.* **390**, 192 (2008), 0804.4476. [I](#)
- [31] J. D. Romano and N. J. Cornish, *Living Rev. Rel.* **20**, 2 (2017), 1608.06889. [I](#)
- [32] P. F. Michelson, *Mon. Not. Roy. Astron. Soc.* **227**, 933 (1987). [I](#)
- [33] N. Christensen, *Phys. Rev.* **D46**, 5250 (1992). [II A](#)
- [34] E. E. Flanagan, *Phys. Rev.* **D48**, 2389 (1993).
- [35] B. Allen and J. D. Romano, *Phys. Rev.* **D59**, 102001 (1999), gr-qc/9710117. [II A](#)
- [36] N. J. Cornish, *Phys. Rev.* **D65**, 022004 (2001), gr-

- qc/0106058. **I**
- [37] A. Lazzarini and R. Weiss, Internal working note LIGO-T040140-00-Z, Laser Interferometer Gravitational Wave Observatory (LIGO) (2004). **I**
- [38] S. W. Ballmer, *Class. Quant. Grav.* **23**, S179 (2006), gr-qc/0510096. **I**
- [39] S. Mitra, S. Dhurandhar, T. Souradeep, A. Lazzarini, V. Mandic, et al., *Phys.Rev.* **D77**, 042002 (2008), 0708.2728. **I, II A, II A, II A**
- [40] A. R. Thompson, J. M. Moran, and G. W. Swenson, Jr., *Interferometry and Synthesis in Radio Astronomy, 2nd Edition* (Wiley-VCH, 2001). **I**
- [41] M. Ryle and A. Hewish, *Mon. Not. Roy. Astron. Soc.* **120**, 220 (1960). **I**
- [42] E. Thrane, S. Ballmer, J. D. Romano, S. Mitra, D. Talukder, S. Bose, and V. Mandic, *Phys. Rev.* **D80**, 122002 (2009), 0910.0858. **I, II A**
- [43] J. Aasi et al., *Physical Review Letters* **113**, 231101 (2014), 1406.4556. **I**
- [44] B. P. Abbott et al., *Nature* **460**, 990 (2009), arXiv:0910.5772.
- [45] B. Abbott et al., *Phys. Rev. D* **76**, 082003 (2007), astro-ph/0703234.
- [46] J. Abadie et al., *Phys. Rev. Lett.* **107**, 271102 (2011), 1109.1809. **I**
- [47] B. Abbott et al., *Phys. Rev. D* **76**, 022001 (2007), gr-qc/0703068. **I**
- [48] C. M. F. Mingarelli, T. Sidery, I. Mandel, and A. Vecchio, *Phys. Rev.* **D88**, 062005 (2013), 1306.5394. **I**
- [49] S. R. Taylor and J. R. Gair, *Phys. Rev.* **D88**, 084001 (2013), 1306.5395.
- [50] S. Taylor, C. Mingarelli, J. Gair, A. Sesana, G. Theureau, S. Babak, C. Bassa, P. Brem, M. Burgay, R. Caballero, et al., *Phys. Rev. Lett.* **115**, 041101 (2015), 1506.08817. **I**
- [51] J. R. Gair, J. D. Romano, and S. R. Taylor, *Phys. Rev.* **D92**, 102003 (2015), 1506.08668. **I**
- [52] A. Ain, P. Dalvi, and S. Mitra, *Phys. Rev.* **D92**, 022003 (2015), 1504.01714. **I, II B, II B, II B**
- [53] E. Thrane, S. Mitra, N. Christensen, V. Mandic, and A. Ain, *Phys. Rev. D* **91**, 124012 (2015). **I**
- [54] K. M. Gorski, E. Hivon, A. J. Banday, B. D. Wandelt, F. K. Hansen, M. Reinecke, and M. Bartelman, *Astrophys. J.* **622**, 759 (2005), astro-ph/0409513. **I, III, III B**
- [55] B. P. Abbott et al. (Virgo, LIGO Scientific), *Phys. Rev. Lett.* **118**, 121102 (2017), 1612.02030. **I**
- [56] L. S. Finn, S. L. Larson, and J. D. Romano, *Phys. Rev.* **D79**, 062003 (2009), 0811.3582. **II A**
- [57] B. P. Abbott et al. (Virgo, LIGO Scientific), *Phys. Rev. Lett.* **118**, 121101 (2017), [Erratum: *Phys. Rev. Lett.* 119, no. 2, 029901 (2017)], 1612.02029. **II B**
- [58] A. Lazzarini and J. Romano, Internal working note LIGO-T040089-00-Z, Laser Interferometer Gravitational Wave Observatory (LIGO) (2004). **II B**
- [59] T. Dal Canton et al., *Phys. Rev.* **D90**, 082004 (2014), 1405.6731. **III**
- [60] S. A. Usman et al., *Class. Quant. Grav.* **33**, 215004 (2016), 1508.02357. **III**
- [61] B. Willke et al., *Class. Quant. Grav.* **19**, 1377 (2002). **V**