Detection and characterization of simulated LISA gravitational wave signals using Bayesian inference and Markov Chain Monte Carlo techniques

Théophile Cantelobre[a], David Lhuillier[b], Philippe Blanc[c]

a. Mines ParisTech theophile.cantelobre@mines-paristech.fr
b. CEA IRFU/DPhN david.lhuillier@cea.fr
c. Mines ParisTech philippe.blanc@mines-paristech.fr

Résumé

Les ondes gravitationnelles intéressent fortement astrophysiciens et cosmologists. Le détecteur Laser Interferometer Space Antenna (LISA) est conçu pour détecter les signaux gravitationnels de fréquence environ 10^{-2} Hz. Les quelques centaines de millions de systèmes binaires de naines blanches estimés contribuent leurs signaux gravitationnels dans cette bande de fréquence. Nous présentons d'abord différentes approches de la détection et la charactérization des signaux de ces sources binaires à partir de données simulées de LISA. Nous présentons alors notre implémentation en Markov Chain Monte Carlo (MCMC) pure d'une chaîne de traitement pour l'étude et la charactérization d'un signal simulé, y compris bruité.

Abstract

Gravitational waves are of great scientific interest to astrophysicists and cosmologists. The Light Interferometer Space Antenna (LISA) is a detector designed to detect gravitational signals in the 10^{-2} Hz range. It is estimated that our Galaxy comprises hundreds of millions of white dwarf binary systems that contribute gravitational signals to this range. First, we present a brief survey of approaches to the detection and characterization of these sources' signals from simulated LISA data. Then we present our approach based on pure-Markov Chain Monte Carlo (MCMC) to build a pipeline meant to study the detection and characterizing a simulated signal with simplified assumptions, but including realistic noise.

Keywords: Markov Chain Monte Carlo, signal processing, time series, gravitational waves

1 Introduction

Although first imagined by Poincaré in 1905, Einstein was the first to theorize the existence of gravitational waves in 1918 [1]. A gravitational wave is the propagation of a deformation in space-time created by a set of masses moving relatively to each other [2]. This deformation has been observed by measuring the variation of distance between two material points, as with the LIGO and VIRGO apparatus [3].

The detection of gravitational waves is interesting first and foremost because they are complementary to electromagnetic radiation to be able to observe deep, faraway stellar systems as well as the early universe. Because they do not interact with matter, gravitational waves propagate very well and are not attenuated nearly as much as electromagnetic radiation is [2].

As we've briefly introduced, gravitational waves can be detected by measuring the deformation of space-time between two material points. However the typical relative variation in distation is around 10^{-21}, about the ratio of the size of an atom, over several million kilometers [2]. The main technical problem is thus to measure extremely small deformations in space-time.

1.1 Interferometric detection of gravitational waves

Because the distance between two material points changes with the passage of a gravitational wave[1], the time for a laser beam to cross the distance between these two points is also modified. This results in a frequency shift that can be measured by having the beam interfere with another stable beam, used as a reference, whose frequency has not been modified by the interaction with a gravitational wave.

This principle has already been applied to gravitational wave detection in several Earth-based detectors, the most famous of which being LIGO and VIRGO (in the United States and Italy, respectively). They use a Michelson and Fabry-Perot-like interferometer to observe the non-isotropic variations in distance caused by the propagation of a gravitational wave [2]. LIGO detected the first gravitational wave on September 14, 2015 [4] [5] and several other detections have occurred since. These detections concern inspirals, for example the coalescence of two white dwarf stars as their orbit around one another becomes tighter and tighter. The signal lasts a few seconds and is often called a *chirp*. Because of the great variations in the gravitational field due to the coalescence, the detected signal is of "high-frequency", around 10^2 Hz.

LIGO and VIRGO are sensitive only to such high frequency signals because of the different sources of noise that such an instrument is subjected to. Although the system is designed to deal with the noise inherent to the detection sensor itself (laser shot noise, mirror stabilization, ...), it remains limited in dealing with external, low-frequency sources of noise, the most important of which is seismic noise, that runs around 1 Hz. This greatly limits LIGO and VIRGO's capacity of observing low frequency systems. As summarized by Figure 1, LIGO is sensitivite to inspiral system signals and systems outside of its sensitivity domain exist. For example, LIGO's low-frequency limit is too high to detect stable binary systems, that is systems that are orbiting far away from each other at around 1 mHz. These systems are scientifically interesting because detecting them would allow cosmologists to constrain models meant to predict the distribution of stars and such systems in the galaxy. Removing their contribution to measured deformations would also permit study of more distant and rare systems [3].

In order to detect permanent signals, far from coalescence and thus at lower frequencies, removing the Earth-induced noise in the detection chain is necessary. The LISA project proposes to this detection noise with a space-borne instrument, thereby exploring a frequency band that is scientifically rich. The following paragraphs explain the way LISA functions and why it is a pertinent solution to the low-frequency limit.

1.2 Presentation of the LISA instrument

In this section we briefly describe the design of the Laser Interferometer Space Antenna (LISA) instrument. Interested readers can consult [2] for more detailed and comprehensive descriptions and explanations.

It is difficult to attribute the paternity of the idea of a space-based gravitation wave interferometer (and it is not the goal of this work) but designs go back to the 1980s (for example, [8]).

[1] However, gravitational waves do not interact with the two points and thus their positions do not change.

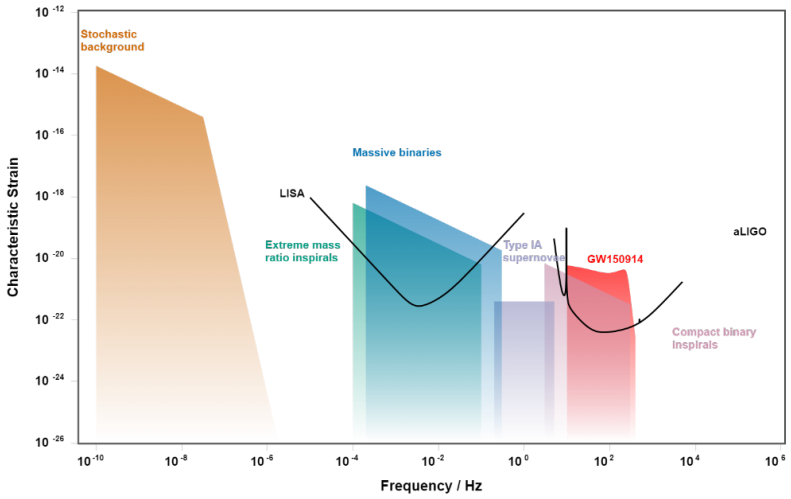


Figure 1: Sensitivity curves and observable objects. From [6], based on [7].

The successive designs combine many different scientific and technological advances of the end of the twentieth century, such as Time-Delay Interferometry (TDI) [9]. LISA is, as of this writing, still being designed and set to be launched in 2034.

LISA in composed of three identical satellites, forming an equilateral triangle whose dimension is of the order of magnitude of one million kilometers[2]. As a first approach, let us consider a pair of these satellites, A and B. Recall the objective is to measure the frequency-shift of a laser beam that traveled a distance dilated by a gravitational wave, by comparing it to a stable source. In a standard Michelson configuration, the laser beam having traveled from A to B would be compared to a beam having traveled a distance not dilated by gravitational waves. However, because of the distance separating the satellites if so large and the laser power so weak (current specifications are for 1 W), only a few photons would arrive back to A after being reflected by B, due to angular dispersion.

Of course, a few photons does not suffice to cause interference. A solution to this difficulty is to make interference between the laser beam emitted from A with a stable source in B. The instrument thus has 6 such Michelson-like configurations: each of the three arms, in both directions.

In each satellite, phase-meters measure the phase-difference, which we'll call "strain" reflecting the frequency-shift produced by the dilation of space-time induced by the gravitation wave. This signal is sampled and sent back periodically to a base station on Earth, for analysis. Due to information transfer constraints, the maximum sampling frequency is 1 Hz. As we've seen, the principle behind LISA is innovative and complex by its dimensions and the interferometric technique used (TDI). This principle was tested by LISA Pathfinder, a prototype satellite that exceeded expectations by several orders of magnitude[10].

1.2.1 Data analysis from galactic binary systems from LISA : problem statement

Once the *strain* signal is acquired by the LISA instrument, the processing chain should be able to detect gravitational signals and retrieve their characteristics: detection and estimation. The goal of LISA data analysis, is to identify gravitational signals and characterize their parameters using experimental, noisy signals. For now, simulated data is studied in place of experimental data. This section presents the problem we study in this work.

Based on population models of the Galaxy, it is estimated that there are around 60 million binary star systems, many of which may be detectable by LISA [3]. As one can imagine, the characteristics of a galactic binary system are reflected in the resulting gravitation wave strain signal, hereinafter called *strain*. Estimating the strain signal's parameters may give a lot of information about the source. For example, for reasons of symmetry, the instantaneous frequency of the strain signal is related to twice the orbital frequency of the galactic binary system [11].

We can therefore summarize the objectives of LISA data analysis as:

[2]This parameter has evolved over time and is set to 1.5 million kilometers. The possibility of having such long arms is also key to LISA's value proposition as they fix the instrument's high frequency detection limit [2].

– Separation of the strain signals of the different sources, because the strain signal may be the result of the millions of sources in the galaxy,

– Estimation of each source's parameters,

– Estimation of the uncertainty in this estimation.

Formally, consider a set of N gravitational signal sources. Let $(h_i(t))_{1\leq i\leq N,t}$ be the LISA instrument's response (the strain signal) to source i's gravitational signal, over a year. Since all the systems are white dwarf binary systems, the observed strain signal only depends on the system's characteristics. Thus, we can rewrite $h_i(t) = h(t, \vec{\lambda}_i)$ where $h : (t, \vec{\lambda}) \mapsto h(t, \vec{\lambda})$ is the strain parametric waveform model with $\vec{\lambda}$ the parameters.

By linearity [2], the instrument's total response to the N gravitational wave signals is, for all t:

$$s(t) = \sum_{i=1}^{N} h(t, \vec{\lambda}_i) \tag{1}$$

Until now, we've assumed that the instrument's response to the gravitational signal is without noise. Although the goal of using a space instrument was to eliminate the most bothersome noise sources (such as low-frequency seismic noise) there remain many sources of noise to take into consideration. The origins of these different noises are detailed in [2] and [12]. By linearity, we can consider that these different noises sum to $(n(t))_t$ ([12] provides a noise model by estimating the noise Power Spectral Density (see Section 3.2)), such that the observed strain, called *signal*, is in fact:

$$s(t) = \sum_{i=1}^{N} h(t, \vec{\lambda}_i) + n(t) \tag{2}$$

The problem is thus to detect the presence of gravitational sources in the noisy strain signal, estimate their number N, and, for each, estimate their characteristics $\vec{\lambda}_i$, $1 \leq i \leq N$.[3] from an observation of the strain signal s sampled every T_s with total time T (one year).[4]

1.3 Goals

The goals of this paper are:

– To briefly review the existing approaches to the above problem of LISA data analysis (in Section 2).

– To present a fundamental approach to the data analysis based on the following steps we developed at CEA IRFU/DPhN in order to gain better understanding of the underlying principles of the data analysis (in Section 3:

  · The creation of a library of reference waveforms that can be used to test different data analysis approaches, as well as the generation of noise realizations to test the pipeline for realistic cases.

  · The implementation of a state-of-the-art retrieval algorithm based on Markov Chain Monte Carlo (MCMC) approach. This approach, although promising, raises computational problems related to computational cost of sampling the parametric waveforms. In this paper, we propose a solution to this problem.

– To present and discuss results and perspectives for future work (in Section 4).

[3] We present what these characteristics are in Section 3.

[4] In order to better understand the problem at hand, one might consider the following analogy. Imagine buying a ticket to a philharmonic concert, knowing the tune the orchestra is to be playing but being blindfolded. It turns out that the conductor is absent so the musicians all play on their own. Furthermore, all through the concert, your neighbors are noisy: they chat away all through the concert. The problem we are dealing with, also called the *cocktail party problem* is analogous to determining the number of musicians that are present, which instrument they are playing, and the characteristics of their instruments.
This analogy is our own but we learned that ESA had a similar analogy on its website : see [13].

We restrict our analysis and work to the following simplified problem :

$$s(t) = h(t, \vec{\lambda}) + n(t) \tag{3}$$

This restriction remains relevant as it is sufficient to study the whole pipeline, from waveform generation to detection and parameter estimation. The separation of sources is out of scope of this work. Nevertheless, the literature does presents several methods of separation that are used today and one of these is briefly presented in Section 2.2.3.

2 Review of existing data analysis techniques based on Bayesian techniques and MCMC

The goal of this section is to present the LISA data analysis process by reviewing existing literature. Instead of reviewing general-purpose statistical signal processing litterature, we've chosen to present the formalization of the problem, the techniques and tools usually used in the context of gravitational wave detector analysis. We've chosen to focus on the analysis of LISA data rather than LIGO and VIRGO data analysis although there are no hard frontiers between the two concepts. Finally, we've chosen to spend more time on the main techniques and only mention some of the many tuning options and optimizations that the community has come up with as this is better aligned with the general goal of this work.

2.1 Bayesian modeling

Recall that the goal of this work, and more generally of LISA data analysis, is to identify gravitational signals and characterize their parameters using acquired, noisy signals.

One cannot definitively "eliminate" the noise from the data and identify the resulting signals as gravitational signals. The solution to this conundrum is to evaluate the different probabilities of different hypotheses.

The necessary framework to these considerations is that of the Bayesian approach. [3] presents such a framework: the acquisition of a LISA signal s can be seen as a stochastic process that could obey to a certain model $\mathcal{M}$ which depends on certain parameters $\vec{\lambda}$. Bayes theorem gives us :

$$p(\vec{\lambda}|s, \mathcal{M}) = \frac{p(s|\vec{\lambda}, \mathcal{M})p(\vec{\lambda}|\mathcal{M})}{p(s|\mathcal{M})} \tag{4}$$

We can thus evaluate the probability that a given set of parameters summarize the data in accordance to model $\mathcal{M}$. We can then compare different models or different parameter vectors. This can allow us to choose between different waveforms (match filtering, as this is done on LIGO and VIRGO data, see the estimated computational cost in [14]) and different parameters. That is, to compare whether $\vec{\lambda}$ of $\vec{\lambda}'$ best summarize the data, we can compute the ratio :

$$\frac{p(s|\vec{\lambda}', \mathcal{M})}{p(s|\vec{\lambda}, \mathcal{M})} = \frac{p(\vec{\lambda}|s, \mathcal{M})p(\vec{\lambda}|\mathcal{M})}{p(\vec{\lambda}'|s, \mathcal{M})p(\vec{\lambda}'|\mathcal{M})} \tag{5}$$

Let us identify the different terms of this ratio:

- $p(\vec{\lambda}|s, \mathcal{M})$ is the posterior distribution of the parameters. Informally, this distribution contains the information about the probability of different parameters given the measured signal.
- $p(\vec{\lambda}'|\mathcal{M})$ is the prior distribution of the parameters. It summarizes our prior knowledge of the parameters, based on the model.
- finally, $p(s|\vec{\lambda}, \mathcal{M})$ is the likelihood of the signal given the parameters[5].

In order to determine the most probable set of parameters we must calculate the mode of the posterior distribution. However, nothing guaranties that this posterior distribution is analytically known, or that its maximum can be easily computed analytically or even, numerically.

In order to solve this problem, stochastic sampling techniques are used. The next section presents the technique generally used in the LISA data analysis community to sample the posterior distribution: Markov Chain Monte Carlo (MCMC) methods.

[5] As we are keeping one model of waveform, throughout our analysis, we drop the dependance on the model in our explanations, and in our formalism.

2.2 Sampling techniques : MCMC, in many forms

One solution to the sampling problem is that of Markov Chain Monte Carlo (MCMC) methods. In this section, we present this algorithm in its general form before exploring the different implementations used by the LISA community.

2.2.1 Markov Chain Monte Carlo method motivation

The Markov Chain Monte Carlo sampling method relies on the stationary property of Markov chains by observing the realizations of the chain over time. The idea is thus to build a Markov chain that has the same sampled distribution as its *stationary distribution*. There is only one parameter that can influence the chain's stationary distribution: the transition kernel we use.

Recall that Markovian property of Markov chains gives :

$$p(x_{n+1} = y|x_0, ..., x_n) = p(x_{n+1} = y|x_n) \tag{6}$$

where $p(y|x)$ is the transition kernel. In our case, the transition kernel is unknown. As [15] outlines, the Metropolis-Hastings method for MCMC ensures that by replacing the transition kernel by a function of the likelihood, prior and proposal distributions, the resulting chain becomes markovian, with its stationary distribution corresponding to the posterior distribution.

Thus, the probability of transitioning from a point $\vec{x}$ to a point $\vec{y}$ in the parameter space is defined as $\min(1, H_{\vec{x}\to\vec{y}})$ where H is the *Hastings ratio* defined as

$$H_{\vec{x}\to\vec{y}} = \frac{p(\vec{y})p(s|\vec{y})q(\vec{x}|\vec{y})}{p(\vec{x})p(s|\vec{x})q(\vec{y}|\vec{x})} \tag{7}$$

where

- $p(\vec{x})$ is the *prior distribution* associated to $\vec{x}$
- $p(s|\vec{x})$ is the *likelihood* function
- $q(\vec{y}|\vec{x})$ is the *proposal distribution*, that is, the distribution characterizing how likely a jump from $\vec{x}$ to $\vec{y}$ is.

The algorithm is summarized in Algorithm 1.

Data:

- LISA strain signal: s
- Proposal ("jump") distribution $\mathcal{P}$
- Number of iterations: N
- Initial parameters: $\vec{\lambda}_0$

Set $\vec{\lambda} = \vec{\lambda}_0$;
Set current iteration $i = 0$;
while $i < N$ **do**
  Jump following $\mathcal{P}$: $\vec{\lambda}' = \vec{\lambda} + \vec{j}$ where $\vec{j} \sim \mathcal{P}$;
  Calculate Hastings ratio: $H = \frac{p(\vec{\lambda}')p(s|\vec{\lambda}')q(\vec{\lambda}|\vec{\lambda}')}{p(\vec{\lambda})p(s|\vec{\lambda})q(\vec{\lambda}'|\vec{\lambda})}$;
  Get random number r between 0 and 1;
  if $r < \min(1, H)$ **then**
    Accept jump: $\vec{\lambda} = \vec{\lambda}'$;
  else
    Reject jump: $\vec{\lambda} = \vec{\lambda}$;
  end
end
Return $\left(\vec{\lambda}_0, ..., \vec{\lambda}_{N-1}\right)$;

Algorithm 1: Markov Chains Monte Carlo (general case).

It is important to note that we have described the algorithm in all generality without taking into consideration the choice of *prior*, *likelihood function*, and *proposal distribution*. This is the object of the following section.

2.2.2 Choice of *prior*, *proposal* and *likelihood*

Here, we adress the choice of *prior*, *proposal* and *likelihood* by the LISA community. We do not cover every choice made but describe the general directions that the community's research has taken. The *prior distribution* should contain all prior information we have about our parameters. In other contexts, this distribution would be very important, but we have no prior information about the parameters of the gravitational source, such as sky position. In the community, uniform priors are generally considered [15] [16]. This simplifies the Hastings ratio to the *likelihood* and the *proposal* distributions.

Any choice of *proposal distribution* is admissible and will allow the algorithm to sample the *posterior* (or *target*) *distribution* correctly. However, it can influence the speed at which the chain samples the space and how many accepted jumps are made. It is generally chosen to be normal with a given σ jump size. The choice of σ is often made using the Fischer information matrix, following for example [15], [3] and [16]. However, these three works do not keep a constant proposal distribution. For example, they use a uniform proposal distribution from time to time, mixing between bolder and more timid proposals. This accelerates the MCMC burn-in phase (the iterations needed before the chain is Markovian) and helps the chain explore the whole parameter space. We'll return to one such acceleration technique in the next section, as they play an important role in existing techniques.

In the case of a normal *proposal distribution*, the distribution q is symmetrical and thus, the Hastings ratio further simplifies to the ratio of the likelihoods.

The choice of likelihood is fixed if we consider the noise to be white and independent and identically distributed. In order to account for the color of the noise[2], we scale the likelihood function with $S_n(f)$, the noise's Power Spectral Density, as is defined below. Formally, the following discrete scalar product is introduced in the Fourier domain:

$$< a, b >= \frac{2}{T} \sum_{\alpha} \sum_{f} \frac{\tilde{a}^*_\alpha(f)\tilde{b}_\alpha(f) + \tilde{a}_\alpha(f)\tilde{b}^*_\alpha(f)}{S^\alpha_n(f)} \tag{8}$$

where

- $\tilde{a}$ is the Fourier transform of a
- $\tilde{a}^*$ is the complex-conjuguate of the the Fourier transform of a
- α is an interferometric channel, between the three possible LISA channels.
- $S^\alpha_n(f)$ the noise power spectral density at frequency f (and along channel α).

The likelihood function is thus [6]

$$p(s|h(\vec{\lambda})) = C \exp\left(-\frac{||s - h(\vec{\lambda})||^2}{2}\right) \tag{9}$$

Using the polarisation formula of scalar products, we derive the relative likelihood function by dividing by the null hypothesis (how likely is it to have measured the given signal given that there are no gravitation signals present, only noise?) as :

$$\mathcal{L}(\vec{\lambda}) := \frac{p(s|h(\vec{\lambda}))}{p(s|0)} = \frac{C \exp\left(-\frac{||s-h||^2}{2}\right)}{C \exp -\frac{||s||^2}{2}} = \exp\left(\frac{< s, h(\lambda) > -||h(\vec{\lambda})||^2}{2}\right) \tag{10}$$

This is the form we'll use throughout this paper (as well as its natural logarithm the *log-likelihood*)[7]. We'll see that the change from likelihood to relative likelihood has (1) no influence on the MCMC algorithm as we are interested in the quotient of likelihoods, as in Equation 7, and (2) speeds up the calculation as the $||s||^2$ term is not calculated at each iteration. Other works consider the inverse relative likelihood (and its natural logarithm) [3], for example.

[6] We do not differentiate between h the binary system waveform and h the sum of the waveforms.

[7] Both the *relative likelihood* and *log-relative likelihood* are often referred to as *likelihood*. In this work, unless noted otherwise, likelihood refers to the relative likelihood and log-likelihood to the log-relative-likelihood.

2.2.3 Source separation

As we've explained, the measured signal contains the superposition of millions of gravitational signals from binary systems, as well as of noise. Separating the signals from the different sources is a open problem. One such approach is to include the complexity of the representation in the parameters, through Bayes' factor [3]. This approach uses the Reverse Jump Monte Carlo Algorithm to switch between models with more or less sources, favoring more complex models only if they significantly favor the data[8]. For the sake of practicality, the frequency space is binned and a certain number of sources, those whose frequencies are in that bin, are separated in a given bin. They are then subtracted from the remaining data. This reduces the parameter-space and limits the number of waveform generations that are done.

However, as MCMC algorithms are sampling algorithms, the number of iterations needed is very large. Also, the MCMC algorithm can become "stuck" on a local maximum of the likelihood function, that is be "stuck" sampling a secondary mode of the posterior distribution. In this case, increasing the number of iterations should in theory improve the sampling. In order to provide an effective solution to this issue, an *acceleration* technique is used.

2.2.4 MCMC acceleration procedure : simulated annealing and parallel tempering

Several papers have successfully combined simulated annealing and parallel tempering to provide accurate posterior sampling even when the problem presents computational difficulties [16] [3].

The principle behind simulated annealing is to "heat" the likelihood surface to make it easier for the chain to sample different parts of the parameter space. Concretely, we replace the likelihood ratio H with H^β where $0 \leq \beta \leq 1$ represents the inverse of the temperature. At high-temperature (low β), H^β is close to 1 and the surface is softened: the chain easily samples vast regions of the parameter space. The chain is gradually cooled ($\beta \to 1$) and samples (ideally) the mode of the posterior distribution.

In order to avoid having the chain become stuck on a secondary mode, the *parallel tempering* technique is used: several chains are heated and cooled. Parameter sampling is then shared between the chains. [16] couple this approach with the Reverse Jump MCMC (RJMCMC) source separation approach.

3 Study of a simplified pipeline.

This work presents the simplified data analysis pipeline we developed at CEA IRFU/DPhN. It implemented a simple Metropolis-Hastings MCMC algorithm to sample the posterior distribution with three parameters. The objective is to develop a simplified LISA data analysis process chain (called test pipeline) from the simulation of the gravitational signal to its dectection and parameter retrieval, in order to better understand the underlying occurring phenomena. Thus, all sampling functions are "handmade" and do not use any of the state-of-the-art improved and sophisticated techniques defined above.

The construction and study of this pipeline took several directions, some of which are presented here: the generation of the model waveform, the generation of a noise realization, the implementation of an MCMC sampler and finally construction of `Jupyter Notebooks` for testing and analyzing the results.

3.1 Model generation

As discussed, the analysis technique we use relies on the waveform model h derived from physical modelling. Thus, we used the waveforms provided in the astrophysical literature for binary systems to build our model generation module. These waveforms are summarized in [15] and are not entirely reproduced here.

It is relevant however to note that for a single channel, the resulting strain waveform can be written as :

$$h(t) = \sum_{i=1}^{4} a_i(A, \psi, \iota, \varphi_0) A^i(t, f, \theta, \phi) \tag{11}$$

[8] In [3], the threshold is arbitrarily placed at 12 times more favorable.

where

- A is the strain amplitude
- ψ, ι give the direction of the system's kinetic moment
- φ_0 is the phase origin
- f is the system's frequency[9]
- and finally, θ and ϕ give the sky position of the source.

This separation between the parameters that do not depend on the source's characteristics, the extrinsic parameters (A, ψ, ι, φ_0), and those that do, the intrinsic parameters (f, θ, ϕ), will be important when we discuss computational issues linked to generating these waveforms.

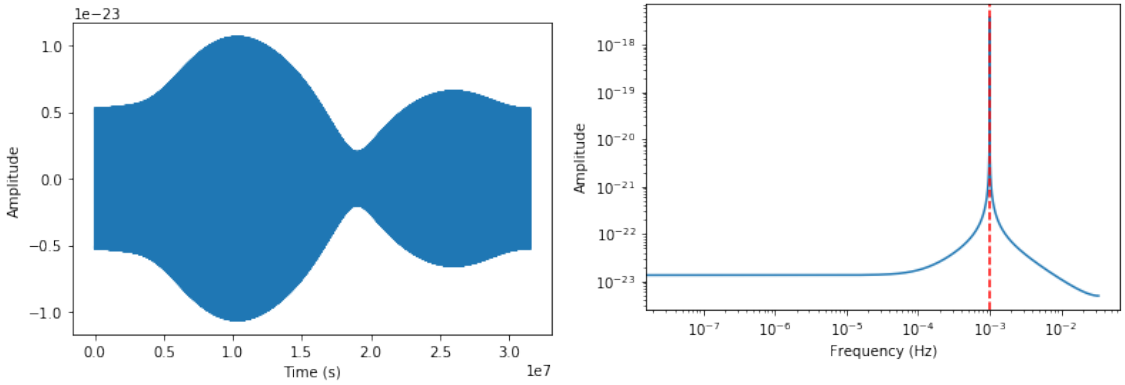
The `Python` model generation module successively builds phase

$$\phi(t) = 2\pi f t + 2\pi t AU \sin\theta \cos\left(2\pi f_m t - \phi\right) \tag{12}$$

then the A_i time series combined into the final time series h. This is done along two linear combinations A and E, that correspond to the two "useful" channels, those that contain gravitational signals.

3.1.1 Module validation

The model generation module has been validated in two ways: first, comparing the envelope modulation to that documented in [17]; then, verifying that the Fourier transform of the signal is coherent with the form of the phase provided in Equation 12. Examples of such plots are shown in Figure 2. Furthermore, our simple numerical experiments have proved consistent with previous initiatives documented in [17].



(a) Gererated waveform amplitude on A channel.

(b) Fourier transform of the generated A channel at 1 mHz.

Figure 2: Model waveform generation validation.

3.1.2 Computational difficulties & possible solutions

As the waveforms are generated over one year, computational difficulties are at the heart of an effective module. As we'll see they severely limit the potential of numerical experimentation, but they also make implementing the module difficult as RAM can be quickly filled up. A table of double float containing the time index and the three components X, Y and Z that are then combined to make A and E channels take up approximately 360 MB of memory. Since these X, Y, and Z components are themselves linear combinations of 4 different components each, memory use quickly adds up and simultaneously creating several different waveforms becomes tedious, if not impossible. Care was taken in optimizing memory use and would still need to be improved in order to allow for more complex and full implementations of the LISA data analysis pipeline.

[9] In some cases, $\dot{f}$ is considered non-null, but not here.

One such solution is that of lowering the sampling rate. Today, the usual sampling frequency is 66 mHz (15 s), significantly larger than the 20 mHz theoretical Shannon-Nyquist frequency taken into consideration mHz gravitational waves. As we tested our pipeline with a strain signal with a frequency of 1 mHz, whose maximum significant frequency are lower than 20 mHz using a much lower sampling frequency (6.6 Hz, 150 s) sped up our calculations by the same factor, and used much less memory. It is possible to push this type of consideration, looking not only at the maximum frequency but also at the minimum positive frequency. Exploiting narrow bandwdith gravitational wave signals can significanlty reduce the effective sampling rate.

3.2 Noise generation

As we've seen, the MCMC algorithm samples a posterior distribution that describes the parameters' probabilities of describing the data by comparing a model signal to the data signal. It is paramount to try to study realistic data signals because real-world data will be very "noisy". Furthermore, as we've seen, the *Power Spectral Density (PSD)* of the noise intervenes in the scalar product that defines the likelihood function on which the entire pipeline depends. It weights the different time samples according to the confidence we have in them, using the noise PSD as a metric. The noise PSD is therefore a hypothesis in our data analysis pipeline. This is why we chose to generate noise that closely resembles the analytical noise predictions. In order to create a mock-data signal, we then add the generated noise to a model waveform, whose parameters we'd like to estimate.

The desired PSD is hypothesized to be, analytically [12]:

$$S_n(f) = 16\sin(2\pi L_0 f)\left(S_{opt}f^2 + (3+\cos(4\pi L_0 f)S_{acc}\left(1+\frac{10^{-4}}{f}\right)\frac{1}{f^2}\right) \tag{13}$$

where

- the S_{opt} term is the optical noise
- and the S_{acc} term is the acceleration noise.

The *shot* noise is reduced by the TDI technique [2].

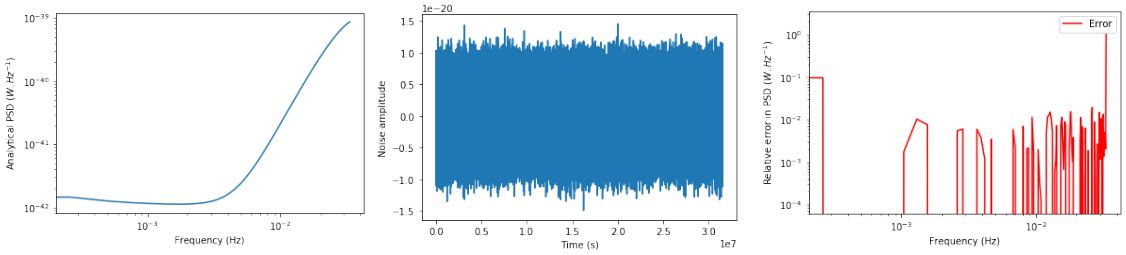
The goal of the noise generation module is thus to generate a realization of noise with the desired PSD. The idea behind the technique is to "color" a realization of white Gaussian noise, using the desired PSD. For this, we use the transfer function defined by the PSD and convolve it with the white noise realization.

Formally, we first define the following quantities:

- The noise auto-correlation function: $R_{nn}(\tau) = \mathbb{E}_t\left(n(t)n(t-\tau)\right)$
- The noise PSD $S_n(f) = \tilde{R}_{nn}(f)$ as the Fourier-transform of R_{nn}.

Thus, if we define $\tilde{H}(f) = \sqrt{Sn(f)}$ and $n_c = n_w \star H$, we have $S_{n_c}(f) = S_n(f)$.

We verified this numerically (Figure 3) by generating a noise realization, then estimating its PSD using the Welch method, using the `scipy.signal.welch` function in `Python`. The Welch-calculated PSD and the analytical $S_n(f)$ are indistinguishable when plotted. Figure 3c shows the relative PSD error. Because we stay below 1% error (except at the edges of the frequency domain), we've validated our colored noise generation process.



(a) Analytical $S_n(f)$ [12]. **(b)** Generated colored noise. **(c)** Relative PSD error.

Figure 3: Noise generation validation.

3.3 Metropolis-Hastings MCMC parameter estimation

As we outlined in the survey of existing techniques in Section 2.2, state-of-the-art pipelines use complex implementations of the Markov Chain Monte Carlo (MCMC) algorithm that we presented. Theses implementations attempt to solve many of the problems raised by sampling algorithms and MCMC in particular.

The longer-term goal of our work is to study the dimensioning characteristics of the data in order to better develop algorithms to analyze it. For example, the effects of gaps and glitches in data which LISA Pathfinder showed would be numerous and important to take into consideration [10]. It is thus natural to start with a pure Metropolis-Hastings MCMC implementation. This is presented in this section.

In accordance with the previous study [15] we chose uniform priors and a Gaussian proposal distribution. Because it is an even distribution, it does not contribute to the Hastings ratio which simplifies to the ratio of the likelihoods. Algorithm 2 schematically presents our implementation of the pure-MCMC algorithm. Implementation-wise, the algorithm was developed using `Python` and used from `Python Jupyter` notebooks.

As with all numerical computation, run-time is always paramount, and raises major issues in our present case. We touch on this problem in Section 4.

Data:

- Source parameters: $\vec{\lambda}$
- Analytical noise PSD: $S_n(f)$
- Jump sizes: $\vec{\sigma}$
- Number of iterations: N
- Initial parameters: $\vec{\lambda}_0$

Generate model waveform $h(\vec{\lambda}_0)$;
Generate noise realization n;
Set $s = h(\vec{\lambda}_0) + n$;
Set $\vec{\lambda} = \vec{\lambda}_0$;
Set current iteration $i = 0$;
while $i < N$ **do**
  Jump following $\mathcal{N}(\vec{\lambda}_i, \vec{\sigma})$: $\vec{\lambda}' = \vec{\lambda}_i + \vec{j}$ where $\vec{j} \sim \mathcal{N}(0, \vec{\sigma})$;
  Calculate Hastings *log-ratio*: $\ln H = l(\vec{\lambda}') - l(\vec{\lambda}_i)$;
  Get random number r between 0 and 1;
  if $< \min(0, \ln H)$ **then**
    Accept jump: $\vec{\lambda}_{i+1} = \vec{\lambda}'$;
  else
    Reject jump: $\vec{\lambda}_{i+1} = \vec{\lambda}_i$;
  end
end
Return $\left(\vec{\lambda}_0, ..., \vec{\lambda}_{N-1}\right)$;

Algorithm 2: Pure MCMC implemented algorithm.

4 Results & analysis

The first part of this section presents one the results we obtained when validating the pipeline on a concrete case. It illustrates that the method is robust to noise, even when the noise amplitude is orders of magnitude over that of the signal. The second part of the section presents the inquiry necessary to build on our work. This ranges from Markov chain introspection and tuning to more advanced analysis methods.

In the result presented here, we fixed all of the parameters on their correct values except: f, β, and λ (the frequency and sky position parameters). The parameters we try to retrieve are the

following: $f_{key} = 1.10^{-3}$ Hz, $\lambda_{key} = 5.199$, $\beta_{key} = 0.474$ (the others are, for reference: $\varphi_{0_{key}} = 0$, $\psi_{key} = 0$, $\iota_{key} = 0$, $h_{0_{key}} = 3.00.10^{-22}$). We run our algorithm for $10,000$ iterations, with sampling at 150 seconds. The jump sizes we use are: $\sigma_f = 1.59.10^{-7}$ Hz, $\sigma_\lambda = 0.05$ and $\sigma_\beta = 0.1$. These are chosen to match the approximate dispersion of the likelihood functions; as we've mentioned in the survey section of this paper, other ways of choosing $\sigma_{\vec{\lambda}}$ are possible. The starting values used are: $f = f_{key} + \frac{1}{2\pi}10^{-8}$, $\lambda = 5.1$ and $\beta = 0.4$. These starting values are quite close to the *key* values. This does not invalidate the numerical validation of the algorithm: as we've seen, other techniques are used to deal with sampling over a broad parameter space.

The goal here is to check that the sampling is reasonable and to check the validity of different values as statistical estimators for the parameter values. Here we focus on the mean of the sampled values as well as the maximum value on the histograms of the different parameters. We could have studied different indicators such as the median, as well as the form of the sampled posterior.

Figures 4, 5 and 6 present the sampling results for the β, λ and f parameters respectively. The figures to the left show a scatter plot of log-likelihood as a function of each parameter. They show that many jumps do not increase the likelihood and the algorithm is *not* a hill-climbing algorithm, but is sampling the parameter space.

On the figures, μ and σ are empirical mean and standard deviation. In the case of β (see Figure 4), we observe that the mean is equal to the key value. Furthermore, these two values are outside the histogram's maximum bin. For λ, the mean value is in the maximal bin but the key value is a bin away. Finally, for f, the mean and key values are coincident in the maximal bin. However, the histogram does not seem mono-mode.

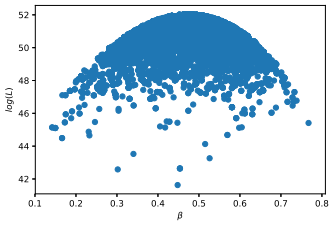
We do not comment and detail the statistical significance of these *errors* and anomalies for the reasons explained in the next paragraph.

Perspectives in MCMC and analysis An important remark is that the presented histograms only concern a single parameter and thus are marginalized over the two other dimensions. There is *a priori* no reason that the maximal bins in the cases of the different parameters correspond to the maximum argument of $p(\lambda|s)$. Our interpretation is therefore quite tenuous and further investigation into the analysis of the resulting sample is necessary. One direction of inquiry would be to use Kernel Density Estimation, the generalization of a histogram for density estimation. However, this raises major computational difficulties: the number of points would need to be much more important in order for a reasonable number of bins to be used in this non-separable, multi-dimensional problem.

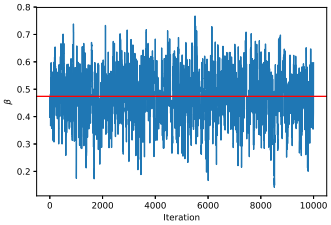
In order for this further work to be undertaken, care is to be given to the efficiency of the waveform generation step, especially when several sources are to be identified. Our module takes 0.5 s to compute the waveform on a standard laptop, when sampling at one tenth of the original 15 seconds.

Also, this work does not cover the examination of the generated Markov chain's properties, such as mixing, variance, ... Further work on this, eventually with *state-of-the-art* samplers, should allow us to better our results, and better understand the pertinence of this pipeline.

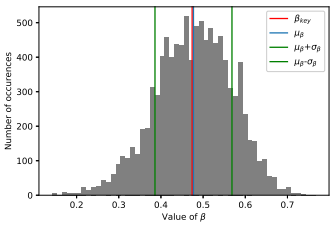
Finally, in order to better understand the uncertainties in our estimation, work on exploring the numerical effects of the Discrete Finite-time Fourier transform, in the absence of apodization windows, seems necessary.



(a) Scatter-plot: $\log(\mathcal{L})$ vs β

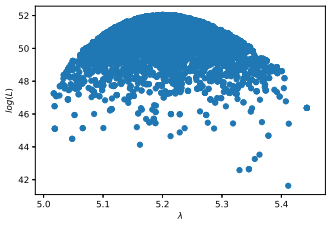


(b) β values through iterations.

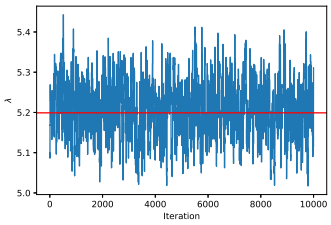


(c) β histogram. Reference values shown.

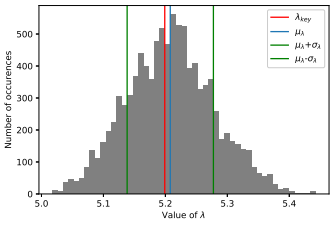
Figure 4: β sampling with generated noise (10000 iterations).



(a) Scatter-plot: $\log(\mathcal{L})$ vs λ

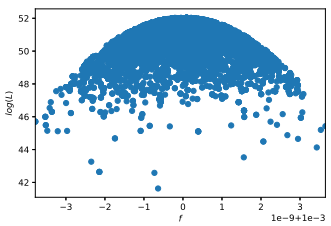


(b) λ values through iterations.

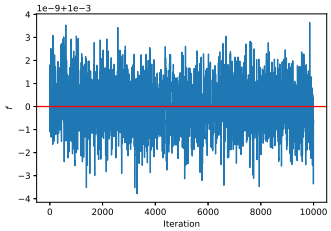


(c) λ histogram. Reference values shown.

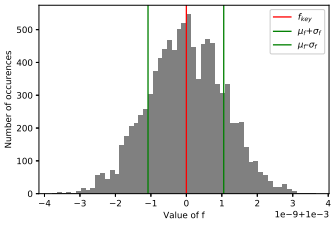
Figure 5: λ sampling with generated noise (10000 iterations).



(a) Scatter-plot: $\log(\mathcal{L})$ vs f



(b) f values through iterations.



(c) f histogram. Reference values shown.

Figure 6: f sampling with generated noise (10000 iterations).

5 Conclusion

This report records and synthesizes our first excursion in LISA data analysis algorithms. We chose a pure-MCMC based approach to understand the way state-of-the-art pipelines are built. The principle behind the Bayesian-MCMC pipeline has been validated on a simple problem with three free parameters in a separable way on a single gravitational source, with generated noise with realistic color. The difficulties encountered lie in: the computational efficiency and the interpretation of the resulting distribution. The perspectives in overcoming these difficulties lie in a deepened study of signal sampling frequencies, density estimation techniques and uncertainties, Markov chain introspection, and making implemented computations more efficient.

Acknowledgements

This report is the result of the research internship I had the honor of pursuing from September 2018 to January 2019, under the direction of David Lhuillier, at Commissariat à l'énergie atomique et aux énergies alternatives (CEA)'s Institut de Recherche sur les lois Fondamentales de l'Univers (IRFU) in the Nuclear Physics Department (DPhN). I'd like to thank CEA and the DPhN for their welcome, financing and logistical assistance. Special thanks to Frank Sabatié and Danielle Coret for their support.

My warm thanks to David, Philippe, Hervé Moutarde, Jérôme Bobin, Andry Raktozafindrabe, Antoine Petiteau, and Adrien Blanchet, for your scientific advice, help and discussion. Thank you to all the *précaires* at DPhN: I hope you all the best with the rest of your PhDs, post-docs, and internships!

Special thanks to Jordan Nicoules for providing some of the building blocks for the pipeline and for sharing your experience!

References

[1] A. Einstein, "Über Gravitationswellen," *Sitzungsber. Preuss. Akad. Wiss. Berlin (Math. Phys.)*, vol. 1918, p. 154, 1918.

[2] A. Petiteau, *From simulation of LISA to data analysis.* Theses, Université Paris-Diderot - Paris VII, June 2008.

[3] T. B. Littenberg, "Detection pipeline for Galactic binaries in LISA data," *Physical Review D*, vol. 84, Sept. 2011.

[4] P. Naselsky, A. D. Jackson, and H. Liu, "Understanding the LIGO GW150914 event," *Journal of Cosmology and Astroparticle Physics*, vol. 2016, pp. 029–029, Aug. 2016. arXiv: 1604.06211.

[5] T. L. S. Collaboration and the Virgo Collaboration, "Observation of Gravitational Waves from a Binary Black Hole Merger," *Physical Review Letters*, vol. 116, Feb. 2016. arXiv: 1602.03837.

[6] "Gravitational Wave Sensitivity Curve Plotter." http://gwplotter.com/.

[7] C. J. Moore, R. H. Cole, and C. P. L. Berry, "Gravitational-wave sensitivity curves," *Classical and Quantum Gravity*, vol. 32, p. 015014, Jan. 2015. arXiv: 1408.0740.

[8] J. E. Faller, P. L. Bender, J. L. Hall, D. Hils, and M. A. Vincent, "Space antenna for gravitational wave astronomy," vol. 226, Apr. 1985.

[9] M. Tinto and S. V. Dhurandhar, "Time-Delay Interferometry," *Living Reviews in Relativity*, vol. 8, Dec. 2005.

[10] M. Armano, H. Audley, J. Baird, P. Binetruy, M. Born, D. Bortoluzzi, E. Castelli, A. Cavalleri, A. Cesarini, A. Cruise, K. Danzmann, M. de Deus Silva, I. Diepholz, G. Dixon, R. Dolesi, L. Ferraioli, V. Ferroni, E. Fitzsimons, M. Freschi, L. Gesa, F. Gibert, D. Giardini, R. Giusteri, C. Grimani, J. Grzyh, I. Harrison, G. Heinzel, M. Hewitson, D. Hollington, D. Hoyland, M. Hueller, H. Inchauspé, O. Jennrich, P. Jetzer, N. Karnesis, B. Kaune, N. Korsakova, C. Killow, J. Lobo, I. Lloro, L. Liu, J. López-Zaragoza, R. Maarschalkerweerd, D. Mance, N. Meshksar, V. Martín, L. Martin-Polo, J. Martino, F. Martin-Porqueras, I. Mateos, P. McNamara, J. Mendes, L. Mendes, M. Nofrarias, S. Paczkowski, M. Perreur-Lloyd, A. Petiteau, P. Pivato, E. Plagnol, J. Ramos-Castro, J. Reiche, D. Robertson, F. Rivas, G. Russano, J. Slutsky, C. Sopuerta, T. Sumner, D. Texier, J. Thorpe, D. Vetrugno, S. Vitale, G. Wanner, H. Ward, P. Wass, W. Weber, L. Wissel, A. Wittchen, and P. Zweifel, "Beyond the Required LISA Free-Fall Performance: New LISA Pathfinder Results down to 20 Hz," *Physical Review Letters*, vol. 120, Feb. 2018.

[11] R. C. Hilborn, "Gravitational waves without general relativity: A tutorial," *American Journal of Physics*, vol. 86, pp. 186–197, Mar. 2018. arXiv: 1710.04635.

[12] K. A. Arnaud, S. Babak, J. G. Baker, M. J. Benacquista, N. J. Cornish, C. Cutler, S. L. Larson, B. S. Sathyaprakash, M. Vallisneri, A. Vecchio, and J.-Y. Vinet, "A How-To for the Mock LISA Data Challenges," *AIP Conference Proceedings*, vol. 873, pp. 625–632, 2006. arXiv: gr-qc/0609106.

[13] "ESA creates quietest place in space," Feb. 2018. http://sci.esa.int/lisa-pathfinder/59961-esa-creates-quietest-place-in-space/.

[14] B. J. Owen and B. S. Sathyaprakash, "Matched filtering of gravitational waves from inspiraling compact binaries: Computational cost and template placement," *Physical Review D*, vol. 60, June 1999. arXiv: gr-qc/9808076.

[15] N. J. Cornish and J. Crowder, "LISA data analysis using Markov chain Monte Carlo methods," *Physical Review D*, vol. 72, Aug. 2005.

[16] T. B. Littenberg and N. J. Cornish, "Bayesian approach to the detection problem in gravitational wave astronomy," *Physical Review D*, vol. 80, Sept. 2009.

[17] J. Nicoules, "Characterizing gravitaional wave sources," tech. rep., CentraleSupelec, CEA Saclay IRFU/DPhN, Gif-sur-Yvette (France), Aug. 2018.