SciPost Physics Codebases Submission

emri_mc: A GPU-based Python code for Bayesian inference of EMRI waveforms

Ippocratis D. Saltas[1★] **and Roberto Oliveri**[2†]

1 CEICO, Institute of Physics, Czech Academy of Sciences
Na Slovance 2, 182 21 Praha 8, Czech Republic
2 LUTH, Laboratoire Univers et Théories, Observatoire de Paris
CNRS, Université PSL, Université Paris Cité,
5 place Jules Janssen, 92190 Meudon, France

★ saltas@fzu.cz , † roberto.oliveri@obspm.fr

Abstract

We describe a simple and efficient Python code to perform Bayesian forecasting for gravitational waves (GW) produced by Extreme-Mass-Ratio-Inspiral systems (EMRIs). The code runs on GPUs for an efficient parallelised computation of thousands of waveforms and sampling of the posterior through a Markov-Chain-Monte-Carlo (MCMC) algorithm. `emri_mc` generates EMRI waveforms based on the so–called kludge scheme, and propagates it to the observer accounting for cosmological effects in the observed waveform due to modified gravity/dark energy. The code provides a helpful resource for forecasts for interferometry missions in the milli-Hz scale, e.g the satellite-mission LISA.

Received Date
Accepted Date
Published Date

Contents

1 Introduction

Parameter forecasting for EMRI signals is not an easy task because of the complex nature of these signals and the high-dimensional parameter space that needs be explored. Most of the attempts in the literature to date are based on the kludge scheme for the waveform generation, as well as on the Fisher information matrix approach for the parameter forecast; see, e.g., [1–6]. An early attempt at parameter estimation using Bayesian inference was performed in [7] and, more recently, in [8, 9]

Exploring the high-dimensional parameter space with accuracy is hindered by the multi-modality of the likelihood, meaning that different sets of parameters can produce waveforms that look very similar to one another; see, e.g., [10]. The resulting parameter degeneracy creates local maxima in the likelihood function, making it difficult to locate the global maximum. Navigating this multi-modal landscape requires sophisticated sampling methods.

Yet another fundamental challenge in parameter estimation is the difficulty to distinguish between an EMRI signal and other non-negligible overlapping signals that originate from different astrophysical sources, such as galactic compact binaries. Addressing this challenge is crucial for LISA data analysis and usually requires fitting the global signal with efficient sampling algorithms [11–13].

Nowadays, modeling of EMRI waveform is achieving sufficient accuracy to detect and analyze EMRI signals in the data from future detectors like LISA. This involves refining self-force calculations [14] and improving computational methods for waveform generation [15].

In view of the burst of GW astronomy and the need for parameter estimation [16–19], including possible effects of modified gravity, `EMRI_MC` [1] provides a simple, yet efficient code for the GW community of astrophysics and cosmology towards parameter estimation and forecasts for the future LISA detector [17, 20–22].

`EMRI_MC` relies on four main elements: **i)** the waveform generator; **ii)** the inclusion of the amplitude damping and modified speed of GWs; **iii)** the posterior sampling through MCMC methods; **iv)** the GPU-based vectorisation of quantities such as the likelihood, in order to accelerate computations. Our code aims to provide a simple and efficient tool that could be of help for the community working on the interface of GWs and modified gravity. We emphasise that the structure of the code provides for enough flexibility to allow extension and improvements in each of its elements, according to the need of the specific task.

i) *The waveform generator*.

Our choice for the waveform generator relies on the popular *Analytic Kludge* (AK) model for the generation of inspiraling EMRI waveforms [1, 2]. The generation, as well as its Fast Fourier Transform (FTT), is implemented using appropriate GPU vectorisation techniques in cuda.

Though the AK model is not the most accurate waveform model to date, this choice is justified as follows. AK waveforms provide a sufficiently good approximation of the binary's dynamics, as long as one remains sufficiently far away from the merger. Additionally, AK waveforms allows for an analytic handle on the physics. The equations can be consistently extended with new post-Newtonian and self-force corrections, as well as the inclusion of new physics such as dark matter effects. In this regard, they provide an excellent proxy to perform parameter estimation for future missions, as well as investigate the significance of effects related to new physics.

The AK model can be replaced with a more accurate waveform model generator. Examples are the *Augmented Analytic Kludge* [4, 23], the *Numerical Kludge* [24–27], the *Fast EMRI Waveforms* [28, 29], and the *Effective One-Body* approach [30–33].

[1] **The code is available at:** https://doi.org/10.5281/zenodo.10204186.

ii) *The inclusion of modified gravity effects.*
We include effects beyond General Relativity during the propagation of GWs on the cosmological background. Specifically, we include the effects of damping of the amplitude and modification of the GW speed [34–36].

iii) *The posterior sampling.*
Most of the codes currently available for posterior samplings for EMRIs parameter space make use of the Fisher information matrix. We adopt a Bayesian approach using Markov-Chain-Monte-Carlo (MCMC) methods. These are implemented via the MCMC package `emcee` [37], which employs an affine-invariant ensemble sampler [38]. The posterior sampling is performed using GPU vectorisation: the likelihood function is computed for each MCMC walker in parallel.

iv) *The GPU-based vectorisation and parallelisation features.*
Bayesian inferences through MCMC methods are rather expensive for CPUs, especially when posteriors evaluations involve high-dimensional parameter space. To overcome this computational limitation, the code adopts GPU-based vectorisation and parallelisation features, notably for the generation of the waveform and for the posterior sampling..

2 Theoretical background

Waveform generation: The generation of accurate GW waveforms for binary systems and efficient parameter estimation is key for current and future GW missions such as LISA. For EMRIs, the accuracy of the waveform in the inspiral phase requires adiabatic, post-adiabatic and self-force approximations [14, 28, 29, 39–44].

The AK model we adopt in our code relies on the Peters-Mathews formalism [45,46], where adiabatic and post-Newtonian approximations are adopted. The main advantage thereof, for our purposes, is the analytic command over the waveform generation and its parameter space. AK model can also be easily extended and modified in the presence of new physics. For technical details on the theoretical framework and equations we will be using, we refer to [1] and references therein.

The system of equations consists of two main parts: **i)** the equations describing the orbital dynamics of the small body with mass μ around the central black hole with mass M and spin S/M^2, and **ii)** the equations for the generation of the waveform under the quadrupole approximation. The first ones form a system of ordinary differential equations (ODEs) as

$$\frac{d\mathrm{Y}}{dt} = f(\mathrm{Y}(t);\theta), \tag{1}$$

where the vector $\mathbf{Y}$ denotes the orbital parameters $\mathbf{Y} = \{\Phi, \nu, e, \gamma, \alpha\}$, i.e., the phase ($\Phi$), the orbital frequency (ν), the eccentricity (e) and two precession angles (γ, α). The vector θ denotes the free parameters in our waveform generation model $\theta = \{M, \mu, S, \ldots\}$, i.e., the masses, spin, angles, parameters due to propagation, etc. We note that we work in **cgs units**. For example, we restore powers of G and c, define the post-Newtonian order parameter $x = 2\pi\nu GM/c^3$, and the spin magnitude of the central black hole as $0 \leq S/M^2 \leq 1$.

The solution of the orbital equations under the quadrupole approximation allows for the computation of the waveform as

$$h_{ij}(t) = \sum_{n=1}^{n_{max}} h_{ij}^n(t) = \sum_{n=1}^{n_{max}} A_{(n)}^{+}(t,\theta)e_{ij}^{+}(t) + A_{(n)}^{\times}(t,\theta)e_{ij}^{\times}(t), \tag{2}$$

where it is understood that $A = A[\Upsilon(t), \theta]$. The polarisation coefficients are computed under a harmonic decomposition up to some overtone n_{max}, according to [45]. We notice that the detector response function, assumed to be included in the above expression, introduces three extra angles on top of the ones related to the orbital dynamics of the system. We assume an ideal detector and a perfect knowledge of the the detector response function. In other words, variations of the LISA detector response because of different noises and signals are neglected. The LISA response function used in the present work can be found, e.g., in [1].

Waveform propagation: Assuming a plane GW travelling far away from the source through the cosmological medium, we can write Eq. (2) as $h_{ij}(t) = h(t)e_{ij}$, and expand the amplitude of each mode of the wave in Fourier modes with spatial wavenumber k as

$$\ddot{h} + 3H(\tau)(2 + \alpha_{\text{M}})\dot{h} + k^2(1 + \alpha_{\text{T}})h = 0. \tag{3}$$

$H(\tau)$ is the Hubble parameter, τ the cosmological time, and the quantities $\alpha_{\text{M}}, \alpha_{\text{T}}$ parameterise effects beyond General Relativity modifying the friction and the wave's propagation speed respectively [34–36, 47]. In redshift domain, and under the WKB approximation, one can solve analytically Eq. (3) to find [35]

$$h(z) = h_{\text{MG}} \times h_{\text{GR}} \equiv \frac{1}{\Xi} \times e^{-ik\Delta T} \times h_{\text{GR}}, \tag{4}$$

with

$$\Xi(z) \equiv \frac{d^{\text{GW}}(z)}{d^{\text{EM}}(z)} \exp\left(\frac{1}{2}\int_0^z d\tilde{z}\,\frac{\alpha_{\text{M}}(\tilde{z})}{1+\tilde{z}}\right), \quad \Delta T \equiv \exp\left(-ik\int_0^z d\tilde{z}\,\frac{\alpha_{\text{T}}(\tilde{z})}{1+\tilde{z}}\right), \tag{5}$$

z the redshift to the source, and h_{GR} the contribution one gets from solving Eq. (3) for $\alpha_{\text{M}} = 0 = \alpha_{\text{T}}$. The possible cosmological evolution of $\alpha_{\text{M}}(z), \alpha_{\text{T}}(z)$ is model-dependent, however, they are in principle very slowly-varying functions of redshift, tracing the evolution of the dark energy density fraction. For a discussion on parametrisations of their time dependence we refer to [48]. For the sake of an example, we choose to parametrise $\Xi(z)$ through the physically well-motivated parametrisation of [49] (see also [50])

$$\Xi(z) = \Xi_0 + \frac{1 - \Xi_0}{(1+z)^n}, \tag{6}$$

with Ξ_0 a free parameter. Of course, any other physically-motivated parametrisation is equally good. There are scenarios where the parameters α_{M} and α_{T} are also frequency-dependent quantities (see e.g [36] for a detailed exploration) as

$$\alpha_{\text{M}} = F(z, f), \quad \alpha_{\text{T}} = G(z, f), \tag{7}$$

with F, G some well-motivated functions of GW frequency (f) and redshift (z). An example includes a power series expansion $\sum_n a_n(z)(f/f_*)^n$. Numerically, such frequency-dependent terms need to act upon a tabulated waveform in frequency space.

3 Numerical approach and statistical pipeline

Overview: Our goal is to streamline an efficient parameter estimation pipeline to get joint constraints on the free parameters of the model.

As a first step, we define a fiducial model with an associated set of fiducial parameters θ_0, which we use to generate the expected waveform $h[\theta_0](f)$ in Fourier space for this model.

This waveform is then used as the (mock) dataset at the heart of our MCMC analysis. At each step of our MCMC, the orbital equations are solved for the current set of parameters θ, and the time-domain waveform computed. The latter is then Fourier-transformed into the frequency domain, yielding the waveform $h[\theta](f)$. It is then compared to the mock dataset via the computation of the log-likelihood, as follows

$$\log \mathcal{L} \propto \frac{1}{2}\frac{(h[\theta](f) - h[\theta_0](f))^2}{S_{\rm noise}(f)}, \tag{8}$$

with $S_{\rm noise}(f)$ the LISA noise function. In the code, we have implemented two noise models; the one presented in [1] and the more recent LISA noise model of [51, 52]. In our example below we use the original noise model of [1].

We emphasize that, as already mentioned below Eq. (2), the waveform $h[\theta](f)$ is understood to be the waveform projected on LISA arms. In particular, the two polarization modes are projected to the LISA frame using the LISA antenna patterns (see e.g., [1]). The explicit expressions of the two polarizations of the waveform signals in the LISA frame can be read off the code in the **waveform.py** module. We further note that we assume *noise-less* waveforms, that is, our waveforms are perfect in their production and the noise enters through the so–called noise curve as a weight in the construction of the likelihood.

Waveform computation: The orbital equations (1) are solved as an initial-value problem on a time grid using standard ODE methods, such as the 7th-order Runge-Kutta scheme. Initial conditions are set at the Last Stable Orbit (LSO) and the equations are integrated backwards for a given time window, typically ranging between few months to one year. The resolution of the time grid is set at 0.1 Hz, which is the typical choice for LISA. As regards the initial value for eccentricity and angles of the system at LSO, we fix these geometrical quantities for both fiducial model and MCMC analysis and vary only masses, spin and other physical parameters. The initial value for the frequency ν at LSO is set according to [1], $\nu_{\rm LSO} = c^3/(2\pi GM)((1-e_{\rm LSO}^2)/(6+2e_{\rm LSO}))^{3/2}$. For the transition to the frequency domain, we use the method of a Fast Fourier Transform (FFT), under an appropriate normalisation choice.
The other parameters of the system are fixed to their true values. However, this assumption is not restrictive: the user may keep any of the parameter free to vary during the sampling.

Posterior sampling: The posterior sampling is performed through the Python MCMC package `emcee`, which allows for various sampling techniques and parallelisation features. Parallelisation is implemented in two ways, from which the user can choose. The first one uses the `multi-processing` library to parallelise the walkers. In this case, each walker runs independently of the others, and the input to the MCMC at each step is a 1D vector of dimension $N_{\rm dim}$ with the parameters of the step. The second parallelisation feature works differently. At each MCMC step, it creates a matrix of dimension $(n_{\rm walkers} \times n_{\rm dim})$, which is fed as the input into the MCMC engine. This parallelisation feature will tend to be more efficient with a large number of parameters.

Functionality overview: GPU vectorisation is achieved mainly through appropriate use of the `ElementwiseKernel` functionality which exploits the GPU parallelisation for mathematical operations, and is implemented through Python's `cupy` library. Computations such as the waveform and the likelihood are GPU-vectorised reducing significantly the evaluation time. For the MCMC exploration we have introduced two different parallelisation methods, for CPUs and GPUs respectively. The first one parallelises the walkers on different CPUs through the `multi-processing` framework. The second approach feeds into the MCMC algorithm a super-matrix of all parameters at given MCMC step, which is then computed in a vectorised manner. For **4** free parameters and 8 walkers, this brings down the evaluation of each MCMC step to about **2.2 seconds**, assuming a waveform integrated over 1 year at resolution of **0.1** Hz.

The code's architecture consists of the following **main files**:

1. global_parameters.py: This module defines the values of physical constants in cgs units, the parameters of the fiducial model, geometrical parameters and initial conditions of the binary system, parameters for the ODE solver (e.g., integration time window and grid resolution), and MCMC-related definitions. It also defines the maximum number of orbital overtones `n_max` in the computation of the waveform. A change in the number of the parameters in the MCMC requires adjusting the parameter vector in this module.

2. waveform.py: This module defines the set of kludge ODE equations (see Eq. (1)), the waveform generator according to Eq. (2), and some GPU-related functionality. Its main functions reads as follows:

– `eqs()`: Defines the set of kludge ODE equations and returns the right-hand-side of them in the sense of Eq. (1). Notice that in the case of the use of an ODE solver other than the native ones in Python, currently used, the return statement of this function might need to be changed.

– `compute_orbit()`: Computes the solutions of the kludge ODE equations defined in `eqs()`. To solve the system of ODEs, we use the Pythonic framework of `solve_ivp()`. This choice allowed to switch between the different native solvers in this library.

– `waveform()`: It calls `eqs()` and `compute_orbit()`, computes the time-domain waveform including the LISA response function (2) and then performs its FFT, via the function `FFT_gpu()`. The computation of waveforms is implemented fully on cuda. GPU vectorisation and acceleration is implemented with appropriate use of `ElementwiseKernel`. For computational convenience, the ouputted waveform **does not** include the overall factor of the GW luminosity distance. This is included in the function `iterate_mcmc()` below.

– `compute_fiducial()`: Computes the fiducial model based on the fiducual values defined in *global_parameters.py*. The parameter vector defined in this function needs adjustment when adding/removing parameters in the MCMC run.

3. mcmc.py: This module defines the MCMC-related functions and the MCMC iterator. Its main functions reads as follows:

– `lnprior()`, `lnprob()`: These functions define the log-prior and the log-probability, respectively. They need be adjusted when the set of parameters in the MCMC run is modified. Their counterparts for the case when the liklelihood is computed through a vectorization process (see module description "run_code.py" below) are labelled as `lnprior_vec()`, `lnprob_vec()`.

– `iterate_mcmc()`: It calls `waveform()` to compute the waveform, and the likelihood in frequency domain for a given choice of parameters around the fiducial model using GPU vectorisation.

– `get_noise()`: This defines the LISA noise function for GPU parallelisation through an `ElementwiseKernel`.

– `get_Likelihood()`: This function computes the likelihood according to Eq. (8) in GPU vectorised form through an `ElementwiseKernel`. It is used in `iterate_mcmc()` to compute the likelihood at each MCMC step. Modifications to the GW luminosity distance enter here. This function needs to be adjusted according to any change of parameters in the MCMC run.

4. propagation.py: This module defines the functions needed for the propagation of the GW wave through the cosmological background in the presence of any modified gravity effects.

– `get_damping()`, `get_modified_speed()`: This defines the possible frequency-dependent damping of the waveform's amplitude, or the respective change in its propagation speed, due to modified gravity. It is defined through an `ElementwiseKernel` for an efficient evaluation on the frequency grid. After defining the functional form of the frequency-dependent damping and/or GW speed, one should modify appropriately the computation of the likelihood in the function `iterate_mcmc()`. Detailed comments are provided in the code.

– `dL()`: This function defines the electromagnetic luminosity.

– `dGW_Xi()`: This function defines a redshift-dependent parametrisation of the GW luminosity distance due to modified gravity according to Eq. (6). It should appear as an overall multiplicative factor of the waveform in the likelihood computation in `iterate_mcmc()`. We remind that the function `waveform()` does not include in the outputted waveform the luminosity distance factor.

5. run_code.py: This module starts the MCMC run, building on the parameter definitions in global_parameters.py. If `vectorize = True`, the code inputs the data of the parameter configuration at a given step into the MCMC engine as a matrix $(n_{\rm walkers} \times n_{\rm dim})$, and parallelises the computation on GPU. If `vectorize = False`, the code parallelises instead the walkers on CPUs through the `multi-processing` framework.

6. main.ipynb: Assuming all parameters and fiducial model are properly defined as explained earlier, this Jupyter notebook serves as an example demonstration of the code. It essentially calls the main functions to initiate the MCMC run, using the package `emcee`. As a simple choice, we have currently set throughout the numerical computation the source location $\{\theta_S, \phi_S\} = \{\pi/4, 0\}$, the orientation of the spin $\{\theta_K, \phi_K\} = \{\pi/8, 0\}$, $\alpha_{\rm LSO} = 0$, the angle $\lambda = \pi/6$, the initial eccentricity $e_{\rm LSO} = 0.3$, and $\gamma_{\rm LSO} = 0$, $\Phi_{\rm LSO} = 0$, for the respective initial conditions. These can be straighforwardly modified in the file *waveform.py*.

Extending the parameters in the MCMC run: First we notice that, in the vector p defining the parameters to be varied in the MCMC, the first three values should be by default the central mass (M), the orbiting mass (μ), and the spin (S/M^2). These are needed by the ODE solver to solve the equations and are passed as `args` = $[p[0], p[1], p[2]]$. Therefore, it is advisable to always keep this convention. Now, to add new parameters in the MCMC run one needs to make changes at the following points in the code:

– *global_parameters.py*: Edit the sections on parameters for the MCMC run and parameter values for the fiducial model.

– *mcmc.py*: Edit the vector `p` in the functions `lnprior()` and `lnprob()`. Edit the parameters to be varied in the MCMC in the function `iterate_mcmc()`, including possible modifications in the GW luminosity distance which enters in the computation of the likelihood in `iterate_mcmc()`.

– *main.ipynb*: Extend the vector `p_init_MC` which initialises the walkers with the new parameters. Ensure that values for the initialisation of the walkers is meaningful given the problem at hand, otherwise the MCMC will not converge as expected.

– *propagation.py*: Any new parameters which affect the GW luminosity distance must be also reflected in this module where the definitions of the luminosity distances are placed.

Computational overhead, ODE solver, overtones summation: An important computational overhead in the evaluation of each MCMC step comes from the choice of the ODE solver. We currently use the native ODE solvers provided by Python's numerical libraries. However, this can be improved using different, external solvers, such as the ones provided by the *Fast EMRI Waveforms* scheme [28,29]. We notice that the choice of the ODE solver enters in the functions `waveform()` and `compute_fiducial()`. We should also notice that, the choice of different solvers (e.g. RK23 or LSODA) leads to different computation times. Contrary to the ODE solver, the for-loop which sums over overtones in the function `waveform()`, does not seem to cause any sizable computational overhead for reasonable choices of the maximum overtone (n_{max}), we therefore decided not to implement any vectorisation on it, but we plan to explore this feature further in the future.

4 Installing and running the code

Installation of the code essentially requires the installation of the supporting packages, explained in the README file of the code. Running the code is particularly simple. Placing all files in the same folder, and setting up all parameters as explained above, one starts the notebook *main.ipynb*, and executes the cells. The first cell computes the fiducial model, and the following cells start the MCMC run around the chosen fiducial. The MCMC results are stored in a `.txt` file. Currently, for the sake of an example we consider a 4-parameter case: 3 source parameters (2 masses and 1 spin), and 1 propagation parameter (Ξ_0).

As an illustration, in Figure 1 we plot the computed waveforms for characteristic values of the eccentricity and spin, as computed by the function `plot_waveform()`. The orbital angles have been fixed according to the conventions mentioned earlier. What is more, Figure 2 shows an example corner plot from a MCMC run with 4 free parameters - 3 for the generation (masses+spin) and 1 for the propagation of the waveform respectively (Ξ_0). As it can be seen, our constraints on the parameter Ξ_0 (see equation (6)), which relates to a modified gravity effect in the propagation of the waveform, are within the same order of magnitude as with very recent results in the literature [9]. It is also interesting to compare our results with those of [6]. Despite the similarity, the differences in the numbers can be due to a multitude of factors, for example, the fact that our MCMC exploration covers a smaller EMRIs' parameter space, the different choice of the noise function, the use of noise-less waveforms, or even the different choice of overtones. We remind that our particular example considers the orientation of the binary parameters fixed, but these can be allowed to vary in the code.

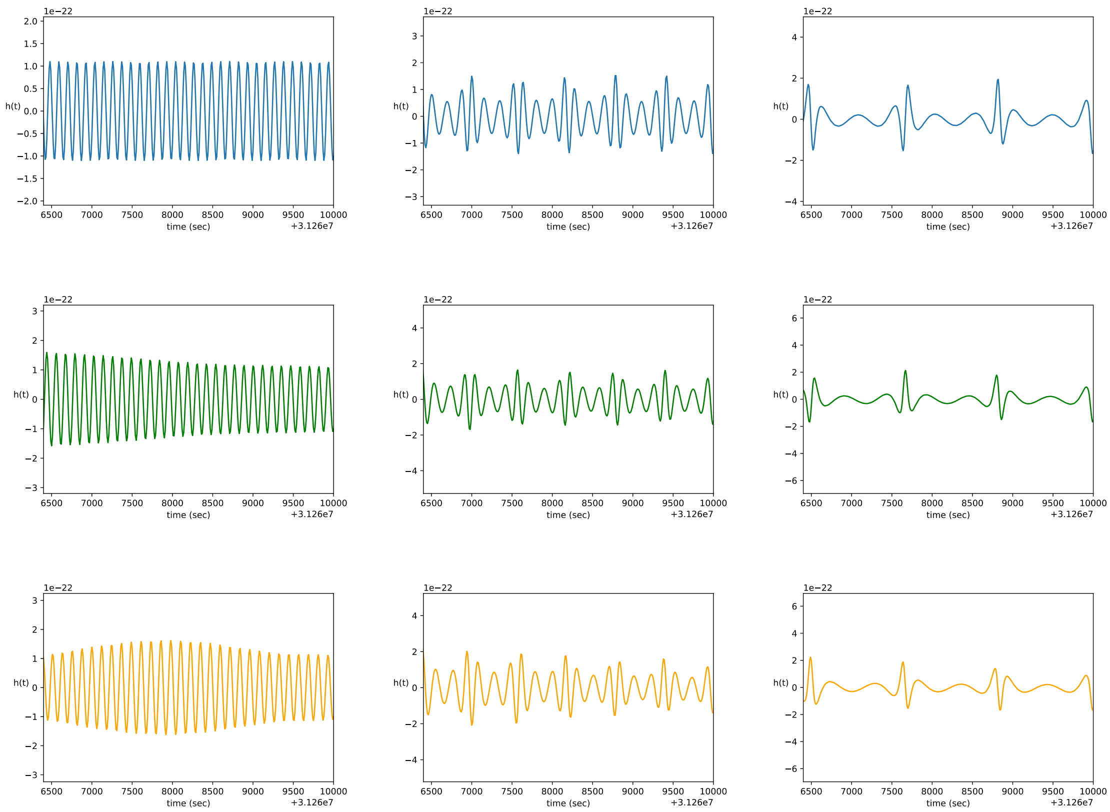

Figure 1: The kludge waveform computed for the last hour before the plunge at the Last Stable Orbit (LSO). Parameters: $M = 10^6 M_\odot$ (central mass), $\mu = 10 M_\odot$ (orbiting mass), $(\theta_S, \phi_S) = (\pi/4, 0)$, $(\theta_K, \phi_K) = (\pi/8, 0)$, $\lambda = \pi/6$ (frame angles), $D = 1$ Gpc (distance to the source). First row: $S/M^2 = 0$ (dimensionless spin of the central black hole), $e_{\mathrm{LSO}} = 0, 0.3, 0.6$ (eccentricity). Second row: $S/M^2 = 0.4$, $e_{\mathrm{LSO}} = 0, 0.3, 0.6$. Third row: $S/M^2 = 0.8$, $e_{\mathrm{LSO}} = 0, 0.3, 0.6$.

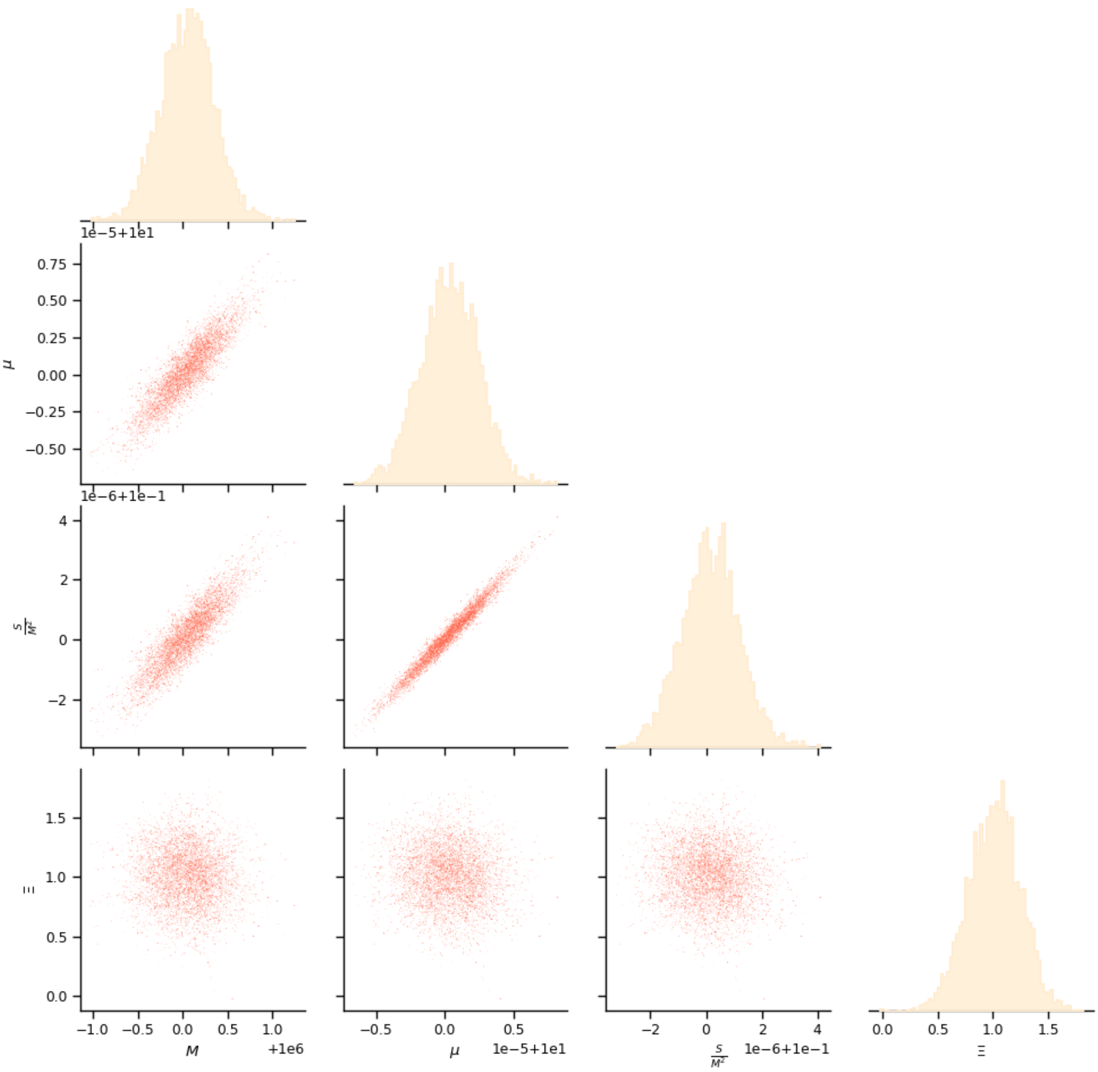

Figure 2: An example corner plot from an MCMC exploration with fiducial/injected values $M = 10^6 M_\odot$ (central mass), $\mu = 10 M_\odot$ (orbiting mass), $S/M^2 = 0.1$ (dimensionless spin of the central black hole), $\Xi_0 = 1$ (no modified GR effects; see Eq. (6)), 2000 steps and 8 walkers. We have assumed an observation of one year. Median and 90% C.I. are $M/(10^6 M_\odot) = 1.28^{+0.148}_{-0.2604}$, $\mu/M_\odot = 10.0000021^{+0.0000010}_{-0.0000019}$, $S/M^2 = 0.1000010^{+0.0000005}_{-0.0000009}$, and $\Xi = 1.1685571^{+0.1043108}_{-0.1965527}$. We have assumed that the distance (redshift) to the source is known, and equal to 1 Gpc. We note that the constraints are somewhat tighter than those in the literature [6]. Some factors contributing to this difference is that our MCMC exploration covers a smaller EMRIs' parameter space, the different choice of the noise function, and the use of noise-less waveforms. The eccentricity and the orbital angles at LSO have been kept fixed in the MCMC run. We have used the LISA noise model of [1].

5 Future directions

Surely the current implementation of this code can be expanded in different interesting and more accurate ways, for example: **i)** The inclusion of environmental effects in the production of the waveform, such as dark matter or baryonic effects due to accretion of the central black hole. Such effects would introduce amongst other things, a new dissipating channel due to the force of the dynamical friction encountered by the orbiting mass. **ii)** Our current use of the standard kludge equations has been based on a trade-off between simplicity and accuracy, combined with the popularity of this formalism for parameter estimation in the literature. However, its waveforms are known to suffer from certain inaccuracies. Improvement can be achieved by implementing the so–called Augmented Kludge Formalism, or the waveforms of *Fast EMRI Waveforms* discussed earlier, which would need a more involved implementation. **iii)** Moreover, the implementation of a more efficient ODE solver for the orbital equations could allow us to achieve even faster iterations in the MCMC sampling run. **iv)** Finally, it is noteworthy to consider implementing a Bayesian approach by means of deep learning techniques tailored to explore the EMRI parameter space, as recently proposed in [53].

Acknowledgements

We are indebted to Stéphane Ilić for collaboration on an early stage of the project, for his critical advice on MCMC simulations, and for his comments on the final version of the draft and code. We further acknowledge the contribution and advice of Josef Dvořáček on GPU optimisation, and other code-development aspects throughout the duration of this project, which have improved the efficiency of the code. We acknowledge the use of the "Phoebe" computer cluster at CEICO/FZU where the code was tested. We also thank Leor Barack, George Loukes-Gerakopoulos, Simone Mastrogiovanni, Adam Pound and Nick Stergioulas for discussions. We would also like to thank the Referee of our work, whose comments has improved our code and manuscript.

Funding information I.D.S. acknowledges funding by the Czech Grant Agency (GAĈR) under the grant number 21-16583M. The work of R.O. is supported by the Région Île-de-France within the DIM ACAV+ project SYMONGRAV (Symétries asymptotiques et ondes gravitationnelles).

References

[1] L. Barack and C. Cutler, *LISA capture sources: Approximate waveforms, signal-to-noise ratios, and parameter estimation accuracy*, Phys. Rev. D **69**, 082005 (2004), doi:10.1103/PhysRevD.69.082005, gr-qc/0310125.

[2] L. Barack and C. Cutler, *Using LISA EMRI sources to test off-Kerr deviations in the geometry of massive black holes*, Phys. Rev. D **75**, 042003 (2007), doi:10.1103/PhysRevD.75.042003, gr-qc/0612029.

[3] M. Vallisneri, *Use and abuse of the Fisher information matrix in the assessment of gravitational-wave parameter-estimation prospects*, Phys. Rev. D **77**, 042001 (2008), doi:10.1103/PhysRevD.77.042001, gr-qc/0703086.

[4] A. J. K. Chua, C. J. Moore and J. R. Gair, *Augmented kludge waveforms for detecting extreme-mass-ratio inspirals*, Phys. Rev. D **96**(4), 044005 (2017), doi:10.1103/PhysRevD.96.044005, 1705.04259.

[5] C. J. Moore, A. J. K. Chua and J. R. Gair, *Gravitational waves from extreme mass ratio inspirals around bumpy black holes*, Class. Quant. Grav. **34**(19), 195009 (2017), doi:10.1088/1361-6382/aa85fa, 1707.00712.

[6] S. Babak, J. Gair, A. Sesana, E. Barausse, C. F. Sopuerta, C. P. L. Berry, E. Berti, P. Amaro-Seoane, A. Petiteau and A. Klein, *Science with the space-based interferometer LISA. V: Extreme mass-ratio inspirals*, Phys. Rev. D **95**(10), 103012 (2017), doi:10.1103/PhysRevD.95.103012, 1703.09722.

[7] A. Ali, N. Christensen, R. Meyer and C. Rover, *Bayesian inference on EMRI signals using low frequency approximations*, Class. Quant. Grav. **29**, 145014 (2012), doi:10.1088/0264-9381/29/14/145014, 1301.0455.

[8] O. Burke, G. A. Piovano, N. Warburton, P. Lynch, L. Speri, C. Kavanagh, B. Wardell, A. Pound, L. Durkan and J. Miller, *Accuracy Requirements: Assessing the Importance of First Post-Adiabatic Terms for Small-Mass-Ratio Binaries* (2023), 2310.08927.

[9] C. Liu, D. Laghi and N. Tamanini, *Probing modified gravitational-wave propagation with extreme mass-ratio inspirals*, Phys. Rev. D **109**(6), 063521 (2024), doi:10.1103/PhysRevD.109.063521, 2310.12813.

[10] A. J. K. Chua and C. J. Cutler, *Nonlocal parameter degeneracy in the intrinsic space of gravitational-wave signals from extreme-mass-ratio inspirals*, Phys. Rev. D **106**(12), 124046 (2022), doi:10.1103/PhysRevD.106.124046, 2109.14254.

[11] N. J. Cornish and J. Crowder, *LISA data analysis using MCMC methods*, Phys. Rev. D **72**, 043005 (2005), doi:10.1103/PhysRevD.72.043005, gr-qc/0506059.

[12] T. B. Littenberg and N. J. Cornish, *Prototype global analysis of LISA data with multiple source types*, Phys. Rev. D **107**(6), 063004 (2023), doi:10.1103/PhysRevD.107.063004, 2301.03673.

[13] M. L. Katz, N. Karnesis, N. Korsakova, J. R. Gair and N. Stergioulas, *An efficient GPU-accelerated multi-source global fit pipeline for LISA data analysis* (2024), 2405.04690.

[14] B. Wardell, A. Pound, N. Warburton, J. Miller, L. Durkan and A. Le Tiec, *Gravitational Waveforms for Compact Binaries from Second-Order Self-Force Theory*, Phys. Rev. Lett. **130**(24), 241402 (2023), doi:10.1103/PhysRevLett.130.241402, 2112.12265.

[15] N. Afshordi *et al.*, *Waveform Modelling for the Laser Interferometer Space Antenna* (2023), 2311.01300.

[16] N. Christensen and R. Meyer, *Parameter estimation with gravitational waves*, Rev. Mod. Phys. **94**(2), 025001 (2022), doi:10.1103/RevModPhys.94.025001, 2204.04449.

[17] F. Iacovelli, M. Mancarella, S. Foffa and M. Maggiore, *GWFAST: A Fisher Information Matrix Python Code for Third-generation Gravitational-wave Detectors*, Astrophys. J. Supp. **263**(1), 2 (2022), doi:10.3847/1538-4365/ac9129, 2207.06910.

[18] F. Iacovelli, M. Mancarella, S. Foffa and M. Maggiore, *Forecasting the Detection Capabilities of Third-generation Gravitational-wave Detectors Using GWFAST*, Astrophys. J. **941**(2), 208 (2022), doi:10.3847/1538-4357/ac9cd4, 2207.02771.

[19] S. Mastrogiovanni, K. Leyde, C. Karathanasis, E. Chassande-Mottin, D. A. Steer, J. Gair, A. Ghosh, R. Gray, S. Mukherjee and S. Rinaldi, *On the importance of source population models for gravitational-wave cosmology*, Phys. Rev. D **104**(6), 062009 (2021), doi:10.1103/PhysRevD.104.062009, 2103.14663.

[20] P. A. Seoane *et al.*, *Astrophysics with the Laser Interferometer Space Antenna*, Living Rev. Rel. **26**(1), 2 (2023), doi:10.1007/s41114-022-00041-y, 2203.06016.

[21] K. G. Arun *et al.*, *New horizons for fundamental physics with LISA*, Living Rev. Rel. **25**(1), 4 (2022), doi:10.1007/s41114-022-00036-9, 2205.01597.

[22] P. Auclair *et al.*, *Cosmology with the Laser Interferometer Space Antenna*, Living Rev. Rel. **26**(1), 5 (2023), doi:10.1007/s41114-023-00045-2, 2204.05434.

[23] A. J. K. Chua and J. R. Gair, *Improved analytic extreme-mass-ratio inspiral model for scoping out eLISA data analysis*, Class. Quant. Grav. **32**, 232002 (2015), doi:10.1088/0264-9381/32/23/232002, 1510.06245.

[24] K. Glampedakis, S. A. Hughes and D. Kennefick, *Approximating the inspiral of test bodies into kerr black holes*, Phys. Rev. D **66**, 064005 (2002), doi:10.1103/PhysRevD.66.064005.

[25] K. Glampedakis and D. Kennefick, *Zoom and whirl: Eccentric equatorial orbits around spinning black holes and their evolution under gravitational radiation reaction*, Phys. Rev. D **66**, 044002 (2002), doi:10.1103/PhysRevD.66.044002.

[26] J. R. Gair and K. Glampedakis, *Improved approximate inspirals of test bodies into kerr black holes*, Phys. Rev. D **73**, 064037 (2006), doi:10.1103/PhysRevD.73.064037.

[27] S. Babak, H. Fang, J. R. Gair, K. Glampedakis and S. A. Hughes, *"kludge" gravitational waveforms for a test-body orbiting a kerr black hole*, Phys. Rev. D **75**, 024005 (2007), doi:10.1103/PhysRevD.75.024005.

[28] A. J. K. Chua, M. L. Katz, N. Warburton and S. A. Hughes, *Rapid generation of fully relativistic extreme-mass-ratio-inspiral waveform templates for LISA data analysis*, Phys. Rev. Lett. **126**(5), 051102 (2021), doi:10.1103/PhysRevLett.126.051102, 2008.06071.

[29] M. L. Katz, A. J. K. Chua, L. Speri, N. Warburton and S. A. Hughes, *Fast extreme-mass-ratio-inspiral waveforms: New tools for millihertz gravitational-wave data analysis*, Phys. Rev. D **104**(6), 064047 (2021), doi:10.1103/PhysRevD.104.064047, 2104.04582.

[30] N. Yunes, A. Buonanno, S. A. Hughes, M. C. Miller and Y. Pan, *Modeling extreme mass ratio inspirals within the effective-one-body approach*, Phys. Rev. Lett. **104**, 091102 (2010), doi:10.1103/PhysRevLett.104.091102.

[31] N. Yunes, A. Buonanno, S. A. Hughes, Y. Pan, E. Barausse, M. C. Miller and W. Throwe, *Extreme mass-ratio inspirals in the effective-one-body approach: Quasicircular, equatorial orbits around a spinning black hole*, Phys. Rev. D **83**, 044044 (2011), doi:10.1103/PhysRevD.83.044044.

[32] S. Albanesi, A. Nagar and S. Bernuzzi, *Effective one-body model for extreme-mass-ratio spinning binaries on eccentric equatorial orbits: Testing radiation reaction and waveform*, Phys. Rev. D **104**, 024067 (2021), doi:10.1103/PhysRevD.104.024067.

[33] A. Albertini, R. Gamba, A. Nagar and S. Bernuzzi, *Effective-one-body waveforms for extreme-mass-ratio binaries: Consistency with second-order gravitational self-force quasi-circular results and extension to nonprecessing spins and eccentricity*, Phys. Rev. D **109**(4), 044022 (2024), doi:10.1103/PhysRevD.109.044022, 2310.13578.

[34] I. D. Saltas, I. Sawicki, L. Amendola and M. Kunz, *Anisotropic Stress as a Signature of Nonstandard Propagation of Gravitational Waves*, Phys. Rev. Lett. **113**(19), 191101 (2014), doi:10.1103/PhysRevLett.113.191101, 1406.7139.

[35] A. Nishizawa, *Generalized framework for testing gravity with gravitational-wave propagation. I. Formulation*, Phys. Rev. D **97**(10), 104037 (2018), doi:10.1103/PhysRevD.97.104037, 1710.04825.

[36] T. Baker *et al.*, *Measuring the propagation speed of gravitational waves with LISA*, JCAP **08**(08), 031 (2022), doi:10.1088/1475-7516/2022/08/031, 2203.00566.

[37] D. Foreman-Mackey, D. W. Hogg, D. Lang and J. Goodman, *emcee: The MCMC hammer*, Publications of the Astronomical Society of the Pacific **125**(925), 306 (2013), doi:10.1086/670067.

[38] J. Goodman and J. Weare, *Ensemble samplers with affine invariance*, Communications in Applied Mathematics and Computational Science **5**(1), 65 (2010), doi:10.2140/camcos.2010.5.65.

[39] E. Poisson, A. Pound and I. Vega, *The Motion of point particles in curved spacetime*, Living Rev. Rel. **14**, 7 (2011), doi:10.12942/lrr-2011-7, 1102.0529.

[40] M. van de Meent, *Gravitational self-force on generic bound geodesics in Kerr spacetime*, Phys. Rev. D **97**(10), 104033 (2018), doi:10.1103/PhysRevD.97.104033, 1711.09607.

[41] L. Barack and A. Pound, *Self-force and radiation reaction in general relativity*, Rept. Prog. Phys. **82**(1), 016904 (2019), doi:10.1088/1361-6633/aae552, 1805.10385.

[42] A. Pound and B. Wardell, *Black hole perturbation theory and gravitational self-force* (2021), doi:10.1007/978-981-15-4702-7_38-1, 2101.04592.

[43] S. A. Hughes, N. Warburton, G. Khanna, A. J. K. Chua and M. L. Katz, *Adiabatic waveforms for extreme mass-ratio inspirals via multivoice decomposition in time and frequency*, Phys. Rev. D **103**(10), 104014 (2021), doi:10.1103/PhysRevD.103.104014, 2102.02713.

[44] N. Warburton, A. Pound, B. Wardell, J. Miller and L. Durkan, *Gravitational-Wave Energy Flux for Compact Binaries through Second Order in the Mass Ratio*, Phys. Rev. Lett. **127**(15), 151102 (2021), doi:10.1103/PhysRevLett.127.151102, 2107.01298.

[45] P. C. Peters and J. Mathews, *Gravitational radiation from point masses in a Keplerian orbit*, Phys. Rev. **131**, 435 (1963), doi:10.1103/PhysRev.131.435.

[46] P. C. Peters, *Gravitational Radiation and the Motion of Two Point Masses*, Phys. Rev. **136**, B1224 (1964), doi:10.1103/PhysRev.136.B1224.

[47] I. D. Saltas, L. Amendola, M. Kunz and I. Sawicki, *Modified gravity, gravitational waves and the large-scale structure of the Universe: A brief report*, In *15th Marcel Grossmann Meeting on Recent Developments in Theoretical and Experimental General Relativity, Astrophysics, and Relativistic Field Theories* (2018), 1812.03969.

[48] E. Bellini and I. Sawicki, *Maximal freedom at minimum cost: linear large-scale structure in general modifications of gravity*, JCAP **07**, 050 (2014), doi:10.1088/1475-7516/2014/07/050, 1404.3713.

[49] E. Belgacem, Y. Dirian, S. Foffa and M. Maggiore, *Modified gravitational-wave propagation and standard sirens*, Phys. Rev. D **98**, 023510 (2018), doi:10.1103/PhysRevD.98.023510.

[50] I. S. Matos, E. Bellini, M. O. Calvão and M. Kunz, *Testing gravity with gravitational wave friction and gravitational slip*, JCAP **05**, 030 (2023), doi:10.1088/1475-7516/2023/05/030, 2210.12174.

[51] N. Cornish and T. Robson, *Galactic binary science with the new LISA design*, J. Phys. Conf. Ser. **840**(1), 012024 (2017), doi:10.1088/1742-6596/840/1/012024, 1703.09858.

[52] T. Robson, N. J. Cornish and C. Liu, *The construction and use of LISA sensitivity curves*, Class. Quant. Grav. **36**(10), 105011 (2019), doi:10.1088/1361-6382/ab1101, 1803.01944.

[53] B. Liang, H. Guo, T. Zhao, H. wang, H. Evangelinelis, Y. Xu, C. liu, M. Liang, X. Wei, Y. Yuan, P. Xu, M. Du *et al.*, *Rapid parameter estimation for extreme mass ratio inspirals using machine learning* (2024), 2409.07957.