Bayesian inference for compact binary sources of gravitational waves

Yann Bouffanais

Thèse préparée
à l'Université Paris Diderot
Ecole doctorale STEP'UP - ED 560
Laboratoire Astroparticule et Cosmologie / Equipe de recherche
Gravitation

Bayesian inference for compact binary sources of gravitational waves

Par Yann Bouffanais

Thèse de Physique de l'Univers

dirigée par Edward K. Porter

Présentée et soutenue publiquement à Paris le
11 Octobre 2017

devant un jury composé de :

Danièle Steer	Professeur, Université Paris Diderot	*Présidente du jury*
John Veitch	Senior lecturer, University of Glasgow	*Rapporteur*
Jonathan Gair	Senior lecturer, University of Edinburgh	*Rapporteur*
Ken Ganga	Directeur de recherche, Université Paris Diderot	*Examinateur*
Nelson Christensen	Directeur de recherche, Université Nice Sophia Antipolis	*Examinateur*
Edward K. Porter	Chargé de recherche, Université Paris Diderot	*Directeur de thèse*

Titre: Inférence Bayésienne pour les sources compactes binaries dondes gravitationnelles

Résumé: La première détection des ondes gravitationnelles en 2015 a ouvert un nouveau plan d'étude pour l'astrophysique des étoiles binaires compactes. En utilisant les données des détections faites par les détecteurs terrestres advanced LIGO et advanced Virgo, il est possible de contraindre les paramètres physiques de ces systèmes avec une analyse Bayésienne et ainsi approfondir notre connaissance physique des étoiles binaires compactes. Cependant, pour pouvoir être en mesure d'obtenir de tels résultats, il est essentiel davoir des algorithmes performants à la fois pour trouver les signaux de ces ondes gravitationnelles et pour l'estimation de paramètres. Le travail de cette thèse a ainsi été centré autour du développement dalgorithmes performants et adaptées au problème physique à la fois de la détection et de l'estimation des paramètres pour les ondes gravitationnelles. La plus grande partie de ce travail de thèse a ainsi été dédiée à l'implémentation dun algorithme de type Hamiltonian Monte Carlo adapté à l'estimation de paramètres pour les signaux dondes gravitationnelles émises par des binaires compactes formées de deux étoiles à neutrons. L'algorithme développé a été testé sur une sélection de sources et a été capable de fournir de meilleures performances que d'autres algorithmes de type MCMC comme l'algorithme de Metropolis-Hasting et l'algorithme à évolution différentielle. L'implémentation d'un tel algorithme dans les pipelines danalyse de données de la collaboration pourrait augmenter grandement l'efficacité de l'estimation de paramètres. De plus, il permettrait également de réduire drastiquement le temps de calcul nécessaire, ce qui est un facteur essentiel pour le futur o de nombreuses détections sont attendues. Un autre aspect de ce travail de thèse a été dédié à l'implémentation d'un algorithme de recherche de signaux gravitationnelles pour les binaires compactes monochromatiques qui seront observées par la future mission spatiale LISA. L'algorithme est une mixture de plusieurs algorithmes évolutionnistes, avec notamment l'inclusion d'un algorithme de Particle Swarm Optimisation. Cette algorithme a été testé dans plusieurs cas tests et a été capable de trouver toutes les sources gravitationnelles comprises dans un signal donné. De plus, l'algorithme a également été capable d'identifier des sources sur une bande de fréquence aussi grande que 1 mHz, ce qui n'avait pas été réalisé au moment de cette étude de thèse.

Mots clefs: ondes gravitationnelles, binaries compactes, étoiles à neutrons, Hamiltonian Monte Carlo, Particle Swarm Optimisation, analyse Bayésienne

Title: Bayesian inference for compact binary sources of gravitational waves

Abstract: The first detection of gravitational waves in 2015 has opened a new window for the study of the astrophysics of compact binaries. Thanks to the data taken by the ground-based detectors advanced LIGO and advanced Virgo, it is now possible to constrain the physical parameters of compact binaries using a full Bayesian analysis in order to increase our physical knowledge on compact binaries. However, in order to be able to perform such analysis, it is essential to have efficient algorithms both to search for the signals and for parameter estimation. The main part of this thesis has been dedicated to the implementation of a Hamiltonian Monte Carlo algorithm suited for the parameter estimation of gravitational waves emitted by compact binaries composed of neutron stars. The algorithm has been tested on a selection of sources and has been able to produce better performances than other types of MCMC methods such as Metropolis-Hastings and Differential Evolution Monte Carlo. The implementation of the HMC algorithm in the data analysis pipelines of the Ligo/Virgo collaboration could greatly increase the efficiency of parameter estimation. In addition, it could also drastically reduce the computation time associated to the parameter estimation of such sources of gravitational waves, which will be of particular interest in the near future when there will many detections by the ground-based network of gravitational wave detectors. Another aspect of this work was dedicated to the implementation of a search algorithm for gravitational wave signals emitted by monochromatic compact binaries as observed by the space-based detector LISA. The developed algorithm is a mixture of several evolutionary algorithms, including Particle Swarm Optimisation. This algorithm has been tested on several test cases and has been able to find all the sources buried in a signal. Furthermore, the algorithm has been able to find the sources on a band of frequency as large as 1 mHz which wasnt done at the time of this thesis study.

Keywords: gravitational waves, compact binaries, neutron stars, Hamiltonian Monte Carlo, Particle Swarm Optimisation, Bayesian analysis

Contents

Résumé

En septembre 2015, les détecteurs terrestres advanced LIGO et advanced Virgo ont fait la première détection directe d'ondes gravitationnelles émises lors de la coalescence de deux trous noirs de masses stellaires. Cette decouverte majeure a signé le début de l'astronomie gravitationnelle qui consiste à étudier les phénomènes astrophysques et cosmologiques de notre Univers en utilisant l'information contenue dans les ondes gravitationnelles. Pour pouvoir extraire l'information physique du signal temporel mesuré par les détecteurs d'ondes gravitationnelles, il est nécessaire d'utliser des techniques avancées d'analyse de données, adaptées à la fois à la recherche de signaux et à l'estimation de paramètres physiques. Le travail de cette thèse a été centré sur le développement de techniques dédiées à l'analyse de données pour les ondes gravitationnelles en utilisant le cadre de l'analyse Bayésienne. Chronologiquement parlant, le premier projet mis en oeuvre a été consacré à la mise en place d'un algorithme de type évolutionnaire pour la recherche des compactes binaires galactiques avec le détecteur LISA. Le deuxième et principal projet de cette thèse était lui consacré à la mise en oeuvre d'un algorithme d'estimation de paramètres pour les binaires d'étoiles à neutrons mesurées par les détecteurs advanced Virgo et advanced LIGO. Nous résumons dans cette partie le principal contenu scientifique détaillé dans ce manuscript ainsi que les différents travaux réalisés durant ce travail de thèse.

Dans le cadre de la relativité générale, l'espace et le temps sont considérés dans un seul ensemble nommé espace-temps. La dynamique de l'espace-temps est dicté par les équations d'Einstein qui mettent en relation la géométrie de l'espace-temps avec son contenu en matière et énergie de la manière suivante,

$$G_{\mu\nu} = \frac{8\pi G}{c^4} T_{\mu\nu}, \tag{1}$$

où $G_{\mu\nu}$ est le tenseur d'Einstein, $T_{\mu\nu}$ le tenseur énergie-moment, c la vitesse de la lumière et G la constante de gravitation universelle. Les ondes gravitationnelles apparaissent comme une conséquence de la relativité générale lorsque l'on écrit les équations d'Einstein précédentes de manière linéarisées. Dans ce cas, on consid'ère une métrique d'espace-temps $g_{\mu\nu}$ définie de la manière suivante,

$$g_{\mu\nu} = \eta_{\mu\nu} + h_{\mu\nu} + \mathcal{O}(h^2), \tag{2}$$

où $\eta_{\mu\nu}$ est la métrique de Minkowski et $h_{\mu\nu}$ est une perturbation se propageant sur $\eta_{\mu\nu}$ et de faible amplitude tel que $\mid h_{\mu\nu} \mid << 1$. En réécrivant le problème en utilisant un choix de gauge adaptée et dénommée la gauge transverse sans trace, les équations linéarisées d'Einstein s'écrivent alors sous la forme,

$$\Box \overline{h}_{\mu\nu} = -\frac{16\pi G}{c^4} T_{\mu\nu}, \tag{3}$$

où $\Box = \partial_\mu \partial^\mu$ est l'opérateur d'Alembertien sur l'espace-temps plat, $\overline{h}_{\mu\nu} = h_{\mu\nu} - \frac{1}{2}\eta_{\mu\nu} h$ et h est la trace de $h_{\mu\nu}$. Ces équations sont les équations fondamentales de la dynamique des ondes gravitationnelles que ce soit en termes de leur propagation, de leur création ou de leur interaction avec la matière.

Si l'on considère maintenant une onde gravitationnelle se propageant dans un espace-temps vide de matière et d'énergie, le terme à droite dans l'Eq. (3) devient égale à 0. Dans ce contexte, il est possible de montrer que la solution des équations d'Einsten linéarisées s'écrit sour la forme d'ondes planes se propageant à la vitesse de la lumière c. De plus, la théorie montre que ces ondes gravitationnelles sont transverses à la direction de propagation et s'expriment seulement en termes de deux polarisations indépendantes.

Une source située à une distance R d'un observateur génère des ondes gravitationnelles, $\overline{h}_{ij}(t, \mathbf{x})$, selon la formule du quadrupole,

$$\overline{h}_{ij}(t, \mathbf{x}) = \frac{2G}{3Rc^4} \frac{\mathrm{d}^2 Q_{ij}}{\mathrm{d}t^2}(t_r), \tag{4}$$

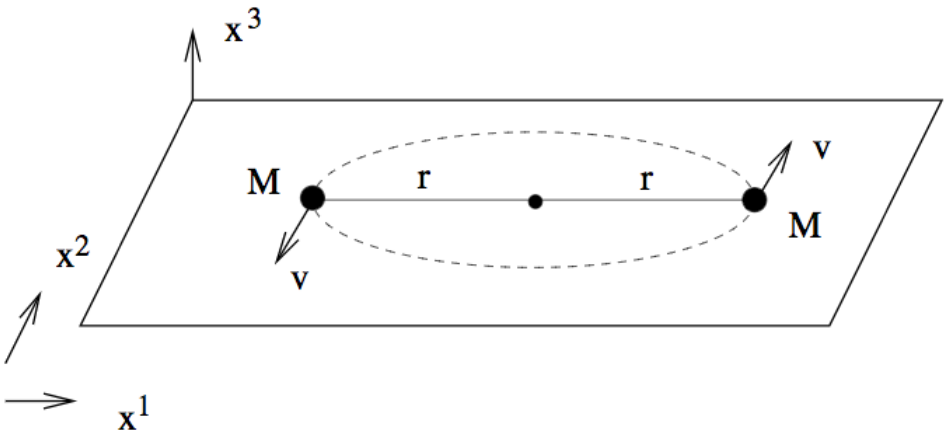

Figure 1: Schéma illustrant une situation où deux étoiles de masses M orbitent l'une autour de l'autre dans le plan (x^1, x^2) autour de leur centre de masse avec une vitesse transverse v et un rayon orbital r.

où t_r est le temps retardé et Q_{ij} est le moment quadrupolaire exprimé de la manière suivante,

$$Q_{ij}(t) = 3 \int y^i y^j T^{00}(t, \mathbf{y}) \mathrm{d}^3 y. \tag{5}$$

On observe alors que l'onde gravitationnelle créée résulte de la variation temporelle du second moment de la densité d'énergie T^{00}. Il est intéressant de noter que ce méchanisme de formation est fondamentalement différent de celui des ondes éléctromagnétiques où la création des champs éléctriques et magnétiques est essentiellement dictée par l'évolution de dipôles. Dans le cas où deux étoiles de masses M orbitent l'une autour de l'autres avec une vitesse orbitale angulaire ω, comme représenté sur la Figure 1, l'expression finale des ondes gravitationnelles donnée par la formule du quadrupole est,

$$\overline{h}_{ij}(t, \mathbf{x}) = \frac{8GM}{Rc^4} \omega^2 r^2 \begin{pmatrix} -\cos(2\omega t_r) & -\sin(2\omega t_r) & 0 \\ -\sin(2\omega t_r) & \cos(2\omega t_r) & 0 \\ 0 & 0 & 0 \end{pmatrix}. \tag{6}$$

On observe que la fréquence de l'onde gravitationnelle est égale au double de la fréquence orbitale du système des deux étoiles.

Finalement, il est intéressant de considérer quel est l'effet mesurable d'une onde gravitationnelle sur un cercle de particules test de matière en chute libre. La théorie prévoit que les ondes gravitationnelles ont un effet sur les distances de séparation entre plusieurs masses tests qui dépend des polarisations de l'onde comme illustré sur la Figure 2. Dans la figure du dessus, nous représentons l'évolution du cercle de particules affectée par la polarisation + de l'onde gravitationnelle écrite h^+. Dans ce cas, on observe que le cercle de particules est compréssée et étirée dans les directions x et y. Dans la figure du dessous, nous représentons l'effet de la déformation pour une polarisation $\times$ où l'on observe que le motif de déformation est alors en rotation de 45 deg par rapport au cas précédent.

Étant donné que les ondes gravitationnelles ont un effet mesurable sur la matière, il est alors possible de construire des appareils de mesure capables de détecter ces ondes. Cependant, l'amplitude des ondes gravitationnelles est extrêmement faible, avec une amplitude typique de l'ordre de $h \sim 10^{-21}$ pour la coalescence d'objets compacts. Il est alors capital d'avoir des instruments de mesures possédant une très grande précision pour pouvoir être capable de mesurer une si petite déformation. Ceci est la raison pour laquelle, l'interférométrie laser a été privilégiée comme méthode expérimentale pour la détection des ondes gravitationnelles.

Un interéromètre est construit sur le principe que les interférences produites par la différence de phase lors de la recombinaison de deux rayons lasers se propageant selon deux axes orthogonales ou bras peuvent être utilisés pour mesurer des différences de longueur très précises. En effet, la déformation d'un bras de l'interféromètre situé sur l'axe y induite par le passage d'une onde gravitationnelle se propageant selon l'axe z comme représenté schématiquement sur la Figure 3 s'écrit de la manière suivante,

$$\frac{\delta L_p}{L_p} = \frac{1}{2} h^{TT}_{yy}(t, z = 0), \tag{7}$$

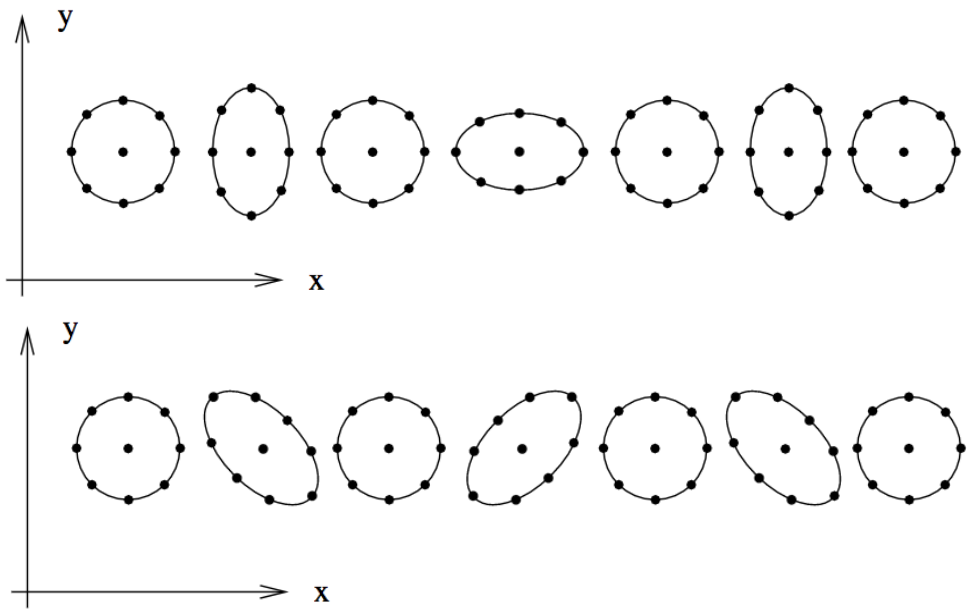

Figure 2: Illustration de la déformation d'un cercle de particules tests en chute libres dûe au passage d'une onde gravitationnelle pour la polarisation h^+ (haut) et $h^\times$ (bas).

où L_p est la longueur du bras de l'interféromètre et δL est la déformation induite par l'onde gravitationnelle h_{yy}^{TT}. Il faut souligner ici que plus la longueur des bras de l'interféromètre est grande, plus grande est la capacité à mesurer des perturbations de faible amplitude. C'est la raison pour laquelle les détecteurs actuels d'ondes gravitationnelles sont des interféromètres avec des longeurs de bras de l'ordre du kilomètre.

Grâce à l'interférométrie laser, il est donc possible de pouvoir mesurer la faible perturbation induite par le passage d'une onde gravitationnelle. Cependant, il est tout de même nécessaire de comprendre et réduire les nombreuses sources de bruit qui viennent contaminer les mesures. À haute fréquence, les fluctuations quantiques du laser viennent perturber les mesures et induisent un bruit nommé "photon shot noise". Dans la bande intermédiaire de fréquences du détecteur, la source principale de bruit provient des fluctuations thermiques des miroirs de l'interféromètre et de leurs suspensions. Enfin dans le domaine des basses fréquences, les perturbations sismiques de la Terre ainsi que les variations locales du champ de gravité induisent un mouvement non voulu sur les miroirs qui doit être atténué au maximum.

Au cours des dernières années, plusieurs détecteurs d'ondes gravitationnelles basés sur le principe de l'interférométrie laser ont été construits. Trois détecteurs sont actuellement en train de prendre des mesures: l'interféromètre européen Virgo situé à Cascina et les deux interféromètres américains LIGO situés aux Etats-Unis à Hanford et Livingston. Ces détecteurs ont subi une phase d'amélioration globale de plusieurs années afin de pouvoir réduire les différents bruits du détecteur, et fonctionnent actuellement dans leur version avancée, advanced Virgo et advanced Ligo. Les détecteurs Advanced Ligo ont commencé à prendre des mesures en septembre 2015 et le détecteur advanced Virgo a commencé à prendre des mesures en août 2017. En plus de ces trois détecteurs, un réseau terrestre de détecteurs d'ondes gravitationnelles est en train d'être construit pour pouvoir être opérationnel dans les années à venir. Le prochain détecteur à être en ligne sera le détecteur japonais KAGRA situé dans la mine de Kamioka. En parallèle, le projet de détecteur LIGO India en Inde est en phase de développement et devrait être opérationnel vers 2025. Tous ces détecteurs font partie de la seconde génération de détecteurs d'ondes gravitationnelles. La troisième génération de détecteurs est également en train d'être planifiée avec la mise en place du projet européen Einstein Telescope et du projet américain Cosmic Explorer. Ces détecteurs seront basés sur un nouvel ensemble de technologies qui leur permettront d'améliorer nettement la sensitivité actuelle des détecteurs de seconde génération. Enfin, la mission LISA est un projet de détecteur spatial d'ondes gravitationnelles qui aura accès à une bande de fréquence différente des détecteurs terrestres et centrée autour du mHz. Pour le moment, la mission a été sélectionnée comme mission L3 de l'ESA et devrait être lancée aux environs de 2034.

Comme vu précédemment, l'amplitude des ondes gravitationnelles est très petite et par conséquent

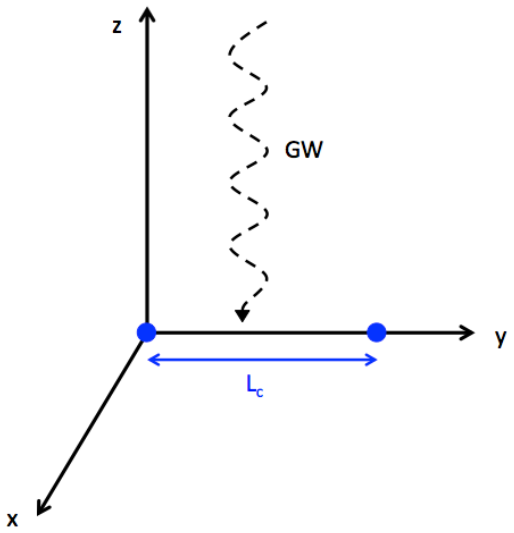

Figure 3: Illustration d'un example où l'on considère la variation de longueur propre L_c sur l'axe y correspondant au passage d'une onde gravitationnelle se propageant sur l'axe z.

ces ondes ne peuvent êre mesurées que lorsque elles sont produites par certains phénomènes et objets astrophysiques. Les principales sources d'ondes gravitationnelles sont les étoiles binaires formées de deux objets dits compacts. Ces objets compacts sont les produits de la fin de vie d'une étoile, c'est-à-dire les naines blanches, les étoiles à neutrons et les trous noirs. Les naines blanches sont le résultat de l'évolution stellaire d'une étoile de faible masse. Dans ces étoiles, la matière est dite dégénérée et la force de gravitation est alors compensée par la pression dégénérée des électrons. Les étoiles à neutrons sont elles le résultat de l'évolution stellaire d'étoile de grandes masses et dans ce cas la force de gravitation est compensée par la pression dégénérée des neutrons. Enfin, les trous noirs représentent le cas extrême où une étoile de grande masse en fin de vie ne parvient plus à supporter sa force de gravitation interne et s'effondre en une singularité de l'espace-temps.

Lorsque deux de ces objets compacts orbitent l'un autour de l'autre, on dit que le système formé est une binaire compacte. Du fait de la variation du moment quadrupolaire de ces objets, ils émettent des ondes gravitationnelles qui peuvent être mesurées par les détecteurs comme advanced LIGO et advanced Virgo. La mesure des ondes gravitationnelles émises par ces objets est capitale et nous permet de mieux comprendre la physique fondamentale de ces objets astrophysiques comme par example leur taux de formation ou leur méchanismes de formation. De plus, l'information véhiculée par les ondes gravitationnelles est à la fois différente et complémentaire de celle apportée par d'autres messagers physiques tels que les ondes électromagnétiques ou les neutrinos.

Le 14 Septembre 2015, les deux détecteurs advanced LIGO ont fait la première détection directe et cohérente des ondes gravitationnelles. Sur la Figure 4, on peut voir les signaux temporels de la déformation mesurées par les détecteurs advanced LIGO provoquée par le passage de l'onde gravitationnelle. Cette onde a été générée lors de la coalescence d'une binaire compacte formée de deux trous noirs de masses stellaires 29 et $36M_\odot$ situés à une distance de 410 Mpc. La coalescence de deux objets compacts correspond au moment où la séparation orbitale entre les deux objets est petite et les deux objets orbitent l'un autour de l'autre de plus en plus rapidement. Du fait de l'émission des ondes gravitationnelles, le système perd de l'énergie et les deux objets se rapprochent l'un vers l'autre jusqu'à ce que les deux objets fusionnent l'un avec l'autre pour former un autre object compact. Dans ce cas, les deux trous noirs de masses stellaires ont formé un trou noir de masse stellaire de 62 $M_\odot$. Du fait de l'augmentation de la fréquence orbitale de la binaire lors de la coalescence, la fréquence des ondes gravitationnelles émises durant ce phénomène augmente de manière à créer un soi-disant "chirp" comme représenté sur le diagramme temps-fréquence en bas de la Figure 4.

Cette première détection directe des ondes gravitationnelles a été d'une très grande importance pour la communauté astrophysique. En effet, cette détection a tout d'abord permis de mettre en évidence directe l'existence des trous noirs. De plus, cette mesure constitue également une autre validation de la théorie de la relativité générale puisque les ondes mesurées étaient celles prédites par Einstein il y a cent ans. Depuis cet événement, deux autres détections d'ondes gravitationnelles émises par la coalescence

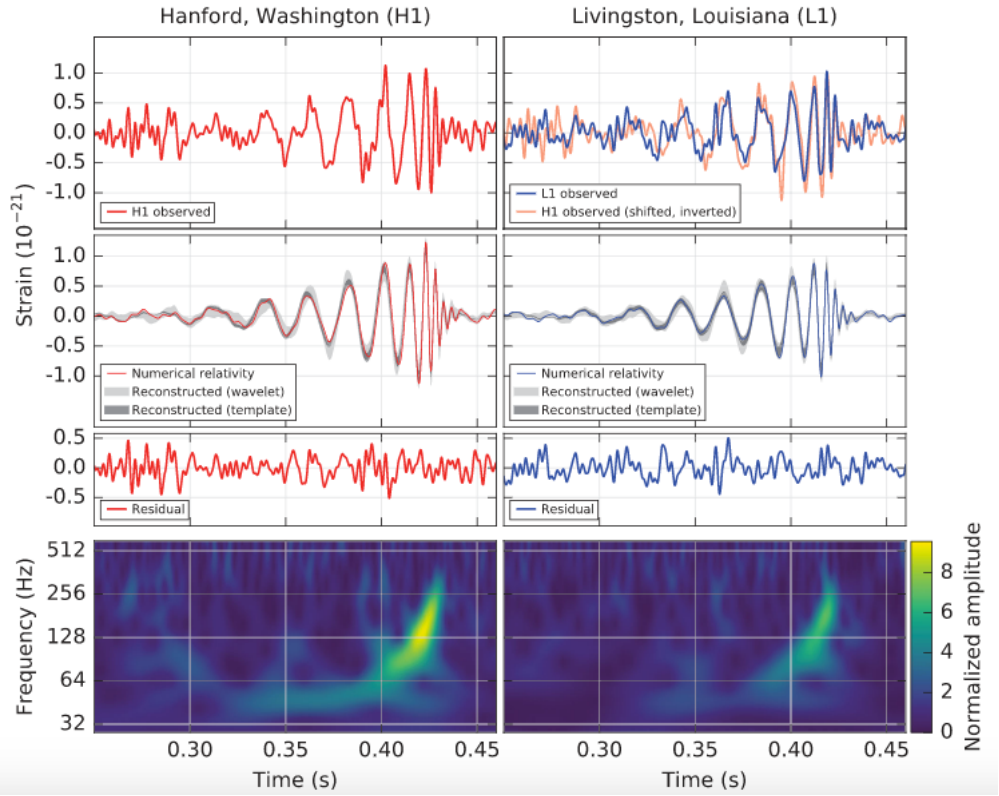

Figure 4: Figure présentant les signaux mesurés par les interféromètres advanced LIGO de l'événement GW150914 correspondant à la coalescence de deux trous noirs de masses stellaires.

de deux trous noirs ont été détectés par les deux détecteurs advanced LIGO le 26 décembre 2015 et le 4 janvier 2017. Enfin, du fait de l'entrée du détecteur advanced Virgo dans le réseau des détecteurs terrestres en août 2017, la première détection triple d'une coalescence de binaire de trous noirs stellaires a été faite en août 2017. De nombreuses autres détections sont attendues dans le futur avec la possibilité de voir également la coalescence de binaires compactes formées d'étoiles à neutrons.

Malgré les nombreuses avancées technologiques et scientifiques des détecteurs d'ondes gravitationnelles, le signal gravitationnel est la plupart du temps noyé dans le bruit de l'instrument. Il est alors essentiel d'avoir des techniques d'analyse de données qui permettent d'analyser et d'interpréter les données brutes mesurées par le détecteur. On peut noter deux aspects complémentaires de l'analyse de données pour les ondes gravitationnelles: la recherche des signaux et l'estimation de paramètres. Dans le premier cas, le but est d'analyser les données afin de rechercher un possible signal d'ondes gravitationnelles en utilisant un modèle théorique ou "template" de l'onde gravitationnelle qui dépend d'un ensemble de paramètres physiques λ^μ. En construisant une statistique, il est alors possible de définir avec quelle confiance le signal identifié est effectivement un signal d'origine astrophysique et non un bruit de l'instrument de mesure. Une fois qu'un signal a été identifié, l'estimation de paramètres permet de faire une analyse du signal afin de déterminer la distribution des paramètres λ^μ du signal, tel que les masses des objets ou la position dans le ciel de la source. La détermination des paramètres est alors exprimée en termes de distribution de probabilités reflétant les erreurs possibles sur la valeur des paramètres du fait du bruit dans le signal.

Pour réaliser cette étude, une des méthodes utilisées dans la communauté des ondes gravitationnelles est le "matched filtering" qui consiste à faire la corrélation croisée entre le signal mesuré par le détecteur et un template théorique du signal gravitationnel. Considérons maintenant que le signal temporel mesuré par un détecteur, $s(t)$, est l'addition d'un signal gravitationnel $h(t)$ et du bruit $n(t)$,

$$s(t) = h(t) + n(t). \tag{8}$$

Nous assumons aussi dans ce cas que le bruit est à la fois stationnaire et Gaussien. Afin de pouvoir déterminer la présence ou non du signal gravitationnel $h(t)$ dans $s(t)$, on peut utiliser comme expliqué précédemment un ensemble de "templates" ou modèles théoriques d'ondes gravitationnelles $h(t, \lambda^\mu)$ dépendent d'un certain nombres de paramètres λ^μ qui décrivent la physique du système. Dans le cadre

de la méthode du matched filtering, il est possible de construire une corrélation dans le domaine de Fourier entre le signal mesuré $\tilde{s}(f)$ et le template de l'onde gravitationnelle $\tilde{h}(f, \lambda^\mu)$. Cette corrélation nous permet de définir à la fois le ratio signal sur bruit ρ, et la log-vraisemblance, $\ln \mathcal{L}(\lambda^\mu)$, de la manière suivante

$$\rho = \frac{\langle s \mid h(\lambda^\mu)\rangle}{\sqrt{\langle h(\lambda^\mu) \mid h(\lambda^\mu)\rangle}} \tag{9}$$

$$\mathcal{L}(\lambda^\mu) = \exp\left[-\frac{1}{2}\langle s - h(\lambda^\mu)|s - h(\lambda^\mu)\right], \tag{10}$$

où l'on définit le produit scalaire

$$\langle h|g\rangle = 2\int_0^\infty \frac{\tilde{h}(f)\tilde{g}^*(f) + \tilde{h}^*(f)\tilde{g}(f)}{S_n(f)}\mathrm{d}f. \tag{11}$$

Une fois qu'un signal a été identifié, il faut alors utiliser des méthodes d'analyse capables de donner des estimations de la valeur des paramètres λ^μ à partir de la mesure s. Pour faire çela, il est possible d'utiliser l'inférence Bayésienne. Dans cette approche, les paramètres λ^μ sont considérés comme des variables aléatoires déterminées par une densité de probabilité $p(\lambda^\mu)$. Le but de l'estimation de paramètres est de calculer cette probabilité de distribution qui contient toutes les informations nécessaires sur λ^μ. Pour pouvoir calculer cette densité de probabilité, on utilise le théorème de Bayes qui s'exprime de la manière suivante

$$p(\lambda^\mu|s, M) = \frac{p(s|\lambda^\mu, M)p(\lambda^\mu|M)}{p(s|M)}, \tag{12}$$

où $p(\lambda^\mu|s, M)$ est dénommée "posterior distribution", $p(s|\lambda^\mu, M)$ est la vraisemblance, $p(\lambda^\mu|M)$ est la "prior distribution" et $p(s|M)$ est l'évidence. La grande force de ce théorème est d'exprimer la densité de probabilité qui nous intéresse, ici la "posterior distribution", à l'aide de trois autres quantités qui peuvent êtres calculées plus simplement.

Cependant, étant donné la complexité de l'espace des paramètres λ^μ et le fait que l'on ne connaisse pas à priori sur quels intervalles de valeurs la posterior distribution prend des valeurs non négligeables, il est nécessaire d'utiliser des algorithmes adaptés pour pouvoir avoir accès à la distribution $p(\lambda^\mu|s, M)$ en un temps raisonnable. Ce problème est intimement relié avec le problème de l'intégration multi-dimensionelle d'une surface à géométrie complexe. Parmi les algorithmes les plus efficaces pour réaliser ce travail, on peut citer les méthodes stochastiques dites de Monte Carlo avec chaînes de Markov ou MCMC. Certaines méthodes MCMC comme l'algorithme de Metropolis-Hastings et le Nested Sampling ont donc été mis en place par la collaboration LIGO/Virgo dans leur chaîne d'analyse de données pour les signaux gravitationnels. Les méthodes ont été appliquées sur les événements mesurés par les détecteurs terrestres et ont permis de fournir les distributions de paramètres associées à ces différentes sources.

Comme expliqué précédemment, de nombreuses détections sont prévues dans le futur grâce à l'amélioration de la sensibilité des instruments de détection. Il est alors capital d'avoir des algorithmes aussi performants que possible pour pouvoir traiter les données en un temps acceptable et fournir les distributions de probabilité permettant de tirer des conclusions physiques. C'est dans ce contexte que s'inscrit ce travail de thèse qui a été centré sur le développement d'algorithmes pour l'analyse de données pour les ondes gravitationnelles.

L'algorithme Hamiltonian Monte Carlo ou HMC est une méthode de type MCMC qui est particulièrement efficace pour traiter des problèmes multi-dimensionels à géométrie complexe comme ceux que nous rencontrons pour les ondes gravitationnelles. Le principe de l'algorithme est de considérer que l'inverse de la surface de log-vraisemblance peut être vue comme correspondant à la surface de potentiel créée par un champ gravitationnel. En considérant des particules se déplacant sur cette surface et paramétrisées par leur position q^μ, pris comme étant égale aux paramètres physiques λ^μ, et leur moments p^μ, on peut résoudre les équations de la mécanique classique d'Hamilton pour proposer la prochaine itération de la chaîne de Markov dans l'espace de paramètres. Cette technique a l'avantage de prendre en considération la géométrie de la surface de log-vraisemblance, ce qui impacte grandement le taux d'acceptation et l'exploration de l'algorithme. Empiriquement, il a été démontré que cette méthode a la capacité d'être D fois plus efficace qu'une méthode classique MCMC où D est la dimension du problème considéré.

L'inconvénient majeur de la méthode HMC qui a empêché sa diffusion provient du fait qu'il faut calculer le gradient de la log-vraisemblance pour résoudre les équations d'Hamilton. Le temps nécessaire

pour cette opération peut souvent rendre l'algorithme globalement non compétitif par rapport aux autres méthodes MCMC si aucune solution n'est apportée pour réduire le temps de calcul du gradient. Dans une étude réalisée sur l'estimation de paramètres pour les ondes gravitationnelles émises pendant la coalescence de trous noirs supermassifs mesurées par LISA, il a été montré que l'algorithme HMC était capable d'être très performant et que le temps de calcul du gradient pouvait être réduit drastiquement en utilisant une méthode de fit polynomial. Cette étude a été le point de départ du travail conduit durant ce projet de thèse dont l'objectif consistait à reproduire l'algorithme HMC avec la méthode de fit pour l'estimation de paramètres dans le cas d'étoiles binaires à neutrons détectées par LIGO/Virgo.

Le modèle que nous avons utilisé pour simuler les ondes gravitationelles émises par les binaires d'étoiles à neutrons ou BNS est le modèle TaylorF2. Ce modèle dépend d'un ensemble de neuf paramètres physiques et donne l'expression de la forme d'onde directement dans le domaine de Fourier. Il est important de signaler que nous avons uniquement modélisé la forme d'onde durant la partie d'inspiral de la coalescence. Ceci est justifié par le fait que la fréquence à laquelle les deux étoiles à neutrons fusionnent est dans le domaine des hautes fréquences du détecteur qui est contaminé par le bruit de fluctuation quantique du laser. En termes de sources considérées pour cette étude, nous avons décidé d'utiliser un échantillon de dix sources provenant d'un catalogue élaboré pour une étude réalisée dans le passé par la collaboration LIGO/Virgo. Concernant les détecteurs, nous avons décidé de considérer dans cette étude un réseau de détecteurs comprenant les deux advanced LIGO et advanced Virgo à leur sensitivité de design. Ceci implique que la position de la source dans le ciel est bien localisé du fait de la triangulation de la source calculée à partir des délais d'arrivée de l'onde sur chacun des trois détecteurs. Enfin nous avons choisi la paramétrisation suivante pour faire tourner nos algorithmes,

$$q^\mu = \{\cos\iota, \phi_c, \psi, \ln D_L, \ln \mathcal{M}_c, \ln\mu, \sin\theta, \phi, t_c\} \tag{13}$$

où ι est l'inclinaison de la binaire, ϕ_c est la valeur de phase à la coalescence, ψ l'angle de polarisation de l'onde, D_L la distance de luminosité, $\mathcal{M}_c$ la chirp mass, μ la masse réduite, θ la colatitude, ϕ la longitude et t_c le temps à la coalescence. Pour les distributions de prior, nous avons utilisé des prior non informatifs pour tous les paramètres en limitant les valeurs des masses des objets entre 1 et 2.6 $M_\odot$ et la distance de luminosité entre 10^{-6} et 200 Mpc.

La première étape du travail que nous avons réalisé sur l'algorithme HMC a été dédié au "fine-tuning" des paramètres libres de l'algorithme. En effet, puisque nous ne pouvons pas résoudre analytiquement les équations d'Hamilton, il est nécessaire de les résoudre de manière discrète sur une trajectoire. Pour cela, on peut utiliser une méthode de résolution sympletique dénommée "leapfrog" qui consiste à faire évoluer à chaque pas de temps ϵ, les positions q^μ et moments p^μ de la manière suivante,

$$\begin{aligned}
\tilde{p}^\mu(\tau+\epsilon^\mu/2) &= \tilde{p}^\mu(\tau) + \frac{\epsilon^\mu}{2}\frac{\partial \ln[\mathcal{L}(q^\mu)]}{\partial q^\mu}\Big|_{q^\mu(\tau)}, \\
q^\mu(\tau+\epsilon^\mu) &= q^\mu(\tau) + \epsilon^\mu \tilde{p}^\mu(\tau+\epsilon^\mu/2), \\
\tilde{p}^\mu(\tau+\epsilon^\mu) &= \tilde{p}^\mu(\tau+\epsilon^\mu/2) + \frac{\partial \ln[\mathcal{L}(q^\mu)]}{\partial q^\mu}\Big|_{q^\mu(\tau+\epsilon^\mu)},
\end{aligned} \tag{14}$$

où l'on définit $\tilde{p}^\mu = s_\mu p^\mu$, $\tilde{\epsilon}^\mu = s_\mu \epsilon$ et s^μ comme étant une valeur d'échelle reliée au problème. Les équations précédentes sont répétées un nombre l fois le long de la trajectoire. En résumé, l'algorithme HMC introduit donc trois paramètres libres: les valeurs d'échelles s^μ, le pas de temps ϵ et la longueur de la trajectoire l.

Pour la longueur de la trajectoire, nous avons fixé la valeur à $l = 200$ car cette valeur avait fourni des résultats satisfaisants dans le cas de l'étude réalisée pour LISA. Pour la valeur d'échelle s^μ, nous avons conduit plusieurs tests et avons trouvé que nous obtenions une bonne exploration de l'espace de paramètre dans le cas où s^μ est pris comme étant égale à l'inverse de la matrice d'information de Fisher $\Gamma^{\mu\nu}$. Ce dernier résultat peut être compris par le fait que notre problème possède des valeurs dynamiques d'échelle très différentes suivant les paramètres considérés. Enfin, concernant l'optimisation du pas de temps ϵ, nous avons lancé une série de simulations de 500 trajectoires avec différentes valeurs de pas de temps ϵ allant de $\epsilon = 10^{-4}$ à $\epsilon = 10^{-2}$. En faisant un compromis entre taux d'acceptation et exploration de l'algorithme, nous avons trouvé que l'algorithme présentait des performances satisfaisantes dans le cas où $\epsilon = 2.5 \times 10^{-3}$.

À ce moment de l'étude, nous avions donc un algorithme qui était à la fois performant en termes de taux d'acceptation (supérieur à 90%) et d'exploration de l'espace des paramètres. Cependant, nous étions alors confrontés au problème détaillé auparavant qui est le temps de calcul élevé pour évaluer le

gradient de la log-vraisemblance par rapport aux neuf paramètres à chaque étape de la trajectoire. Nous avons décidé de tester en premier lieu la méthode de fit cubique mise en place dans le cas de LISA, qui consiste à approximer chacun des gradients de la log-vraisemblance par la fonction suivante,

$$f(q^\mu) = \sum_{i=1}^{D} a_i q^i + \sum_{j=1}^{D}\sum_{k=j}^{D} a_{jk} q^j q^k + \sum_{l=1}^{D}\sum_{v=l}^{D}\sum_{w=v}^{D} a_{lvw} q^l q^v q^w, \tag{15}$$

où D est la dimension du problème et a_i, a_{jk} et a_{lvw} représentent les coefficients du fit. Pour trouver la valeur des coefficients, on peut alors appliquer une méthode des moindres carrées en utilisant un ensemble de valeurs pour les gradients qui aura été calculé au préalable. Du point de vue de l'algorithme, cela signifie que nous avons désormais trois phases distinces. Durant la phase I, on fait tourner l'algorithme HMC sur N trajectoires en utilisant le gradient calculé de manière numérique et en gardant en mémoire les valeurs du gradient des points des trajectoires qui ont été acceptées. Dans la phase II, nous utilisons les valeurs en mémoire afin de dériver les valeurs des coefficients pour le fit cubique grâce à une méthode des moindres carrés. Enfin pendant la phase III, on utilise les valeurs dérivées pour les coefficients pour calculer les valeurs analytiques du gradient afin d'augmenter la rapidité de l'algorithme pour toutes les trajectoires restantes. Il faut souligner que la qualité du fit en phase II dépend du nombre de points génnérés en phase I intrinséquement relié au nombre de trajectoires numériques N.

Nous avons implémenté cette méthode dans notre cas et avons lancé une série de tests en prenant différentes valeurs pour N. Sur le graphe à gauche de la Figure 5, nous représentons l'évolution du taux d'acceptation durant la phase III de l'algorithme en fonction du nombre d'itérations pour les différentes valeurs de N. On aperçoit que le taux d'acceptation a une valeur maximale de 45% dans le cas où $N = 3000$ ce qui correspond à près de 600000 points pour le fit. Cette valeur est beaucoup plus basse que la valeur de 95% obtenue durant la phase I, et indique donc que le fit ne fonctionne pas correctement. Pour mieux évaluer la situation, nous avons décidé de faire un autre test où cette fois nous avons fixé le nombre de trajectoires numériques initiales à $N = 750$ durant la phase I et utilisé l'approximation analytique du gradient pour un seul paramètre en gardant le calcul des autres gradients numériques dans la phase III. Sur le graphe à droite de la Figure 5, nous présentons l'évolution du taux d'acceptation que nous avons obtenu en fonction du nombre d'itérations pour les différentes simulations, en indiquant dans la légende le paramètre pour lequel nous avons utilisé l'approximation analytique. Les résultats obtenus font apparaître deux cas différents suivants les paramètres. Pour le premier ensemble de paramètres, $\{\ln \mathcal{M}_c, \ln \mu, \phi_c, \ln t_c, \sin(\theta), \phi\}$, le taux d'acceptation reste presque constant durant la phase III et est proche de 95%. Dans le cas de l'ensemble de paramètres $\{\cos \iota, \psi, \ln D_L\}$, on observe une chute du taux d'acceptation lorsque nous utilisons l'approximation du gradient qui est encore plus accentuée dans le cas de $\cos \iota$ et $\ln D_L$. L'interprétation de ce résultat peut se faire en remarquant que ces paramètres sont intrinsèquement reliés à la multimodalité de la posterior distribution, ce qui pourrait expliquer pourquoi le fit ne parvient pas à produire une bonne approximation des gradients de la log-vraisemblance.

Après avoir identifié le problème, nous avons essayé différentes méthodes pour pouvoir améliorer la qualité du fit pour l'ensemble de paramètres concerné et une partie importante du travail de thèse a été dédiée à résoudre ce problème non trivial. La première idée que nous avons eue a été d'augmenter l'ordre du polynôme utilisé pour approximer le gradient en allant à l'ordre quartique et quintique. Nous avons relancé la même série de simulations réalisées pour le cas cubique et avons observé que cette solution ne permettait pas d'améliorer la qualité du fit avec toujours un faible taux d'acceptation durant la phase III de l'algorithme. La deuxième proposition pour résoudre ce problème a été d'essayer de mieux modéliser la bimodalité de la distribution en inclinaison en réalisant deux fit indépendants, l'un pour ι inférieur à $\pi/2$ et l'autre pour ι supérieur à $\pi/2$. Dans ce cas, nous avons trouvé que le taux d'acceptation durant la phase III était légèrement meilleur avec des valeurs proches de 50%. Même si ce résultat n'est toujours pas aussi bon que souhaité, cela nous a poussé à regarder d'autres méthodes de fit prenant plus en compte la géométrie locale de la surface de log-vraisemblance. C'est la raison pour laquelle nous avons ensuite essayé une méthode nommée "radial basis functions" pour approximer le gradient. Le principal problème de cette méthode provient du fait qu'il est nécessaire d'inverser une matrice carré de taille n où dans notre cas n est de l'ordre de 10^5. Cette inversion s'est révélée problématique et ne nous a pas permis d'obtenir de bonnes approximations pour le gradient.

Finalement, nous avons réussi à trouver une solution pour approximer le gradient en utilisant une méthode de fit locale avec des tables de correspondance triées. La première étape de la méthode est de réarranger les points obtenus pour faire le fit durant la phase I dans trois tableaux qui sont triés respectivement selon $\cos \iota$, ψ et $\ln D_L$. Ensuite, lorsque l'on doit approximer la valeur du gradient à un point donné durant la phase III de l'algorithme, on sélectionne les n_1 points les plus proches dans

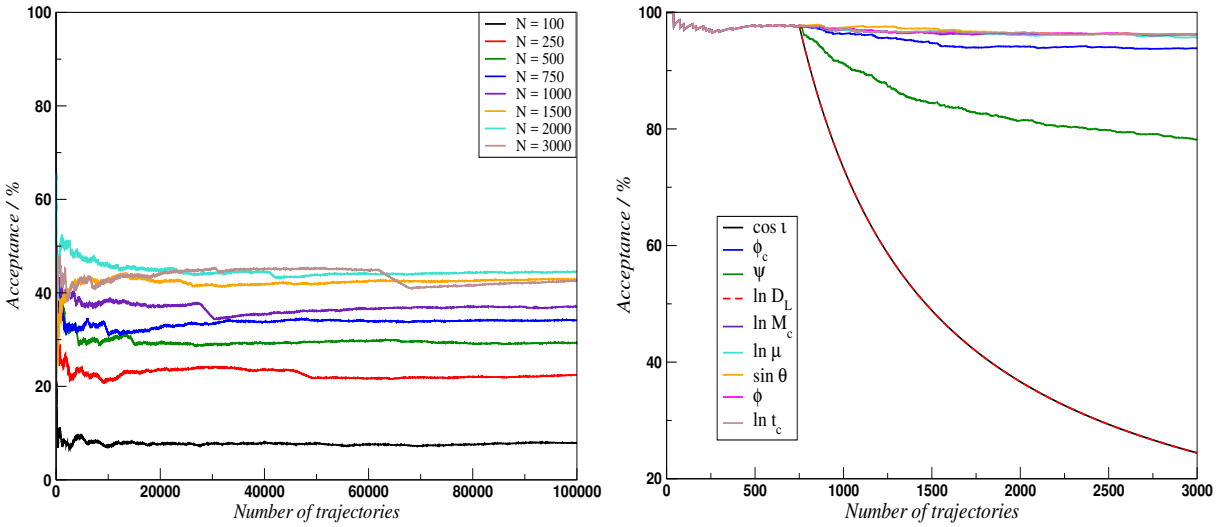

Figure 5: (gauche) Taux d'acceptation en fonction du nombre de trajectoires durant la phase III de l'algorithme en utilisant un fit cubique et différentes valeurs de trajectoires numériques initiales N. (droite) évolution du taux d'acceptation pour une simulation de 3000 trajectoires où les 750 premières trajectoires utilisent une valeur de gradient numérique et les trajectoire restantes utilisent des valeurs numériques pour le gradient excepté pour le paramètre donné dans la légende.

le tableau afin de pouvoir obtenir une bonne représentation locale de l'espace de paramètres. Du fait de la bimodalité de notre posterior distribution, nous avons également rajouté une autre étape où nous sélectionnons un sous-ensemble n_2 des n_1 points en utilisant un critère de distance dans le sous-espace des paramètres $\cos\iota, \psi, \ln D_L$. Pour le choix de la distance nous avons testé à la fois des distances de type Euclidienne et Mahalanobis. Enfin ,pour pouvoir ajuster les différents paramètres du fit, c'est à dire les nombres n_1, n_2 et le choix de distance, nous avons lancé plusieurs séries de simulations en regardant l'évolution du taux d'acceptation en fonction du nombre d'itérations comme représenté sur la Figure 6. On observe ici que dans le cas $n_1 = 2000$ et $n_2 = 100$ avec une distance de type Euclidienne, la méthode de fit nous permet d'obtenir un taux d'acceptation proche de 80% durant la phase III, ce qui démontre que notre méthode parvient à bien approximer le gradient.

À ce point du projet, nous avions un algorithme HMC performant qui était capable de tourner en un temps acceptable du fait de la méthode de fit employé durant la phase III. Nous avons alors décidé de tester l'algorithme sur une situation réélle d'estimation de paramètres sur la premire source de notre catalogue en prenant un nombre total de 10^6 trajectoires sur la source 1 de notre catalogue. Pour avoir un moyen de comparaison, nous avions également développé au préalable un algorithme classique de type évolution différentielle à chaînes de Markov ou DEMC. Les principaux crièteres utilisés pour la comparaison était de voir si l'algorithme HMC était capable de produire les bonnes posterior distribution et de voir quel était le taux de production d'échantillons statistiquement indépendants. Sur la Figure 7, nous présentons les distributions que nous avons obtenues avec l'algorithme DEMC (courbe rouge) et HMC (courbe bleue). Nous voyons que l'algorithme HMC est capable à la fois de bien représenter la multimodalité du problème, et est également capable de mieux traiter certaines limites artificielles comme celles du cas où les masses sont égales pour μ. En termes de taux de génération d'échantillons statistiquement indépendants, nous avons également observé que l'algorithme HMC était capable de meilleures performances avec un taux presque un ordre de magnitude plus élevé que dans le cas de l'algorithme DEMC.

Le prochain objectif du travail a ensuite été de vérifier comment l'algorithme HMC se comporte sur l'ensemble des 10 sources sélectionnées dans le catalogue. Les résultats obtenus nous ont montré que sur la moitié des sources l'algorithme produisait de bons résultats comme ceux obtenus dans le cas précédent. Pour le reste des sources, nous avons rencontré un certain nombre de problèmes qu'il nous a fallu résoudre. Le premier problème était relié aux valeurs d'échelles calculées à partir de la matrice d'information de Fisher qui dans certains cas étaient beaucoup trop élevées entraînant un crash de l'algorithme. Pour pallier à cela, nous avons décidé d'introduire un certain nombre de limites pour que les valeurs d'échelles restent dans un domaine de valeurs acceptables. Le deuxième problème majeur rencontré était que dans certains cas l'approximation du gradient avec le fit local dans la phase III de l'algorithme ne parvenait pas à produire de bons résultats ayant pour conséquence de bloquer l'algorithme dans certaines parties

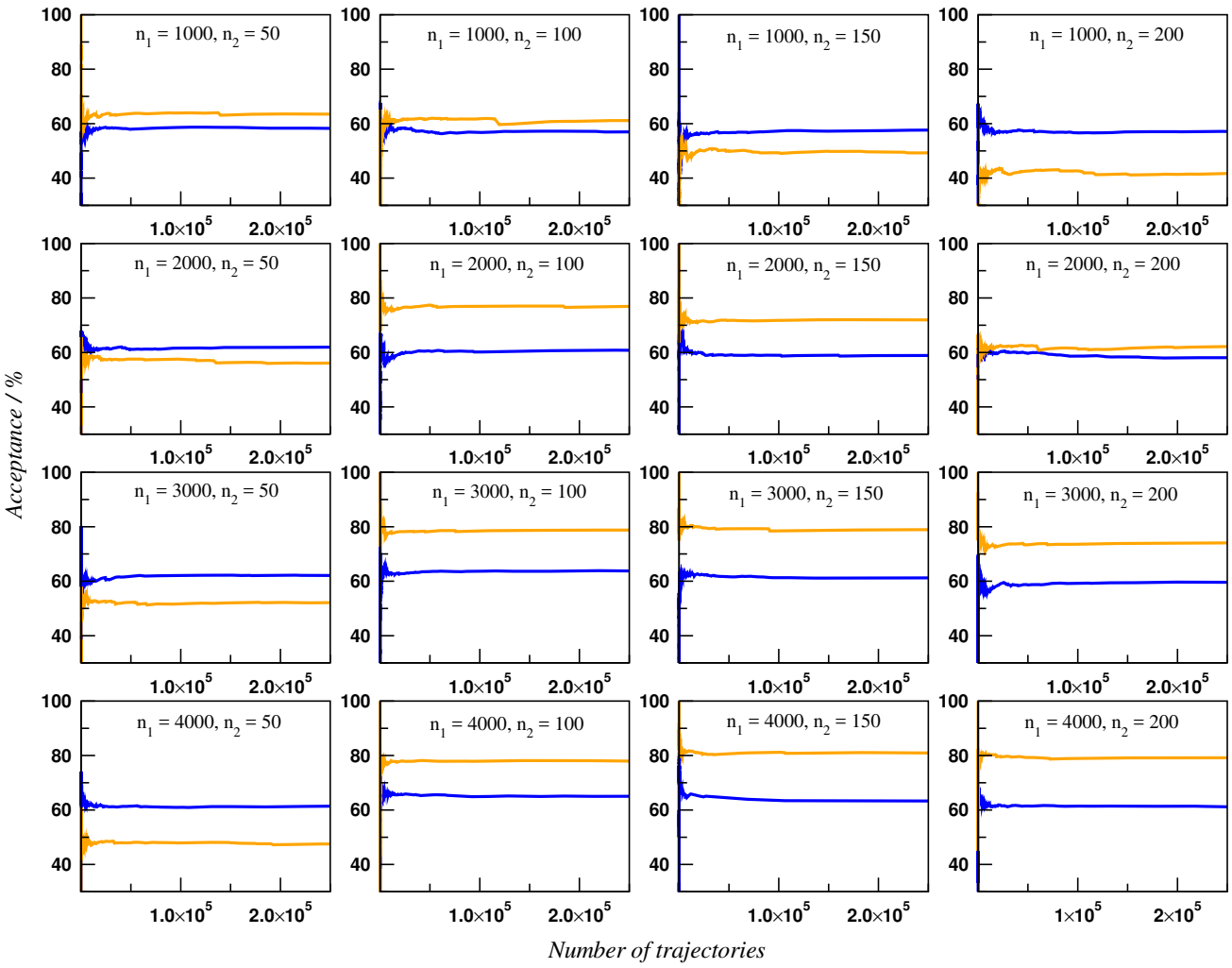

Figure 6: Taux d'acceptation en fonction du nombre de trajectoires durant la phase III de l'algorithme et en utilisant un fit local avec tables de correspondance pour les paramètres $\{\cos\iota, \psi, \ln D_L\}$ et cubiques pour les autres. Le fit initial a été produit en utilisant 1500 trajectoires numériques et différentes valeurs de n_1 et n_2 ont été utilisées pour le fit local. Ces simulations ont été réalisées à la fois dans le cas d'une distance de type Euclidienne (orange) et de Mahalanobis (bleu).

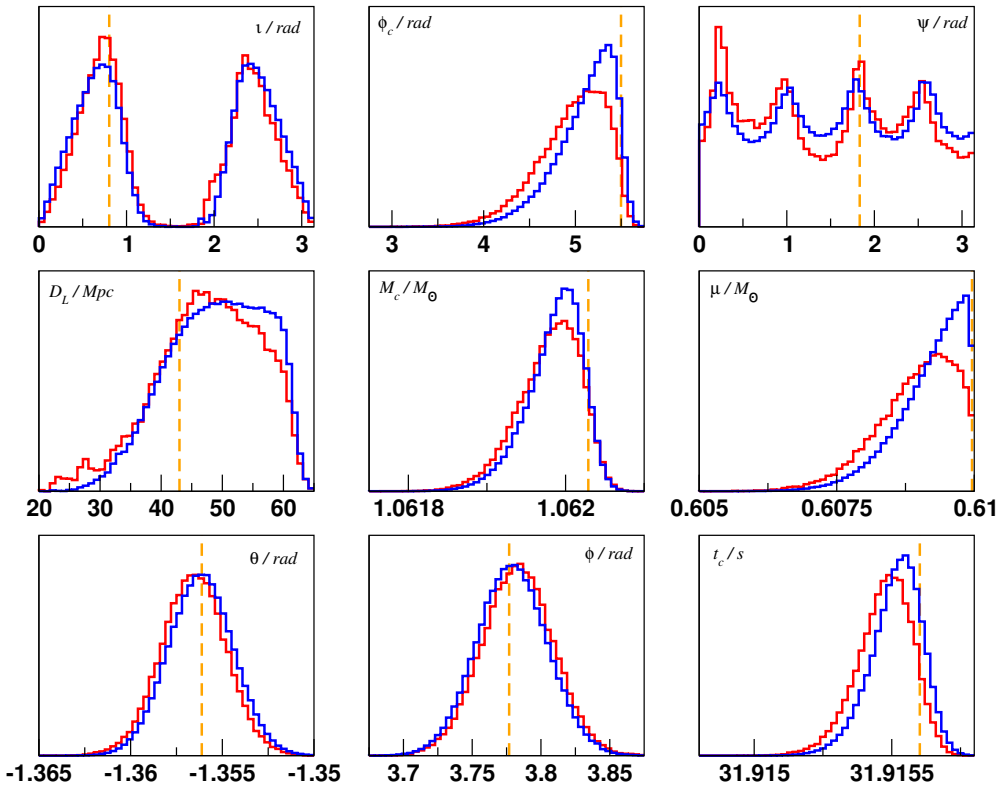

Figure 7: Posterior distributions pour la source BNS1 évaluée à partir d'une chaîne HMC de 10^6 trajectoires (courbe bleue) et d'une chaîne DEMC de 10^6 itérations (courbe rouge). Les valeurs réelles sont représentées par des lignes pointillées orange.

de l'espace de paramètres. La solution mise en place a été d'introduire des trajectoires dites hybrides en phase III où le gradient est calculé numériquement pour $\{\cos\iota, \psi, \ln D_L\}$ et analytiquement pour les autres paramètres. De plus, nous avons également rajouté un autre critère permettant d'utiliser des trajectoires complètement numériques dans le cas où les trajectoires hybrides ne parviennet pas à décoincer la chaîne. Enfin, en améliorant les techniques de fit, nous avons également introduit un raffinement du fit en phase III à la fois en termes de fit cubique pour $\{\ln \mathcal{M}_c, \ln\mu, \phi_c, \ln t_c, \sin(\theta), \phi\}$ et de fit local pour $\{\cos\iota, \psi, \ln D_L\}$.

La nouvelle version de l'algorithme HMC a permis d'obtenir de très bons résultats sur l'ensemble des dix sources. Dans tous les cas, nous avons trouvé que le taux de génération d'échantillons statistiquement indépendants était entre 5 et 20 fois plus grand que dans le cas de l'algorithme DEMC, avec un temps moyen de production d'échantillons proche de 1 seconde pour la majorité des simulations. Bien qu'il n'est pas possible de comparer directement ces résultats avec les algorithmes actuellement utilisés au sein de la collaboration LIGO/Virgo, il est toujours possible d'avoir un ordre d'idées de la possible amélioration des performances dans le cas où l'on utilise l'algorithme HMC. Dans les différentes études présentées dans la littérature, on trouve ainsi que le temps moyen pour produire un sample se situe entre 77 et 227 secondes, ce qui est nettement supérieur au temps que nous avons réussi à obtenir avec notre algorithme.

Ce projet de thèse a donc bien démontré les possibilités d'amélioration que présente l'utilisation de l'algorithme HMC pour l'estimation des paramètres associés aux ondes gravitationnelles émises par des binaires d'étoiles à neutrons. Un nombre de travaux complémentaires est également prévu pour le futur afin de prolonger cette étude. En premier lieu, nous souhaiterions introduire l'algorithme HMC dans les pipelines d'analyse de données de LIGO/Virgo afin de pouvoir le tester dans des conditions réelles d'analyse de données et pour pouvoir faire une véritable comparaison avec les algorithmes actuels. Ensuite, nous voulons également voir comment l'algorithme se comporte avec des modèles de forme d'ondes plus complexes pour les BNS en introduisant par exemple des modélisations des effets de matière avec le paramètre de déformations de marée. Enfin, nous voulons également tester l'algorithme sur d'autres types de sources binaires compactes comprenant un ou deux trous noirs de masses stellaires.

Le second travail réalisé durant cette thèse a été centré sur l'analyse de données pour le futur observatoire spatial LISA. Le but du projet a été de mettre en place un algorithme de recherche pour les sources binaires compactes monochromatiques dans notre Galaxie, qui sont les sources les plus nombreuses qui

seront détectées par LISA. Dans la bande de fréquence de LISA centrée autour du mHz, les binaires compactes sont détectées dans une phase de leur évolution où les deux objets orbitent l'un autour de l'autre à une fréquence orbitale presque constante. On peut donc modéliser la phase de la forme d'onde associée à ces objets de la manière suivante,

$$\Phi(t) = 2\pi f_0 \left[t + R_\oplus \sin(\theta)\cos(2\pi f_m t - \phi)\right], \tag{16}$$

où f_0 est la fréquence de l'onde gravitationnelle, $R_\oplus = 1AU$ est le rayon de l'orbite de LISA et $f_m = 1/yr$ est la fréquence de modulation du détecteur. On observe que cette phase comprend à la fois un terme monochromatique en fréquence dépendant de f_0 ainsi qu'un terme de modulation dépendant de f_m et qui est généré par l'effet Doppler dû au mouvement de LISA autour du Soleil. Au final, le modèle utilisé pour la forme d'onde dépend de sept paramètres que nous avons choisi d'exprimer avec la paramétrisation suivante,

$$\{A, \iota, \phi_0, \psi, \ln f_0, \theta, \phi\}, \tag{17}$$

où A est l'amplitude de l'onde gravitationnelle, ι l'inclinaison de la binaire, ψ l'angle de polarisation, ϕ_0 la phase initiale et (θ, ϕ) la position de la source dans le ciel.

Du point de vue de la recherche de sources, il est possible de réduire la dimensionalité de l'espace de paramètres en utilisant une statistique particulière nommée F-statistique. Cette statistique est obtenue en maximisant analytiquement la log-vraisemblance sur l'ensemble de paramètres $\{A, \iota, \phi_0, \psi\}$. En pratique, cela signifie qu'il est possible de réduire l'espace de recherche seulement sur l'ensemble de paramètres $\{\ln f_0, \theta, \phi\}$. Il faut cependant attirer l'attention ici sur le fait que ceci n'est valable que lors de la recherche de sources. En effet, du point de vue de l'estimation de paramètres, il est important de ne pas maximiser la log-vraisemblance afin de ne pas contraindre les posterior distribution sur le set de paramètres $\{A, \iota, \phi_0, \psi\}$.

Naturellement, le modèle de la forme d'onde pour modéliser les binaires compacts monochromatiques est exprimé dans le domaine temporel. Cependant, comme notre analyse se déroule dans le domaine de Fourier en utilisant le matched filtering, nous avons cherché à réexprimer l'expression de la forme d'onde directement dans le domaine de Fourier. Cela nous permet d'éviter d'utiliser un algorithme de type FFT et par conséquent d'accélerer notre générateur de forme d'onde. En suivant les travaux déjà réalisés dans la littérature, il est possible d'écrire le signal temporel sous la forme d'une série de Fourier comme ceci,

$$s(t) = \sum_n \tilde{s}_n e^{2\pi i n \frac{t}{T_{obs}}}. \tag{18}$$

où T_{obs} est le temps total d'observation. Les coefficients de Fourier $\tilde{s}_n$ sont ensuite exprimés de la manière suivante,

$$\tilde{s}_n = \frac{1}{2} e^{i\varphi_0} \sum_k \tilde{a}_k \sum_l \tilde{b}_l \sum_m \left(A_+ \tilde{p}_m^+ + e^{i3\pi/2} A_\times \tilde{p}_m^\times \right), \tag{19}$$

où $\tilde{a}_k$, $\tilde{b}_l$, $\tilde{p}_m^+$ et $\tilde{p}_m^\times$ sont également des coefficients de Fourier qui peuvent être dérivés en séparant les différentes contributions de la forme d'onde temporelle. De manière similaire, il est également possible de dériver une expression de la log-vraisemblance en utilisant la F-statistique directement dans le domaine de Fourier. Afin de tester la validité de notre approximation de la forme d'onde et de la F-statistique dans le domaine de Fourier, nous avons calculé la corrélation entre nos expressions analytiques dans le domaine de Fourier et les expressions données par la transformée de Fourier des valeurs temporelles. Dans les deux cas, nous avons trouvé que les deux expressions donnaient de très bonnes valeurs de corrélation, ce qui justifie notre usage des expressions analytiques dans le domaine de Fourier dans les algorithmes de recherche que nous allons maintenant présenter.

L'idée principale de ce projet de recherche était de mettre en place un algorithme basé sur un algorithme évolutionnaire nommé Optimisation par essaim moléculaire ou PSO. Cette algorithme s'inspire des mouvements observés dans la nature parmi les groupes ou essaims d'organismes afin de résoudre des problèmes complexes d'optimisation. Dans notre cas, l'idée est de considérer un ensemble de particules sur notre espace tridimensionnel de paramètres où chaque particule est représentée par un vecteur de position $X^i(t)$ et de vitesse $V^i(t)$, tel que i est l'index de l'individu considéré et t est un temps factice. Le but de l'algorithme est de faire évoluer les individus de l'essaim à l'aide des équations suivantes,

$$X^i(t_{j+1}) = X^i(t_j) + V^i(t_j), \tag{20}$$
$$V^i(t_{j+1}) = wV^i(t_j) + c_1\xi_1(P^i(t_j) - X^i(t_j)) + c_2\xi_2(G(t_j) - X^i(t_j)). \tag{21}$$

Ces équations font intervenir un nombre de paramètres libres qu'il faut fixer tel que l'inertie w, c_1 et c_2, ainsi que des nombres ξ_1 et ξ_2 générés de manière uniforme entre 0 et 1. P^i ets nommé meilleure position personnelle et représente la position de l'individu i où la valeur de la log-vraisemblance était la plus élevée dans son histoire passé. De manière similaire G est nommé meilleur position de groupe et représente la meilleur position de l'essaim complet dans son histoire passé. On voit ainsi que l'équation des vitesses s'exprime en fonction de trois termes différents. Le premier terme est le terme usuel d'inertie tandis que les deux autres termes dépendants respectivement de P^i et G sont des termes d'accélération.

Pour pouvoir évaluer les performances de l'algorithme PSO dans notre cas, nous avons tout d'abord voulu tester l'algorithme sur une seule source et dans un domaine réduit de recherche en fréquence. Ceci est justifié par le fait que les sources sont localisées très précisément en fréquence dans l'intervalle de mesure de LISA, ce qui rend la recherche difficile du fait de la complexité de la surface de log-vraisemblance. Cette étude préliminaire a ainsi démontré que nous étions capables de retrouver la source en prenant un intervalle de fréquence d'une largeur maximale de $10^4 f_m \approx 0.1$ mHz. A partir ce de moment, il semblait que nous avions atteint les limites de l'algorithme PSO car nous étions confrontés à deux types de problèmes différents. Dans certains cas, l'algorithme manquait d'exploration globale car celui-ci restait coincé dans des maxima secondaires de la surface de log-vraisemblance traduisant ainsi un manque d'exploration globale. Le deuxième problème était que parfois l'essaim parvenait à identifier certaines positions intéressantes proches de la source à l'aide des valeurs de P^i, mais était tout de même attiré vers d'autres positions éloignées de la source du fait de l'accélération vers G, traduisant ainsi un manque d'exploration locale.

Pour pallier à ces deux problèmes, nous avons décidé de combiner l'algorithme PSO avec deux autres algorithmes. Le premier est un algorithme de type évolution différentielle qui permet à l'essaim d'avoir une meilleure exploration globale. Le deuxième algorithme est une méthode MCMC nommée Uphill Climber et appliquée seulement sur les positions P^i. Pour cela nous avons imposé à la chaîne d'avoir un critère d'acceptation où la nouvelle position dans l'espace des paramètres proposée est acceptée si et seulement si celle-ci possède une valeur de log-vraisemblance supérieure à celle actuelle. Le but de cet algorithme secondaire est alors de pallier le manque d'exploration locale en explorant les positions intéressantes autour des valeurs de P^i. Enfin, pour éviter que la chaîne reste coincée dans des maxima locaux, nous avons aussi décidé d'introduire une phase de recherche locale finale autour de la position G en ne considérant qu'un essaim de taille réduite. La forme de l'algorithme finale est alors une combinaison des trois algorithmes précédents, à savoir PSO, évolution différentielle et Uphill Climber, avec une phase finale locale de recherche.

Grâce à ces améliorations, notre algorithme a été capable d'avoir de bien meilleures performances qu'auparavant. Sur la Figure 8, nous présentons l'évolution de certains paramètres de l'essaim lors d'une recherche effectuée sur une source binaire de vérification pour LISA de type naine blanche - naine blanche nommée RXJ0806.3+1527. Dans la colonne du haut, nous présentons l'évolution de trois individus de l'essaim en fonction du nombre d'itérations pour la fréquence f_0, la colatitude θ, la longitude ϕ et le rapport signal sur bruit ρ. Dans la colonne du milieu, nous donnons l'évolution des valeurs de P^i associés à ces trois particules, tandis que dans la colonne du bas nous présentons l'évolution de G. On aperçoit sur cette figure que notre algorithme est capable de localiser rapidement la position de la source en termes de fréquence tandis qu'il faut plus de temps pour que celui-ci parvienne à localiser la position de la source dans le ciel. En parallèle, on observe une nette augmentation de ρ à l'itération 300 et 600, ce qui correspond au moment où l'on applique l'Uphill Climber. Enfin, entre les itérations 600 et 750, nous voyons que la phase finale de recherche locale parvient effectivement à améliorer ρ de 20 à 25, ce qui illustre que l'algorithme ne reste pas coincé dans les maxima locaux de la surface de log-vraisemblance.

À ce moment de l'étude, nous avons alors décidé d'appliquer l'algorithme sur deux ensembles de données comportant plusieurs sources et destinés à tester différentes capacités de l'algorithme. Le premier ensemble de données comportait ainsi 18 sources sur une large bande de fréquence de 1 mHz avec des rapports signal sur bruit entre 10.5 et 28. Dans ce cas, nous avons décidé de n'introduire aucune confusion entre les sources avec une distance minimale entre les sources de $273 f_m$. Dans le deuxième ensemble de données, nous avons voulu tester comment l'algorithme parvient à gérer des données où il existe une faible confusion entres les sources. Pour cela, l'ensemble de données comportait 30 sources sur un domaine de fréquence réduit de 30μHz avec une distance minimale entre les sources de $8 f_m$. Concernant la stratégie

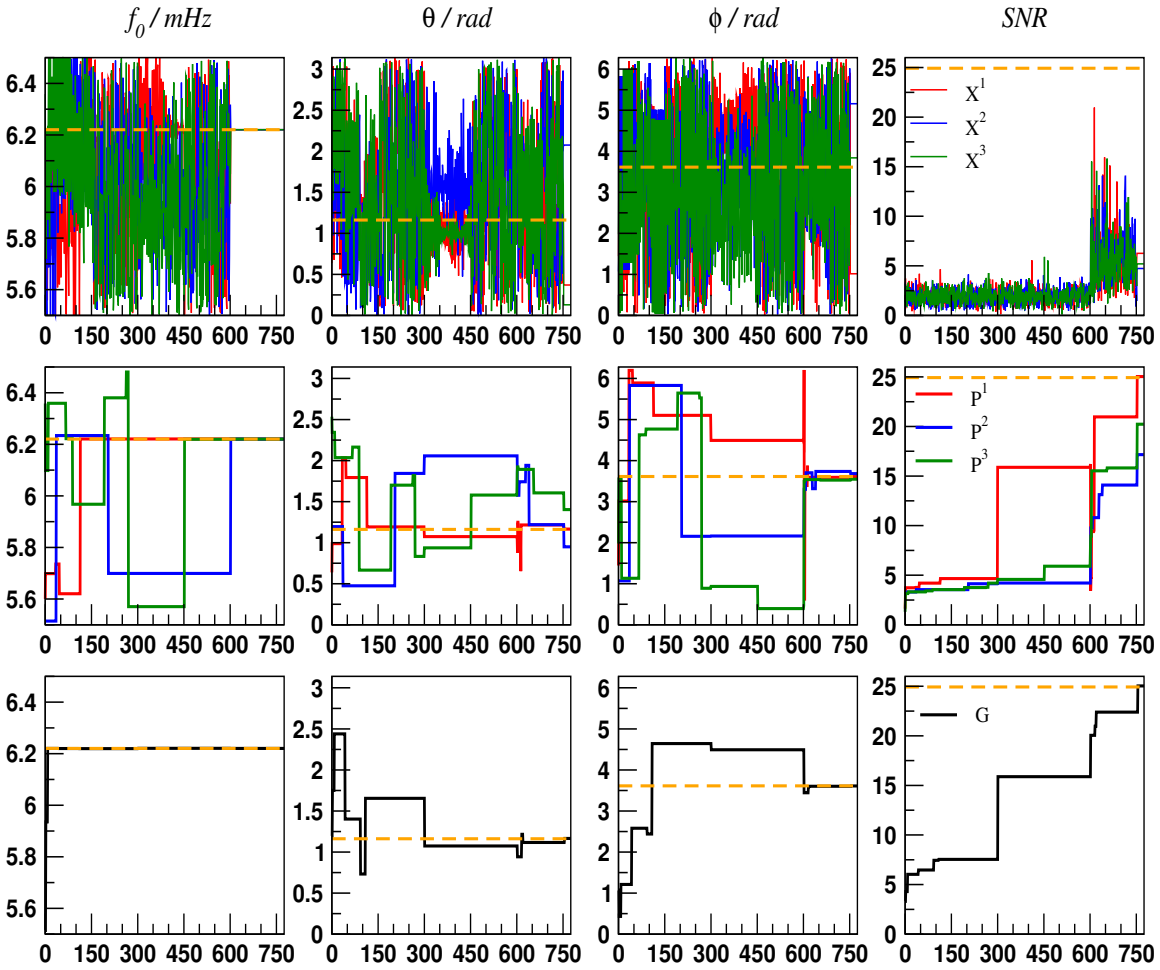

Figure 8: Graphique représentant l'évolution des paramètres (f_0, θ, ϕ) et du rapport signal sur bruit en fonction du nombre d'itérations de l'algorithme pour trois particules de l'essaim, X^i, avec leurs meilleures positions personnelles, P^i, et la meilleures position de l'essaim, G, pour la recherche de la source RXJ0806.3+1527. Dans chaque cellule du graphe, les valeurs réelles sont représentées en orange.

de recherche des sources, nous avons décidé d'adopter une recherche séquentielle où l'algorithme recherche une seule source à la fois. Dès qu'une source est identifiée, elle est ensuite soustraite au signal originel, et l'algorithme est relancé pour chercher la prochaine source.

Sur la figure (9), nous présentons les résultats que nous avons obtenus pour la recherche des sources du premier (gauche) et deuxième (droite) ensemble de données avec notre algorithme. On représente en rouge le spectre de puissance du signal original, en bleu clair le spectre de puissance du bruit instrumental et en bleu foncé le spectre de puissance du signal résiduel correspondant à la différence entre le signal original où l'on a retiré tous les signaux trouvés par l'algorithme et le bruit instrumental. On observe que pour tous les signaux des deux ensembles de données, le spectre de puissance du signal résiduel est en dessous du spectre instrumental, ce qui montre que l'algorithme est parvenu à identifier correctement toutes les sources présentes dans les données.

De plus, pour chaque source identifiée par l'algorithme, une étude d'estimation de paramètres a également été réalisé grce à un algorithme de type DEMC. Sur la figure 11, nous représentons dans la colonne du haut les valeurs des intervales de confiance à 99% identifiés pour l'amplitude A (gauche), l'inclinaison ι (milieu) et la fréquence f_0 (droite).On observe encore une fois que dans les deux scénarios, les vraies valeurs des paramètres de la source sont comprises dans l'intervalle de confiance à 99 %. Dans la colonne du bas, nous présentons la valeur de l'erreur dans le ciel (gauche), la distance orthodromique (milieu) ainsi que le rapport signal sur bruit originel en rouge et celui identifié par l'algorithme en bleu (droite). Encore une fois, on observe que l'algorithme parvient bien à retrouver les sources avec le même signal sur bruit, indiquant que la recherche a réussi. Il est intéressant de noter que nous parvenons à retrouver les sources par ordre de signal sur bruit bien que l'algorithme n'ait pas été conçu pour faire

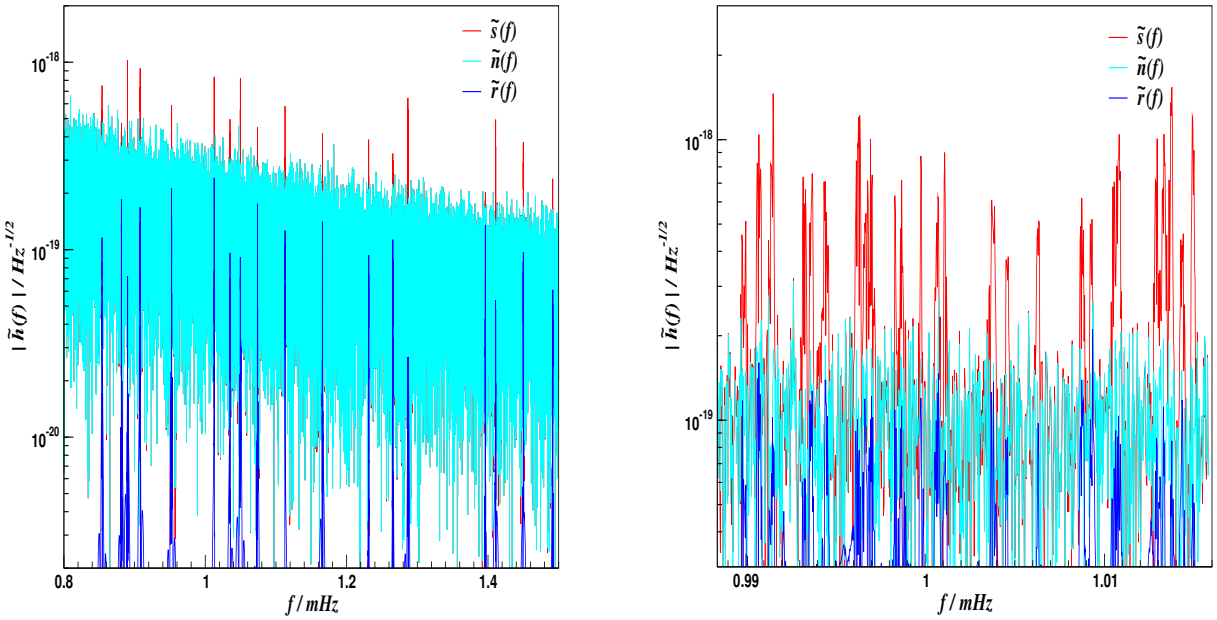

Figure 9: Spectre de puissance pour le signal mesuré, le bruit et le signal résiduelle pour le scénario 1(gauche) et le scénario 2 (droite)

cela.

L'algorithme dévelopé pendant ce travail de thèse a par conséquent été une réussite et a réussi à identifier des sources sur des intervalles de fréquence de 1 mHz, ce qui n'avait pas été fait à l'époque de ce travail. Cependant de nombreux travaux sont encore prévus pour le futur. Tout d'abord, il sera intéressant de voir comment l'algorithme parvient à gérer le problème de la haute confusion entre les sources, ce qui est une situation attendue pour le cas de LISA. De plus, nous aimerions également tester l'algorithme dans une situation proche de la réalité que sera l'analyse de données avec LISA, c'est à dire avec une population entière de compactes binaires galactiques. Ensuite, il sera intéressant de tester des modèles de formes d'ondes plus complexes faisant intervenir la dérivée de la fréquence.

En conclusion, ce travail de thèse a permis d'aborder des aspects variés de l'analyse de données pour les ondes gravitationnelles en considérant à la fois les détecteurs terrestres, LIGO/Virgo, et spatiale, LISA. De plus, il faut souligner que les algorithmes dévelopés dans chacun des cas sont flexibles et peuvent s'adapter à d'autres situations complexes d'analyse de données. Un travail intéressant serait par exemple d'appliquer l'algorithme HMC pour l'estimation de paramètres dans le cas des compact binaires galactiques avec LISA.

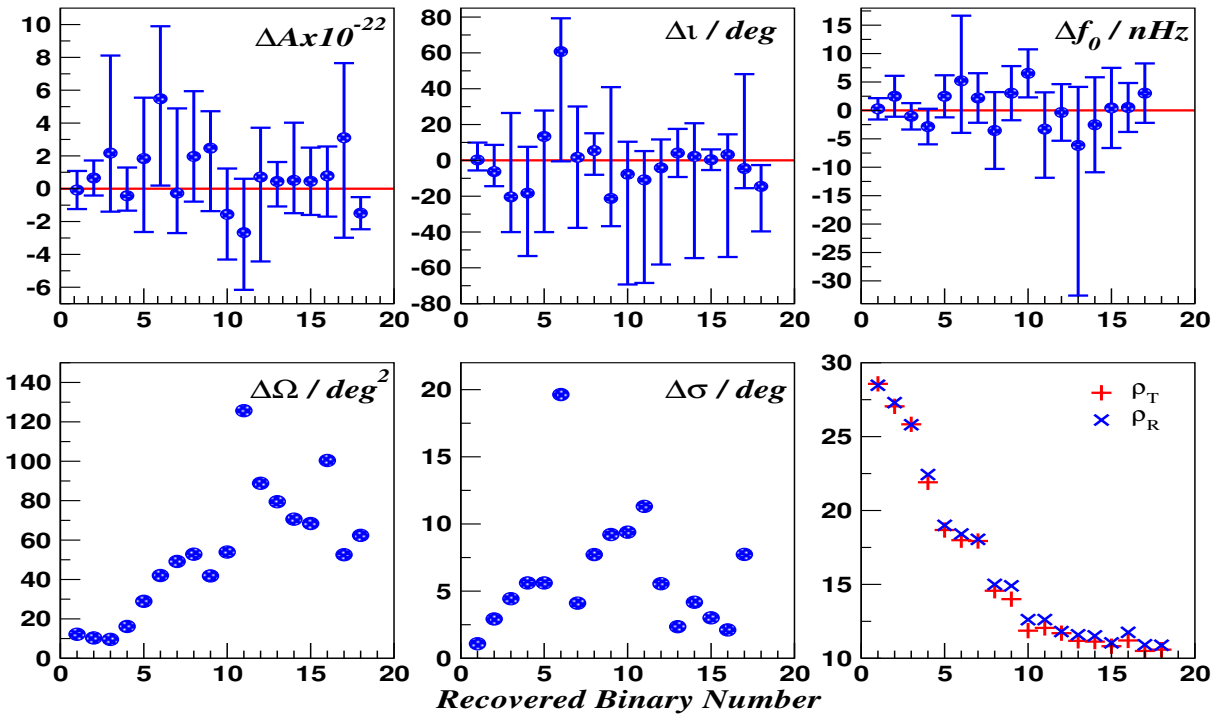

Figure 10: Résutats obtenus pour les sources retrouvées par l'algorithme pour le scénario 1. Dans la ligne du haut, on donne les intervalles de confiance à 99% ainsi que la valeur des médianes soustraites à la valeur réélles pour l'amplitude (gauche), l'inclinaison (milieu) et la fréquence (droite). Dans la ligne du bas, on donne les valeurs d'erreur pour les angles dans le ciel, la distance orthodromique et les valeurs réelles et trouvées par l'algorithme du rapport signal sur bruit.

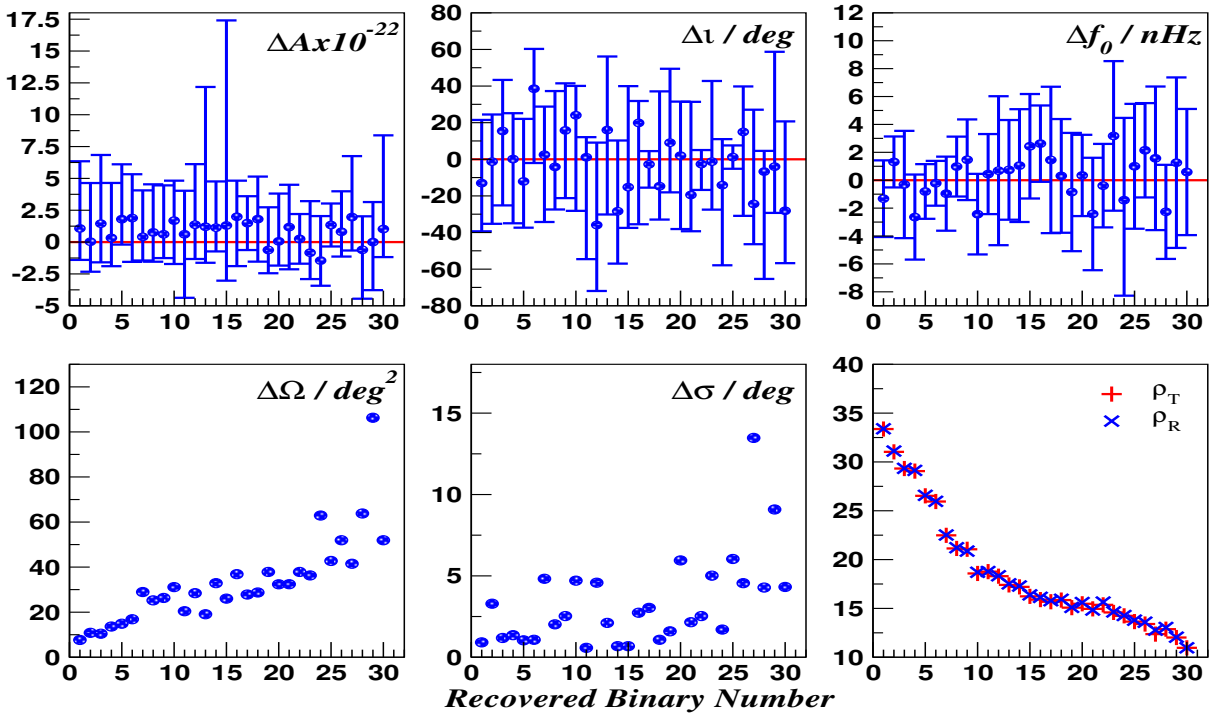

Figure 11: Résutats obtenus pour les sources retrouvées par l'algorithme pour le scénario 2. Dans la ligne du haut, on donne les intervalles de confiance à 99% ainsi que la valeur des médianes soustraites à la valeur réélles pour l'amplitude (gauche), l'inclinaison (milieu) et la fréquence (droite). Dans la ligne du bas, on donne les valeurs d'erreur pour les angles dans le ciel, la distance orthodromique et les valeurs réelles et trouvées par l'algorithme du rapport signal sur bruit.

Introduction

Gravitational waves are deformations of space-time propagating at the speed of light and resulting from movements of masses that are asymmetrically distributed. These waves were first predicted by Einstein in 1916 in the framework of the theory of general relativity [1, 2]. One of the main sources of gravitational waves are compact binaries that are composed of end products of stellar evolution, namely white dwarfs, neutron stars and black holes. As the two compact objects orbit around each other, they emit gravitational waves with frequency equal to twice the orbital frequency. During this process, the binary components lose energy through the emission of gravitational waves and the two compact objects inspiral towards each other. When their orbital separation is small, they start to rapidly approach each other and merge in a single compact object. A number of gravitational waveform models have been developed during the last decades to predict what form should take the gravitational waves emitted during coalescence. If we can measure gravitational waves, we can then confront these models with the data to understand and constrain the properties of such compact objects.

The first proof for the existence of the gravitational waves was from a study conducted by Hulse and Taylor, by measuring the energy loss of the binary pulsar system PSR B1913+16 [3, 4]. This study found that change of orbital period of the binary to be consistent with the prediction from general relativity, should the orbit decay from the emission of gravitational waves. While the evidence for gravitational waves from this study is strong, they were not directly detected but were inferred from indirect measurements. In fact, to directly detect gravitational waves, it is necessary to have high-precision experiments that are capable of measuring the very small deformation of space induced by a passing gravitational wave. In previous decades, a network of ground-based detectors based on laser interferometry have been designed and built to have enough sensitivity to be able to measure the typical deformation of the gravitational waves of the order of 10^{-21}. As of now, three detectors are currently in operation; Advanced Virgo in Cascina, Italy [5], and the two Advanced LIGO detectors in Hanford and Livingston, USA [6]. In the meantime, a number of detectors are currently being constructed or designed to be operational in the upcoming years. These will include a cryogenic underground detector called KAGRA in Japan [7] and another Advanced LIGO detector to be built in India. Plans are also underway for the study and development of third generation detectors such as the Einstein Telescope in Europe and Voyager/Cosmic Explorer in the USA. Finally, the space-based detector LISA should be launched in 2034 [8].

In September 2015, the two advanced LIGO detectors made the first direct coherent detection of gravitational waves emitted during the coalescence of two stellar mass black holes [9]. Two additional coalescences of stellar mass black holes were then detected during the first and second observation run of advanced LIGOs in December 2015 [10] and January 2017 [11]. In August 2017, the advanced Virgo interferometer joined the network of ground-based detectors which led to the first triple coincident detection of a binary black hole coalescence [12]. These detections are of extremely high importance and opened a new window to probe the astrophysics of compact objects. Through the correlation of theoretical models with the data, it was possible to constrain the parameters of the two compact objects such as the luminosity distance, the masses and the spins. With this information, it is then possible to put constraints on current astrophysical models and formation scenarios of such objects [13, 14]. In addition, while the detection of gravitational waves is a proof of validity for the theory of general relativity, their measurement can also be used to test the theory of general relativity and measure deviations from the theory if any. For now, none of the detected systems has shown any measurable deviation from general relativity [15, 12].

Analyzing gravitational wave data measured by a detector is a complex subject that requires advanced data analysis techniques. Two complementary aspects need to considered for data analysis, the search for the source and parameter estimation. Even though gravitational wave detectors are capable of reducing the noise a very low level, the gravitational wave signal can be buried in the noise of the detector. To make the signal emerge from the noise, one can use matched filtering techniques where the data are correlated

with theoretical gravitational wave templates, which are a function of the parameters of the astrophysical system, to build a detection statistics. Given the number of parameters involved and the complexity of the parameter space, it can prove to be quite difficult to search for the gravitational wave signals. As a consequence, various search techniques have been developed and are currently being designed in order to search optimally for gravitational wave signals. A part of this thesis work was dedicated to the development of a search algorithm using evolutionary algorithms for the search of monochromatic compact galactic binaries with eLISA.

When a gravitational signal is detected, the next stage of data analysis is to estimate the parameters of the sources as inferred from the signal measurement. This analysis step is crucial since it produces the final values that are used to constrain physical and astrophysical models. Studies have shown over the last few years, that the most accurate results results for parameter estimation are obtained when adopting a full Bayesian approach. In this framework, our knowledge of the signal parameters is expressed in terms of a probability density, or posterior distribution. Given the complexity and high dimension of the parameter space we need to deal with, one of the most efficient techniques to compute the posterior distribution is to use sampling algorithms based on Markov Chain Monte Carlo approach. However, the computational time required to run a full Bayesian analysis is often very high due to the complexity of the problem at hand. This can be problematic in the near future when the improvement of the detectors will allow to detect many sources of gravitational waves. As a consequence, it is necessary to develop sampling algorithms that are as efficient as possible to speed up the parameter estimation process. The main part of this thesis work was dedicated to the implementation of a Hamiltonian Monte Carlo algorithm for the parameter estimation of gravitational waves emitted by binary system composed of two neutron stars.

In Chapter 1, we briefly review the theory of gravitational waves as predicted by general relativity. We discuss the production of gravitational waves along with their interaction with matter. In Chapter 2, we outline how gravitational waves can be measured by detectors based on laser interferometry and what are the gravitational wave sources expected for these detectors. We present the current network of ground-based detectors and discuss aspects of the localisation of a source with this network of detectors. We then describe the future detectors that will be operational in the upcoming years along with the third generation of ground-based detectors that are currently being designed. Finally, we give details on the future space-based detector LISA.

In Chapter 3, we first review stellar evolution for a single star and describe the formation of compact objects at the end state of stellar evolution. Secondly, we describe the main properties of these white dwarfs, neutron stars and stellar mass black holes. Finally, we describe the stellar evolution of binary systems and see how it can lead to the formation of compact binaries, composed of mixtures of the above objects.

In Chapter 4, we present the main aspects of gravitational wave data analysis. First, we describe the method of matched filtering and how it is applied to gravitational wave data analysis. Secondly, we introduce the framework of Bayesian inference for parameter estimation. We review some aspects of probability and Markov Chains chains, and describe a number of Markov Chain Monte Carlo algorithms. Finally, we illustrate how we can assess the convergence of these algorithms with a simple example. In Chapter 5, we describe the gravitational waveform models we used during this work, including how we can derive the response of a network of ground-based detectors to an incoming gravitational wave.

In Chapter 6, we describe the introductory study we used for parameter estimation of binary neutron stars. In this study, we built a Differential Evolution Monte Carlo algorithm that we used as a reference to compare the results we obtained with the HMC algorithm developed during this thesis. We describe the main features of the Differential Evolution Monte Carlo algorithm, before presenting the results obtained with this algorithm on the set of ten sources .

In Chapter 7, we describe the Hamiltonian Monte Carlo algorithm that we implemented during this thesis. We present the algorithm in a general context and explain how this algorithm can be used in the Bayesian framework. We also highlight that the algorithm has a number of free parameters that need to be fine tuned to the problem at hand, and introduce one of the main problem that has prevented this algorithm from being used as a primary sampler, that is the computation cost associated with the evaluation of the gradient of the log-likelihood.

In Chapter 8, we present the main study conducted to implement the Hamiltonian Monte Carlo for the parameter estimation of gravitational waves. In this part of the study, we used a single binary neutron star source to benchmark the algorithm. We show how we derived optimal values for the free parameters of the algorithm so that we obtained the best performances for our algorithm. We then outline a major part of this research that was dedicated to solving the gradient bottleneck. We present all the methods we used, including the ones that failed, and give the solution we found for the problem. Finally, we give

the results obtained with this algorithm on the single source we used. We compare it with the result obtained with the Differential Evolution algorithm and highlight that our algorithm performed much better on all aspects of parameter estimation.

In Chapter 9, we describe the upgrades and modifications we made to the algorithm. We then present the final results we obtained and compare it with the Differential Evolution Monte Carlo algorithm. Once again, we found that the algorithm was much more efficient than the Differential Evolution Monte Carlo for all sources. Finally, we discuss the performances of this algorithm compared to the performances of the current algorithm used by the Ligo/Virgo collaboration and highlight that this algorithm could both speed-up and increase the performances of parameter estimation of binary neutron stars.

In Chapter 10, we present the part of this thesis work dedicated to the search for monochromatic compact galactic binaries with eLISA. In this study we built a search algorithm based on an a combination of evolutionary algorithms, Differential Evolution and Particle Swarm Optimisation. We describe our research process to benchmark and optimise the algorithm up to a point where we were able to detect a single source in a 1 mHz band of frequency. Finally, we present the performances of the search algorithm on two different sets of compact galactic binaries sources with no confusion and mild confusion, and highlight that the algorithm was able to recover all sources in both cases.

Chapter 1

General relativity and gravitational waves

In the framework of general relativity, gravitation is interpreted as the curvature of spacetime and thus originates from the geometry of spacetime itself that is described in terms of a spacetime metric $g_{\mu\nu}$. Einstein's field equations, or Einstein's equations, describe the connection between the curvature of spacetime and the mass and energy contained within it as,

$$G_{\mu\nu} = R_{\mu\nu} - \frac{1}{2}R = \frac{8\pi G}{c^4}T_{\mu\nu}, \tag{1.1}$$

where $G_{\mu\nu}$ is the Einstein tensor, $R_{\mu\nu}$ is the Ricci tensor, R is the Ricci scalar and $T_{\mu\nu}$ is the stress energy momentum tensor.

In the first section below, we will derive Einstein's equations when we consider their linearised form and see how they give rise to gravitational waves (GWs). In the second section, we will study the solutions of the linearised Einstein's equations in vacuum. Then, we will see how GWs are produced in the third section and finally investigate their interaction with matter.

1.1 From linearised Einstein equations to gravitational waves

We consider a spacetime metric $g_{\mu\nu}$ that is the combination of the flat Minkowski metric $\eta_{\mu\nu}$ with a metric perturbation $h_{\mu\nu}$,

$$g_{\mu\nu} = \eta_{\mu\nu} + h_{\mu\nu} + \mathcal{O}(h^2), \tag{1.2}$$

where the Minkowski metric is given by its canonical form $\eta_{\mu\nu} = \text{diag}(-1, 1, 1, 1)$ and $h_{\mu\nu}$ is a small perturbation such that $\mid h_{\mu\nu} \mid \ll 1$. Since the perturbation is small, we will compute all following quantities by neglecting higher than first orders for $h_{\mu\nu}$. By doing so, the indices for the tensors are raised and lowered using the Minkowski metric, meaning that the inverse perturbation matrix is,

$$h^{\mu\nu} = \eta^{\mu\sigma}\eta^{\nu\sigma}h_{\mu\nu}, \tag{1.3}$$

and the inverse metric is given by,

$$g^{\mu\nu} = h^{\mu\nu} - \eta^{\mu\nu}. \tag{1.4}$$

For the partial derivatives with respect to the space-time coordinate x^μ, we will use the notation

$$\partial_\mu = \frac{\partial}{\partial x^\mu}. \tag{1.5}$$

The Christoffel connection is given by,

$$\Gamma^\rho{}_{\mu\nu} = \frac{1}{2}g^{\rho\lambda}(\partial_\mu g_{\nu\lambda} + \partial_\nu g_{\lambda\mu} - \partial_\lambda g_{\mu\nu}), \tag{1.6}$$

$$= \frac{1}{2}\eta^{\rho\lambda}(\partial_\mu h_{\nu\lambda} + \partial_\nu h_{\lambda\mu} - \partial_\lambda h_{\mu\nu}) + \mathcal{O}(h^2). \tag{1.7}$$

The next quantity we are interested in is the Riemann tensor,

$$R^\mu{}_{\nu\rho\sigma} = \partial_\rho\Gamma^\mu{}_{\sigma\nu} - \partial_\sigma\Gamma^\mu{}_{\rho\nu} + \Gamma^\mu{}_{\rho\lambda}\Gamma^\lambda{}_{\sigma\nu} - \Gamma^\mu{}_{\sigma\lambda}\Gamma^\lambda{}_{\rho\nu}. \tag{1.8}$$

Since we neglect all terms with higher orders of h, we can neglect the product of Christoffel symbols in the previous equation and rewrite the expression of the Riemann tensor as,

$$R^{\mu}{}_{\nu\rho\sigma} = \frac{1}{2}(\partial_\rho\partial_\nu h^{\mu}{}_{\sigma} + \partial^\mu\partial_\sigma h_{\nu\rho} - \partial_\sigma\partial_\nu h^{\mu}{}_{\rho} - \partial^\mu\partial_\rho h_{\nu\sigma}) + \mathcal{O}(h^2). \tag{1.9}$$

The Ricci tensor is then given by the contraction of the Riemann tensor as,

$$R_{\mu\nu} = R^{\sigma}{}_{\mu\sigma\nu}, \tag{1.10}$$

$$= \frac{1}{2}(\partial^\sigma\partial_\mu h_{\sigma\nu} + \partial^\sigma\partial_\nu h_{\sigma\mu} - \partial_\mu\partial_\nu h - \Box h_{\mu\nu}) + \mathcal{O}(h^2), \tag{1.11}$$

where $\Box = \partial_\mu\partial^\mu$ is the d'Alembertian operator on a flat spacetime and $h = \eta^{\mu\nu}h_{\mu\nu} = h^{\mu}{}_{\mu}$ is the trace of $h_{\mu\nu}$. The Ricci scalar is obtained by contracting the Ricci tensor,

$$R = R^{\mu}{}_{\mu}, \tag{1.12}$$

$$= \partial^\sigma\partial^\mu h_{\sigma\mu} - \Box h + \mathcal{O}(h^2). \tag{1.13}$$

Finally, we compute the Einstein tensor at first order in $h_{\mu\nu}$ as,

$$G_{\mu\nu} = R_{\mu\nu} - \frac{1}{2}R, \tag{1.14}$$

$$= \frac{1}{2}(\partial^\sigma\partial_\mu h_{\sigma\nu} + \partial^\sigma\partial_\nu h_{\sigma\mu} - \partial_\mu\partial_\nu h - \Box h_{\mu\nu} - \eta_{\mu\nu}\partial^\sigma\partial^\rho h_{\sigma\rho} + \eta_{\mu\nu}\Box h). \tag{1.15}$$

The expression for the Einstein tensor can be simplified by introducing the trace-reversed perturbation metric $\overline{h}_{\mu\nu}$ as

$$\overline{h}_{\mu\nu} = h_{\mu\nu} - \frac{1}{2}\eta_{\mu\nu}h, \tag{1.16}$$

which allows us to rewrite the Einstein tensor as,

$$G_{\mu\nu} = \frac{1}{2}\left[\partial^\rho\partial_\nu\overline{h}_{\mu\rho} + \partial^\rho\partial_\mu\overline{h}_{\nu\rho} - \Box\overline{h}_{\mu\nu} - \eta_{\mu\nu}\delta^\rho\delta^\sigma\overline{h}_{\rho\sigma}\right]. \tag{1.17}$$

This expression can be further simplified by taking advantage of gauge freedom. Let us consider the infinitesimal coordinate transformation,

$$x'^{\mu'} = x^\mu + \xi^\mu, \tag{1.18}$$

where we take $\mid \partial_\mu\xi_\nu \mid = \mathcal{O}(h)$. Since the coordinate transformation of a metric $g_{\mu\nu}$ is given by

$$g'_{\mu\nu} = g_{\alpha\beta}\frac{\partial x^\alpha}{\partial x'^\mu}\frac{\partial x^\beta}{\partial x'^\nu}, \tag{1.19}$$

we can then express the transformation for the metric perturbation under the infinitesimal coordinate transformation of Eq. (1.18) as,

$$h'_{\mu\nu}(x') = h_{\mu\nu}(x) - (\partial_\mu\xi_\nu + \partial_\nu\xi_\mu). \tag{1.20}$$

In terms of the trace reversed metric perturbation, this becomes

$$\overline{h}'_{\mu\nu}(x') = \overline{h}_{\mu\nu}(x) - (\partial_\mu\xi_\nu + \partial_\nu\xi_\mu - \eta_{\mu\nu}\delta_\rho\xi^\rho), \tag{1.21}$$

Now if we look at the perturbation for the Riemann tensor defined in Eq. (1.9) by introducing this coordinate transformation, $\delta R^{\mu}{}_{\nu\rho\sigma}$, we find,

$$\begin{aligned}\delta R^{\mu}{}_{\nu\rho\sigma} &= \frac{1}{2}[\partial_\rho\partial_\nu\partial^\mu\xi_\sigma + \partial_\rho\partial_\nu\partial^\sigma\xi_\mu + \partial_\sigma\partial^\mu\partial_\nu\xi_\rho + \partial_\sigma\partial^\mu\partial_\rho\xi_\nu \\ &\quad -\partial_\rho\partial_\nu\partial^\mu\xi_\sigma - \partial_\rho\partial_\nu\partial^\sigma\xi_\mu - \partial_\sigma\partial^\mu\partial_\nu\xi_\rho - \partial_\sigma\partial^\mu\partial_\rho\xi_\nu], \qquad (1.22)\\ &= 0. \qquad (1.23)\end{aligned}$$

That means that the curvature of spacetime is left unchanged under our coordinate transformation ξ^μ, and that the laws of physics are also left unchanged when moving to the new coordinate system. We can then fix a gauge, or in other words, fix a coordinate transformation as in Eq. (1.18), such that the

Einstein's tensor in Eq. (1.17) has a simpler form. The gauge that we choose is the so-called de Donder gauge, or Lorentz gauge, given by

$$\partial^\nu \overline{h}_{\mu\nu} = 0. \tag{1.24}$$

In terms of the coordinate transformation, the de Donder gauge can be achieved by considering the transformation of $\partial^\nu \overline{h}_{\mu\nu}$ under ξ^μ,

$$(\partial^\nu \overline{h}_{\mu\nu})' = \partial^\nu \overline{h}_{\mu\nu} - \Box\xi_\mu, \tag{1.25}$$

and set the infinitesimal coordinate transformation such that,

$$\Box\xi_\mu = \partial^\nu \overline{h}_{\mu\nu}. \tag{1.26}$$

In this new coordinate system, the traceless perturbation metric satisfies the de Donder gauge. Note that we still have some gauge freedom left, because we can always make another coordinate transformation $\overline{\xi}^\mu$ such that,

$$\Box\overline{\xi}^\mu = 0, \tag{1.27}$$

hence still satisfying the de Donder gauge. As we will see, this can be used later on to constrain the expression for the perturbation metric. Applying the de Donder gauge, we find that the Einstein tensor then reduces to the condensed form,

$$G_{\mu\nu} = -\frac{1}{2}\Box\overline{h}_{\mu\nu}, \tag{1.28}$$

and Einstein's equations are now expressed as,

$$\Box\overline{h}_{\mu\nu} = -\frac{16\pi G}{c^4} T_{\mu\nu}. \tag{1.29}$$

We see that the metric perturbation satisfies a wave equation with a source term given in terms of $T_{\mu\nu}$. This metric perturbation is referred to as gravitational waves. Note that if we take the derivative of the previous equation and use the de Donder gauge, we find that we have the following conservation law for $T_{\mu\nu}$,

$$\partial^\nu T_{\mu\nu} = 0, \tag{1.30}$$

which takes the same form as a conservation law on flat spacetime.

1.2 Vacuum solutions

In this section, we will study a specific case of solutions for Eq. (1.29), where we consider that we are in vacuum, i.e. $T_{\mu\nu} = 0$. In this case, the linearised Einstein's equations are written as,

$$\Box\overline{h}_{\mu\nu} = 0. \tag{1.31}$$

We observe that this equation is a source-free wave equation for $\overline{h}_{\mu\nu}$, where the waves are travelling at the speed of light c. Let us write a solution for $\overline{h}_{\mu\nu}$ as a standard plane wave,

$$\overline{h}_{\mu\nu} = C_{\mu\nu} e^{ik_\rho x^\rho}, \tag{1.32}$$

where $C_{\mu\nu}$ is a symmetric tensor representing the amplitude of the wave and $k_\rho = (\omega, k_1, k_2, k_3)$ is the wave vector, where ω is the angular frequency of the GW and k_1, k_2 and k_3 are the spatial components of the wave vector. If we plug this solution into Einstein's equations Eq. (1.31), we find that the wave vector needs to satisfy the following wave dispersion constraint,

$$k_\mu k^\mu = 0. \tag{1.33}$$

Let us see now how we can apply the various gauges we introduced in the previous section in order to constrain the form of the plane wave solution even more. First, the application of the de Donder gauge in Eq. (1.24) gives the following relationship between the amplitude tensor and the wave vector,

$$k_\mu C^{\mu\nu} = 0. \tag{1.34}$$

From this equation, we see that the perturbation amplitude $C^{\mu\nu}$ is orthogonal to the direction of propagation of the wave given by k_μ. Now we still have the gauge freedom left for a coordinate transformation

ξ^μ as given in Eq. (1.27). This coordinate transformation also satisfies a wave equation and can be expressed as,

$$\xi^\mu = B^\mu e^{ik_\nu x^\nu}, \tag{1.35}$$

where the wave vector is the same as for $\overline{h}_{\mu\nu}$ and B^μ is the amplitude vector. In this new gauge, we have the following transformation for the coordinate of the GW amplitude,

$$C'^{\mu\nu} = C^{\mu\nu} - ik_\mu B_\nu - ik_\nu B_\mu + i\eta_{\mu\nu}k_\rho B^\rho. \tag{1.36}$$

By choosing an appropriate set of components for B^μ, it is possible to set the components of $C^{\mu\nu}$ in the new gauge to satisfy the following constraints

$$C^\mu = 0, \tag{1.37}$$
$$C_{0\mu} = 0. \tag{1.38}$$

The calculation detailing the form that B^μ should take so that the amplitude tensor satisfies the previous conditions are detailed in [16]. Without loss of generality, we can take the case where a GW travels along the x^3 direction with wave vector $k^\mu = (\omega/c, 0, 0, \omega/c)$. In this case, the tensor amplitude for the GW solution is given by,

$$C_{\mu\nu} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & C_{11} & C_{12} & 0 \\ 0 & C_{12} & -C_{11} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \tag{1.39}$$

where C_{11} and C_{12} are two independent coefficients. The GW is then fully constrained by the coefficients C_{11}, C_{12}, and the frequency ω. This specific expression of the amplitude $C_{\mu\nu}$ or equivalently gravitational field $\overline{h}_{\mu\nu}$, is said to be in the transverse traceless gauge, or TT gauge. Note that, in the transverse traceless gauge, the expressions of $h_{\mu\nu}$ and $\overline{h}_{\mu\nu}$ are the same,

$$\overline{h}^{TT}_{\mu\nu} = h^{TT}_{\mu\nu}. \tag{1.40}$$

1.3 Generation of gravitational waves

So far we have studied the propagation of a linear perturbation of the metric in spacetime. We now focus on the mechanism that creates GWs. This means that we now look at the linearised Einstein's equations with a non-zero source term in the right hand side,

$$\Box\overline{h}^{\mu\nu} = \frac{8\pi G}{c^4}T_{\mu\nu}. \tag{1.41}$$

In order to solve these equations, we will use the same Green's function formalism set up to solve Maxwell's equations for electromagnetic waves. In our case, we introduce the Green function associated with the d'Alembertian operator which is the solution of the wave equation with a point particle in 4D spacetime,

$$\Box G(x^\mu - y^\mu) = \delta^{(4)}(x^\mu - y^\mu). \tag{1.42}$$

Given the linearity of Eq. (1.41), we can write the solution of the linearised Einstein's equation as

$$\overline{h}_{\mu\nu}(x^\sigma) = -\frac{16\pi G}{c^4}\int G(x^\sigma - y^\sigma)T_{\mu\nu}(y^\sigma)\mathrm{d}^4y. \tag{1.43}$$

All that is left to do now is to find an expression for the Green function. For the d'Alembertian operator, the Green functions are well known and are given by,

$$G(x^\sigma - y^\sigma) = -\frac{1}{4\pi|\mathbf{x}-\mathbf{y}|}\delta[|\mathbf{x}-\mathbf{y}| - (x^0 - y^0)]\theta(x^0 - y^0), \tag{1.44}$$

where $\mathbf{x} = (x^1, x^2, x^3)$ and $\mathbf{y} = (y^1, y^2, y^3)$ refer to spatial component of vectors in the isosurfaces of constant x^0 and y^0, $|\mathbf{x}-\mathbf{y}| = \left[\delta_{ij}(x^i - y^i)(x^j - y^j)\right]^{1/2}$ and $\theta(x^0 - y^0)$ is the Heaviside function that is equal to 1 when $x^0 > y^0$ and 0 otherwise. If we put this expression in Eq. (1.43), we find that

$$\overline{h}_{\mu\nu}(t, \mathbf{x}) = \frac{4G}{c^4}\int \frac{1}{|\mathbf{x}-\mathbf{y}|}T_{\mu\nu}(t_R, \mathbf{y})\mathrm{d}^3y, \tag{1.45}$$

where t_R is the retarded time given by $t_R = t - (1/c)|\mathbf{x} - \mathbf{y}|$.

Now let us consider a specific case where the matter source is an isolated source at distance R from an observer with a spatial extension δR small compared to R. Under those assumptions, one can derive a meaningful expression for the gravitational field created by this source as,

$$\overline{h}_{ij}(t, \mathbf{x}) = \frac{2G}{3Rc^4} \frac{\mathrm{d}^2 Q_{ij}}{\mathrm{d}t^2}(t_r), \tag{1.46}$$

where Q_{ij} is the quadrupole moment and is defined as

$$Q_{ij}(t) = 3 \int y^i y^j T^{00}(t, \mathbf{y}) \mathrm{d}^3 y. \tag{1.47}$$

We then see that the gravitational field is created when the second moment (or quadrupole) of the energy density, T^{00} is varying in time. Note that this is fundamentally different from electromagnetism where electric and magnetic fields are influenced by the evolution of dipoles.

For this thesis, we will be interested in the form for $\overline{h}_{ij}(t, \mathbf{x})$ in the specific case where we consider a binary system as illustrated in Figure 1.1. We assume here that the two objects have the same mass M and that their motion is in the plane (x^1, x^2) and can be described by Newtonian's dynamics. If we take r to be the distance of the objects with respect to their common center of mass and v the transverse velocity of the components, Newton's equations give use the relationship,

$$\frac{GM^2}{(2r)^2} = \frac{Mv^2}{r}, \tag{1.48}$$

allowing us to express the transverse velocity as,

$$v = \left(\frac{GM}{4r} \right)^{1/2}. \tag{1.49}$$

The orbital period for each component is then given by

$$T = \frac{2\pi r}{v}, \tag{1.50}$$

and the orbital angular frequency by

$$\omega = \frac{2\pi}{T} = \frac{GM}{4r^3}^{1/2}. \tag{1.51}$$

We can now write the circular motion for the two objects in terms of their respective coordinates (x_a^1, x_a^2) and (x_b^1, x_b^2) as,

$$\begin{aligned} x_a^1 &= r\cos(\omega t), & (1.52)\\ x_a^2 &= r\sin(\omega t), & (1.53)\\ x_b^1 &= -r\cos(\omega t), & (1.54)\\ x_b^2 &= -r\sin(\omega t). & (1.55) \end{aligned}$$

The energy density of the system $T^{00}(t, \mathbf{x})$ is then expressed as,

$$\begin{aligned} T^{00}(t, \mathbf{x}) &= \delta(x^3) \left[\delta(x^1 - x_a^1)\delta(x^2 - x_a^2) + \delta(x^1 - x_b^1)\delta(x^2 - x_b^2) \right], & (1.56)\\ &= \delta(x^3) \left[\delta(x^1 - r\cos(\omega t))\delta(x^2 - r\sin(\omega t)) + \delta(x^1 + r\cos(\omega t))\delta(x^2 + r\sin(\omega t)) \right]. & (1.57) \end{aligned}$$

Using the expression for the energy density, we can now express the quadrupole moment of the system using Eq. (1.47) as,

$$\begin{aligned} Q_{11} &= 6Mr^2\cos^2(\omega t) = 3Mr^2(1 + \cos(2\omega t)), & (1.58)\\ Q_{22} &= 6Mr^2\cos^2(\omega t) = 3Mr^2(1 - \cos(2\omega t)), & (1.59)\\ Q_{12} &= q_{21} = 6Mr^2\cos(\omega t)\sin(\omega t) = 3Mr^2\sin(2\omega t), & (1.60)\\ Q_{i3} &= q_{3i} = 0 \text{ for } i = 1, 2, 3. & (1.61) \end{aligned}$$

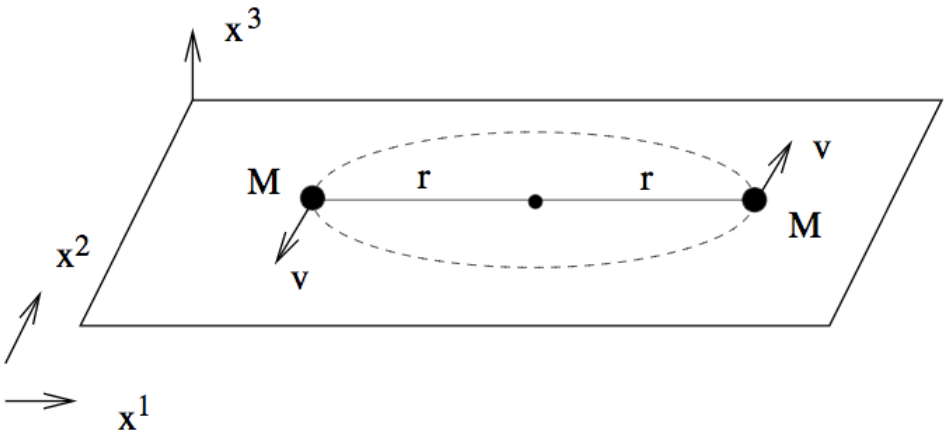

Figure 1.1: Illustration of the problem where we have two stars of mass M orbiting in the plane (x^1, x^2) around their center of mass with transverse velocity v and orbital radius r [16].

Now if we consider that the binary system is at a distance R from the observer, the resulting spatial components for the metric perturbation $h_{\mu\nu}$ in the transverse traceless gauge are given by

$$\overline{h}_{ij}(t, \mathbf{x}) = \frac{8GM}{Rc^4}\omega^2 r^2 \begin{pmatrix} -\cos(2\omega t_r) & -\sin(2\omega t_r) & 0 \\ -\sin(2\omega t_r) & \cos(2\omega t_r) & 0 \\ 0 & 0 & 0 \end{pmatrix}. \tag{1.62}$$

We see that the frequency of the gravitational wave is equal to twice the orbital frequency of the binary. Moreover, if we rewrite these expressions in the detector frame, the spatial components of the metric perturbation are given in terms of the two gravitational wave polarisations,

$$h^+(t) = -\overline{h}_{11}(t) = \overline{h}_{22}(t), \tag{1.63}$$
$$h^\times(t) = \overline{h}_{12}(t) = \overline{h}_{21}(t). \tag{1.64}$$

1.4 Interaction of gravitational waves with matter

So far we have derived the expression for the propagation of gravitational waves in vacuum and their emission by matter sources. Now we would like to see what is the effect of the gravitational wave induced on matter. In the framework of general relativity, the equation of motion for a free-falling test mass is given by the geodesic equation,

$$\frac{\mathrm{d}^2 x^\mu}{\mathrm{d}\tau^2} + \Gamma^\mu{}_{\nu\rho}\frac{\mathrm{d}x^\nu}{\mathrm{d}\tau}\frac{\mathrm{d}x^\rho}{\mathrm{d}\tau} = 0, \tag{1.65}$$

where x^μ is the coordinate vector of the test mass and τ its proper time. We highlight here that this equation of motion is only valid for free-falling test masses. If the test mass is originally not moving with respect to the test mass frame, and if we apply the transverse traceless gauge to our set of coordinates, we find that

$$\frac{\mathrm{d}^2 x^i}{\mathrm{d}\tau^2} = 0. \tag{1.66}$$

This means that the spatial coordinates of the test mass in free-fall are not affected by the gravitational waves in the transverse traceless gauge. In other words, the coordinates of the test mass in the gauge we chose are moving with the gravitational waves. However, as we will see, the gravitational waves do have an effect on distance separation.

Now let us consider a ring of test particles in free fall lying on the (x, y) plane as shown in Figure 1.2 with a GW arriving in the direction z. If we write the separation vector between two particles S^μ, we can write the evolution of this separation vector with the geodesic equation as [16],

$$\frac{\partial^2}{\partial t^2}S^\mu = \frac{1}{2}S^\sigma\frac{\partial^2}{\partial t^2}h^\mu{}_\sigma. \tag{1.67}$$

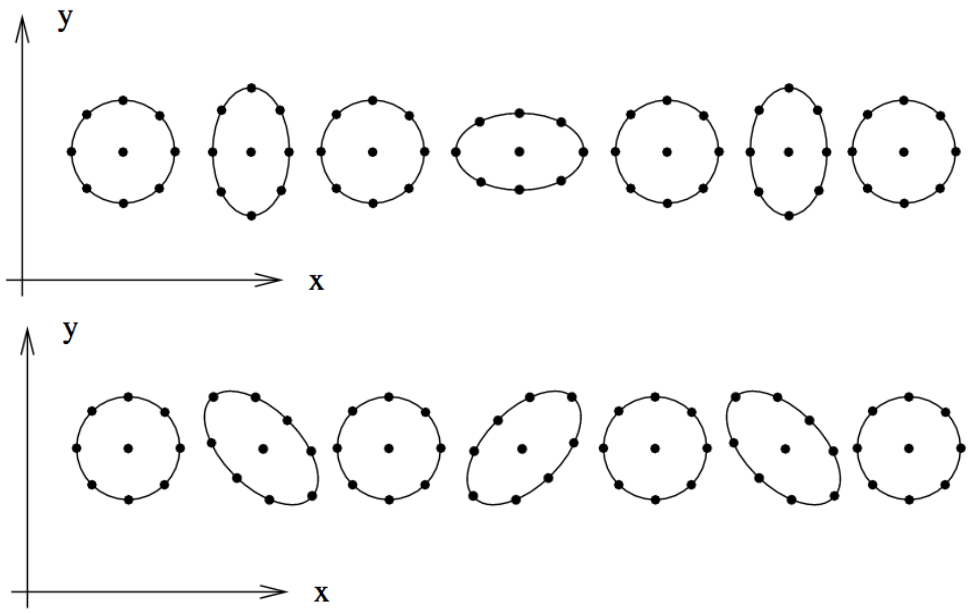

Figure 1.2: Illustration of the deformation induced on a ring of test particles in free-fall lying in the (x, y) plane by the gravitational waves polarisations h^+ (top) and $h^\times$ (bottom), for a wave travelling along the orthogonal z direction [16].

Since the GW travels in the z direction, the last equation implies that the separation vector S^μ will only be affected in the x and y direction, which corresponds to S^1 and S^2. First, we can derive what is the effect induced by h^+ only, by setting the other polarisation $h^\times = 0$. In this case, if we inject the solutions found in Section 1.2, the previous equation is written as,

$$\frac{\partial^2}{\partial t^2} S^1 = \frac{1}{2} S^1 \frac{\partial^2}{\partial t^2} h^+ \mathrm{e}^{ik_\rho x^\rho}, \tag{1.68}$$

$$\frac{\partial^2}{\partial t^2} S^2 = \frac{1}{2} S^2 \frac{\partial^2}{\partial t^2} h^+ \mathrm{e}^{ik_\rho x^\rho}. \tag{1.69}$$

which yields at lower order,

$$S^1 = \left(1 + \frac{1}{2} h^+ \mathrm{e}^{ik_\rho x^\rho}\right) S^1(0), \tag{1.70}$$

$$S^2 = \left(1 - \frac{1}{2} h^+ \mathrm{e}^{ik_\rho x^\rho}\right) S^2(0). \tag{1.71}$$

where $S^1(0)$ and $S^2(0)$ are the initial separation at initial time in the x and y direction. From these equations, we see that two particles with an initial deformations $S^1(0)$ in the x direction, will be deformed by the polarisation h^+ of a GW such that they oscillate in the x direction. And we have the same pattern for the y direction. Putting the two deformations together, we see that a ring of particles will be deformed in the form of a + as shown in the top panel of Figure 1.2.

If we now consider only the $h^\times$ polarisations of the GW, the equations for the deformation then becomes

$$S^1 = S^1(0) + \frac{1}{2} h^\times \mathrm{e}^{ik_\rho x^\rho} S^2(0), \tag{1.72}$$

$$S^2 = S^2(0) + \frac{1}{2} h^\times \mathrm{e}^{ik_\rho x^\rho} S^1(0). \tag{1.73}$$

In this case, the deformation induced on the ring of test particles take the form of a $\times$ as presented in the bottom panel of Figure 1.2.

Chapter 2

Gravitational wave detection

In the previous chapter, we have seen that a gravitational wave interacts with matter through proper length variation between two particles in free-fall. Each polarisation of the gravitational wave, $h^{+}(t)$ and $h^{\times}(t)$, induces a typical deformation pattern on a ring of free-fall particles as shown in Figure 1.2. However, the level of deformation or strain is extremely weak because the rigidity of space-time is very high. The challenge of gravitational wave astronomy is then to build an instrument that is capable of measuring these very small length variations.

Laser interferometry is based on the principle that two laser beams traveling along two orthogonal arms can track down precise variations in the length of the arms through phase difference of the two laser beams when they are combinated at the end. Thus, if we manage to build a laser interferometer that essentially mimics the ring of free-fall particles described before, then variations in lengths can be translated as variations of phases in laser beams. The challenge of measuring very small length variations is then translated into having technologies capable of measuring very small differences in phases of the laser beams. This is the reason why laser interferometry has been selected as a favored option for the detection of gravitational waves.

In this chapter, we present the current laser interferometers used for the detection of gravitational waves. We present briefly some of the key aspects that made possible to achieve a sensitivity in phase differences required for gravitational waves detection. We also present the future detectors that will be operational in the upcoming years.

2.1 How are gravitational waves detected ?

Let us consider an introductory example where two test masses are separated by a coordinate separation L_c on the axis y and a gravitational wave is going in the z-direction as shown in Figure 2.1, the proper separation is expressed in the TT gauge as

$$L_p = \int_0^{L_c} \sqrt{g_{yy}} \mathrm{d}y. \tag{2.1}$$

Note that the bounds of the integral are fixed because the coordinate are not perturbed by the gravitational waves in TT gauge as shown earlier. Working through the calculation, one can find that the proper distance in the plane $z = 0$ is given by,

$$L_p = L_c[1 + \frac{1}{2} h_{yy}^{TT}(t, z = 0)]. \tag{2.2}$$

We have derived the proper separation in the TT gauge, but since it is a coordinate invariant quantity, we know that it will be the same in any coordinate system. Thus, we see that the proper distance between two test masses is influenced by the gravitational wave and the relative variation of proper distance is

$$\frac{\delta L_p}{L_p} = \frac{1}{2} h_{yy}^{TT}(t, z = 0). \tag{2.3}$$

A Michelson interferometer measures the phase difference between two light beams that travel along two orthogonal arms with length L. At the ends of each arm, there are mirrors referred to as test masses, that reflect the laser beams and are taken as references for distance computation. These mirrors are

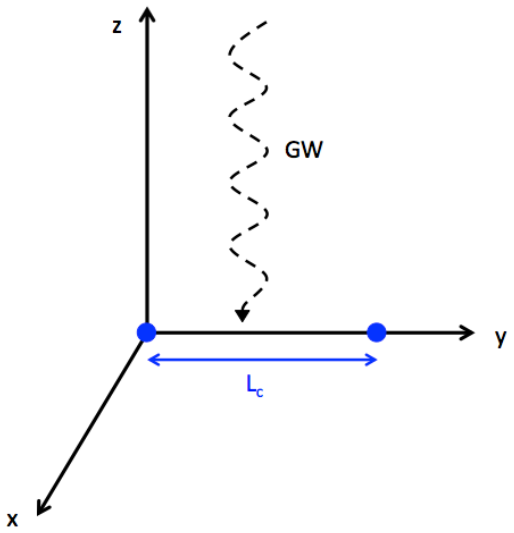

Figure 2.1: Illustration for the case where we consider two free-falling test masses in blue separated by proper distance L_c and a gravitational wave travelling along the z-axis. In this case, the relative variation of proper distance is given by Eq. (2.3).

put in such conditions that they are as close as possible to being in free-fall. The beams are finally recombined at the end of their path and the phase difference is measured using photodetectors. When a gravitational wave passes through the interferometer, the proper distance between the end mirrors of the arms in free-fall will vary as given by Eq. (2.3) and induce a difference in proper length between the two arms $\Delta L(t)$ as,

$$\Delta L(t) = \delta L_x - \delta L_y = h(t)L \tag{2.4}$$

where δL_x and δL_y represent the arm length variation between the test masses for each arm of the interferometer and $h(t)$ is the gravitational wave strain amplitude formed of the linear combination of the two gravitational wave polarisations $h^+(t)$ and $h^\times(t)$ projected onto the geometry of the detector. The quantity $\Delta L(t)$ is measured by deduction of the phase difference observed at the end path of the laser beams.

The strain induced on the detector arms by the passing gravitational wave depends both on the gravitational wave polarisation angle ψ and the incident angle from which it arrives at the detector. By projecting the gravitational wave signal from the source frame of reference where it is emitted to the detector frame of reference, we can write the detector response in terms of beam pattern functions, $F_+(\alpha, \delta, \psi)$ and $F_\times(\alpha, \delta, \psi)$, where α and δ represent the right ascension and declination of the source. The beam pattern functions tells us how sensitive the detector is to the gravitational wave polarisations $h^+(t)$ and $h^\times(t)$. The total gravitational wave strain measured by the detector in Eq. (2.4) is then expressed as,

$$h(t) = F_+(\alpha, \delta, \psi)h^+(t) + F_\times(\alpha, \delta, \psi)h^\times(t). \tag{2.5}$$

The typical amplitude of a gravitational wave emitted by the coalescence of two compact objects is extremely small, $h \sim 10^{-21}$ [9]. In order to reach this sensitivity level, extensions are added to the basic optical design of a Michelson interferometers [9, 17]. We present here only a small collection of these improvements but the reader can find more information in [5, 6]. First of all, the initial laser output is increased by a power recyling mirror that builds up resonance in the interferometer [18]. In addition to that, a resonant optical cavity formed by the two test masses in each arm of the interferometer increases the laser power even more. In the case of Advanced LIGO (aLIGO), the initial laser power input of 20 W is increased to a total of 100 kW in the interferometer arms. The other main advantage of the resonant optical cavity in the interferometer arms is to increase the optical path of the laser beams resulting in a factor of 300 improvement in phase measurements [19]. Finally, at the output of the interferometer, the bandwidth of the arm cavities is broadened thanks to a partially transmissive signal recycling mirror before the photodetector [20]. All these upgrades represented in Figure 2.2 make it possible to reach a level of measurement precision for the mirror displacements of the order of 10^{-18} m required for gravitational wave detection.

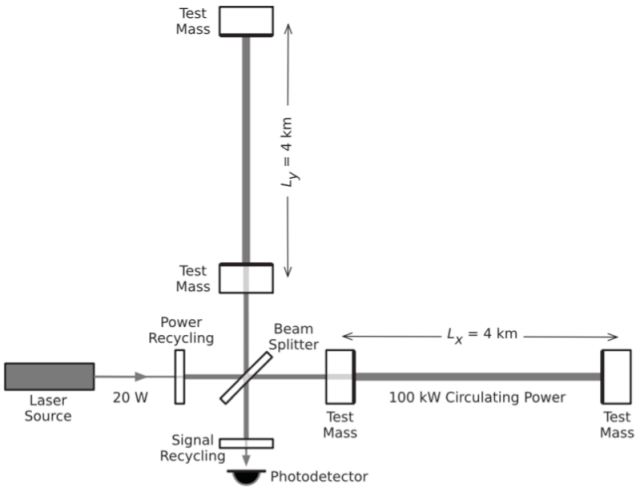

Figure 2.2: Simplified diagram of the LIGO detector based on a modified Michelson interferometer [9]

Despite all these efforts, there are still noise sources that remain and parasite the signal measured by the detector. These sources of noises have distinct effects and occur in various parts of the frequency band, that stands between ten and a few thousand Hz for current ground-based detectors. For high frequencies, the main source of noise comes from quantum fluctuations in the number of emitted laser photons and is referred as photon shot noise. The interferometry techniques described before are designed to reduce this noise, but it still remains a limiting source of noise in the high frequency regime of the detector. In the intermediate band, thermal fluctuations in the mirror and their suspensions result in thermal noise that dominates in this frequency band of frequency. To mitigate thermal noise, both the mirrors and suspensions are made of low-mechanical-loss materials, fused silica substrates for the mirrors and fused silica fibers for the suspensions. For low frequencies, the main source of noise are seismic vibrations of the Earth that couple with the test masses. Advanced suspension attenuation systems are designed to reduce the level of these vibrations. Finally for the very low frequency regime, local variations of gravitational potential originating from moving objects close to the detector, pressure waves or thermal fluctuations in the atmosphere, introduce an unwanted gravitational attraction on the test masses. These noise sources are referred to as Newtonian noise and will become important especially for third generation detectors [21]. If we put everything together, we can define an operating band of frequencies in which the detector is sensitive, and characterise the noise level in this band with the one-sided noise power spectral density $S_n(f)$. In the next section, we present the current and future detectors and their sensitivity.

2.2 Current detectors

The current operating gravitational wave detectors are the three ground-based advance LIGO and Virgo detectors. The LIGO detectors consist of two twin detectors located in Livingston, Louisiana and Hanford, Washington, with arm lengths $L = 4$km [6]. After an upgrading break for system improvements, the detectors are now operating in their advanced version aLIGO and started taking science data in September 2015. The Virgo detector is a European gravitational wave detector located in Cascina near Pisa with arm lengths $L = 3$km [5]. Similarly to the LIGO detectors, they were upgraded from their initial design to advanced Virgo, aVirgo, and started operating in science mode in August 2017.

Using a network of gravitational wave detectors is of particular importance for confident source detection, since the source is coherently detected by several detectors. But this coherent detection is also of high importance for the sky localisation of the source. The beam pattern functions $F_+(\alpha, \delta, \psi)$ and $F_\times(\alpha, \delta, \psi)$ are detector dependent and have different sensitivities depending on the detector orientation and localisation. The typical spatial response, or antenna pattern, of a single gravitational wave interferometer is illustrated in the left hand plot of Figure 2.3. For some regions of the sky, the detector has very poor sensitivity and is essentially unable to detect the gravitational wave [22]. But if we combine several detectors with different orientations, we can make these blind spots in the sky shrink, or even disappear. For the current network of detectors, the two aLIGOs have almost the same arm directions and aVirgo has a different orientation. In the right hand plot of Figure 2.3 [23], we illustrate the sky sensitivity of

the network (aLIGO,aVirgo) for the detection of a face-on binary neutron star at 160 Mpc. We observe that the detector network is capable of confidently detect most of the sources for a variety of source sky positions and only a handful of sources are not detected by the network. Furthermore, when the source is detected, its sky localisation is comprised within a small patch in the sky. The reason for this comes from the coherent detection of the gravitational wave signal by the network of detectors.

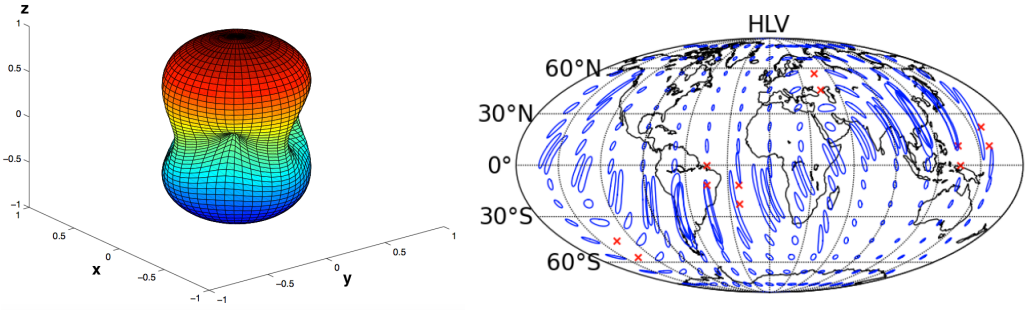

Figure 2.3: (Left)Typical antenna pattern of an interferometric gravitational wave detector where the arms are in the (x,y) plane and oriented along the x and y axis [24]. (Right) Sky coverage accuracy for a network of three gravitational wave detectors (aLIGOs,aVirgo) at their designed sensitivities [23]. The ellipses represent the 90% confidence interval for the sky angles of the source and the red crosses represent sources that are not confidently detected by the network. The sources taken in this study are face-on binary neutron stars at luminosity distance of 160 Mpc.

As we have seen, general relativity predicts that gravitational waves propagate at the speed of light c. Thus, an incident gravitational wave will arrive with different time delays at the detectors. In the case of the two aLIGOs the time delay is around 10 ms, while the time delay between aVirgo and the two aLIGOs is around 27 ms. These time delays can be computed analytically and are functions of α and δ. Thus, by measuring the delays on the detector signal outputs we can have better constraints on α and δ via source triangulation [25, 26, 27]. In Figure 2.4, we illustrate how source triangulation works for the network aLIGO Hanford (H), aLIGO Livingston (L) and aVirgo (V). For each pair of detector, (H,L, orange), (H,V, green) and (L,V, blue), the triangulation with the time delays give us constrains on the source localisation, that must be located on an annulus on the sky concentric about the line between the two detectors represented on the figure. If we have a coherent detection with a three detector network (H,L,V), the source is then located on the intersections of the previous annuli. In this case, we have two possible solutions for the source: the true source position (S on the figure) and a mirror image ($S^{'}$). In the case of a four detector network, the mirror image disappears and the source is then located within a single region on the sky.

In Figure 2.5, we present the design sensitivites for aLIGO and aVirgo. For the sensitivity of aLIGO, we have used the fit derived in [24]

$$S_n(f) = 10^{-49}\left(x^{-4.14} - 5x^{-2} + \frac{111(1-x^2+0.5x^4)}{1+0.5x^2}\right), \tag{2.6}$$

where $x = f/215$. For aVirgo we used a linear interpolation from the data published online associated with the curves presented in [5]. he lower frequency cutoff of these detectors is 20 Hz. There are a variety of gravitational wave sources that can be detected in this band of frequency, whether they are transient or continuous sources. Most of the transient sources are expected to come from the coalescence of a compact binary formed by neutron stars, black holes or the combination of the two. In the case of the black holes, the masses expected are either stellar mass black holes (between $5M_\odot$ and approximatively $100M_\odot$) or intermediate mass black holes with masses superior to $100M_\odot$. Since the frequency at the merger is inversely propoertionnal to the total mass of the black holes, the limiting mass for detection is dependent on the lower frequency cutoff of the detector. The corresponding detection ranges for binary neutron stars with masses $1.4M_\odot$ is 210 Mpc and 140 Mpc for aLIGO and aVirgo at design sensitivity respectively [5, 6]. There are also a number of other transient sources that can be measured such as supernovae in or near the galaxy, long gamma ray bursts and cosmic strings. For the continuous sources, the detectors are searching for gravitational waves emitted by fast-spinning neutron stars and stochastic gravitational wave backgrounds.

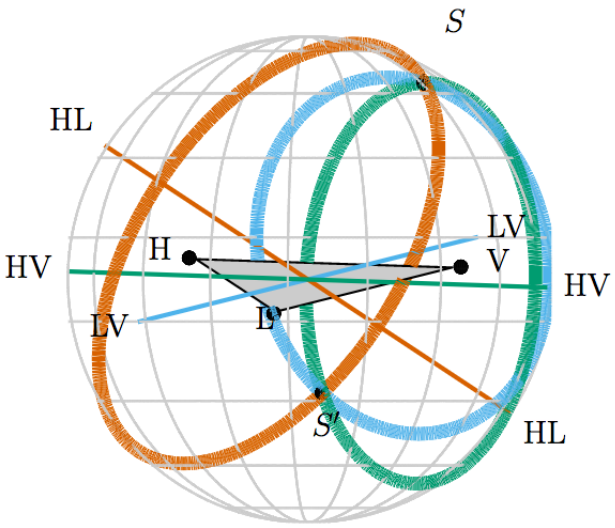

Figure 2.4: Illustration of sky localisation for a source using triangulation with networks of two and three detectors with aLIGO and aVirgo [23].

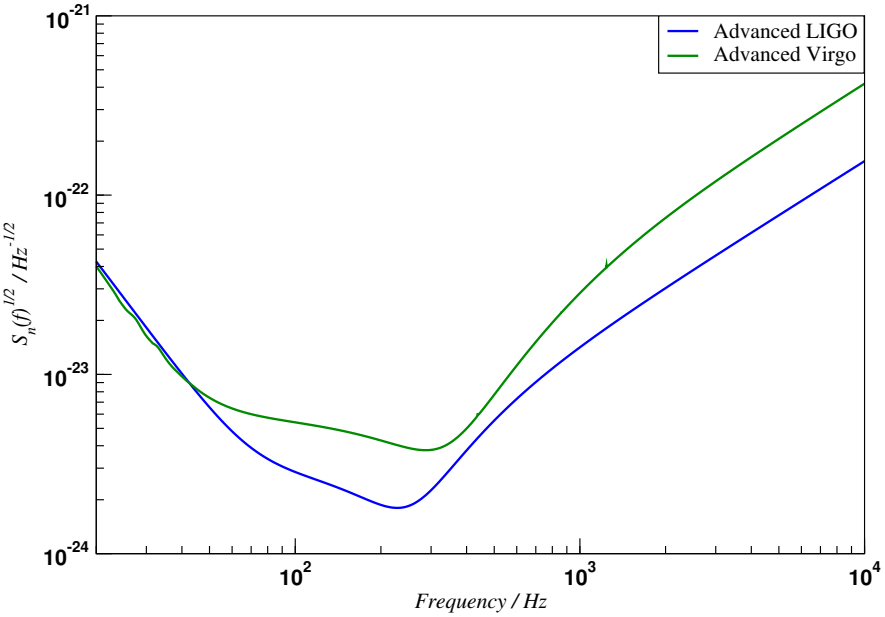

Figure 2.5: Design sensitivity of advanced LIGO [6, 24] and advanced Virgo [5]

2.3 Future detectors

A variety of gravitational waves detectors will be operational in the upcoming years. For ground-based observations, other detectors will join the current second generation while a third generation of detectors are currently designed for the future. In addition to that, the space-based observatory LISA will come online in the future to observe gravitational waves in a different frequency band.

2.3.1 Second-generation ground-based detectors

In Figure 2.6, we present a summary of all the second generation gravitational waves detectors along with their localisations and we give the designed sensitivity of KAGRA in Figure 2.7. KAGRA is a Japanese underground interferometer with arm lengths $L = 3$ km that is located at the Kamioka mine. In addition to the technology developed for aLIGO and aVirgo, KAGRA should implement cryogenic cooling in the

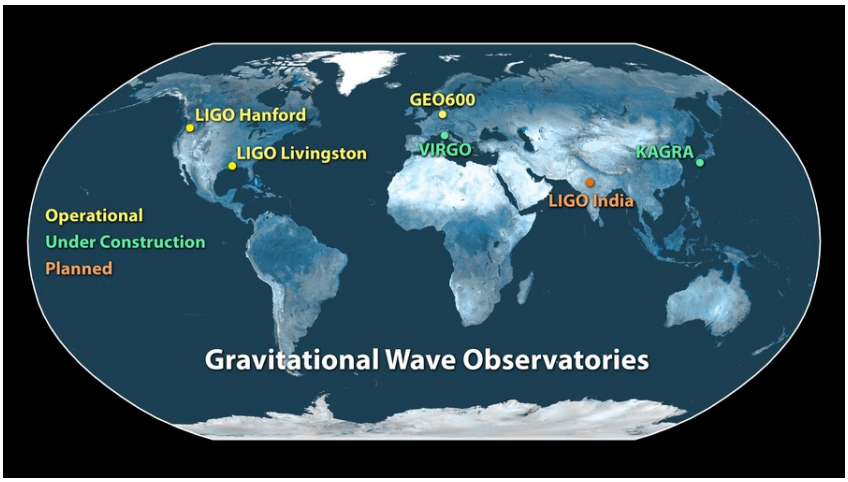

Figure 2.6: Summary of ground-based second generation gravitational waves detectors network around the world [28].

detector that will reduce the various noise sources. The designed sensitivity of KAGRA is expected to be similar to the sensitivity of aLIGO and aVirgo with a binary neutron star range detection around 140 Mpc. LIGO-India (iLIGO) is an Indian gravitational wave detector that will be built using the same technological design developed for the LIGO detectors. The project has recently received approvement from the Indian government and the choice for the detector site is currently in discussion.

2.3.2 Third-generation ground-based detectors

Unlike the previous detectors, the third generation ground-based detectors will use brand new technological concepts to significantly decrease the sensitivity of current detectors. Two detectors are currently being proposed: Einstein Telescope (ET) that will replace aVirgo, and Cosmic Explorer that will replace aLIGO.

Einstein Telescope is a European proposal for a gravitational wave detector with arm lengths of $L = 10$ km. [29]. The final setup of the detector is not fixed yet but most of the studies consider that the detector will be located deep underground to shield it from gravitational perturbations, and will use a setup with three laser links instead of two and will have a triangular configuration. As a consequence, we expect that the lower frequency cutoff will be decreased to 1 Hz and the sensitivity will be lower on the entire frequency band. Cosmic Explorer is an American proposal for a gravitational wave detector that would also have a three laser link setup with arm lengths of $L = 40$km [30]. Owing to the large scale of the arms, the detector will likely be on the surface. In Figure 2.7, we give the designed sensitivity of Einstein Telescope and Cosmic Explorer. These third generation detectors will be capable of detecting all the sources that are currently being searched with the current network with a large increase in the sensitivity and detection range. Since the lower frequency cutoff of the detector is lowered, the detectors will also be far more sensitive to the coalescence of intermediate mass black hole binaries. In addition, ET and Cosmic Explorer will also be able to put better constraints on tests of general relativity.

2.3.3 Space-based detectors

Even though third generation detector will lower the frequency cutoff of the detector, the seismic noise and Newtonian noise will be limiting noise sources in the low frequency band. To have access to lower frequency, one then needs to have a space-based detector where these noise sources do not affect anymore the measurements. LISA is a space-based gravitational wave detector that has been selected as the L3 mission for the Cosmic Vision program of the European Space Agency [8]. The bandwidth of frequencies in which LISA is sensitive is different from ground-based detectors, with frequencies spanning between 10^{-4} and 1 Hz. The technology for LISA is also based on interferometry, where laser beams are sent back and forth between three satellites orbiting in a triangular shape around the Sun. In the current design of the mission, there are three inteferometry channels with laser links along the three arms and the arm length will be $L = 2 \times 10^6$ km. Recently, crucial technology for the mission was tested with the technological demonstrator LISA Pathfinder [31], and the final mission should be launched in 2034.

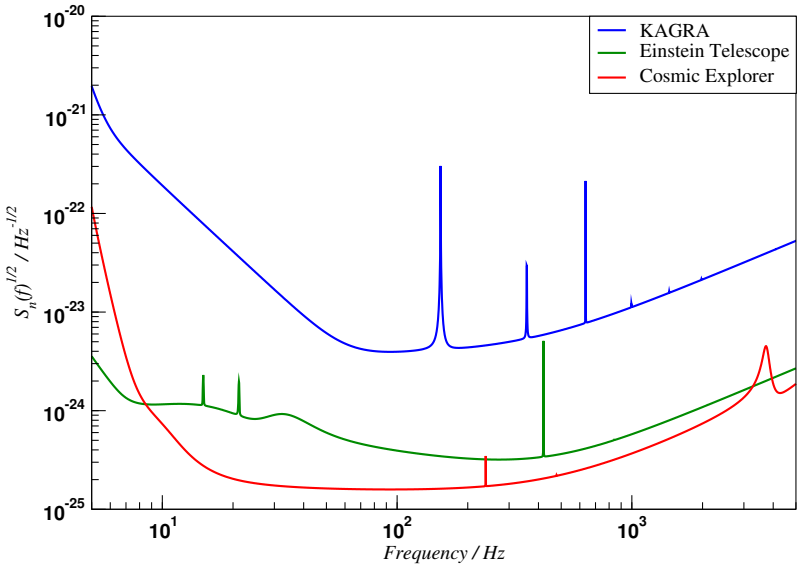

Figure 2.7: Designed noise power spectral density for KAGRA (version B from [7]), Einstein Telescope [29] and Cosmic Explorer [30].

At the beginning of this thesis, a reduced version of LISA called eLISA was planned [32], where there were only two laser links with length $L = 1 \times 10^6$ km. In the bottom plot of Figure 2.8, we give the designed sensitivity of eLISA and LISA. The band frequency of LISA is different from the frequency band of ground-based detectors. As a consequence, the gravitational wave sources that can be detected by this detector are different. The most numerous sources that LISA is expected to detect are compact galactic binaries composed of white dwarves, neutron stars and stellar mass black holes orbiting each other with short periods. In addition to that, LISA will also be capable of detecting the coalescence of supermassive black hole binaries with black hole source-frame masses ranging from 10^3 to $10^8 M_\odot$ at redshift z between $0 \leq z \leq 20$. Another interesting type of source in the LISA band will be extreme mass ratio inspirals where a stellar mass compact object spirals towards a supermassive black hole. Finally, LISA will also be capable of probing cosmology by measuring the stochastic gravitational wave background.

Another space-based Japanese detector DECIGO is planned to be launched in the future. As for LISA, this detector is designed as a triangular interferometer with three spacecrafts separated by $L = 10^3$ km. The frequency band of the detector will be between 0.1 and 10 Hz and the main gravitational wave sources expected in this band of frequency are intermediate mass black holes and compact galactic binaries transitioning from the LISA to the ground-based band of frequencies.

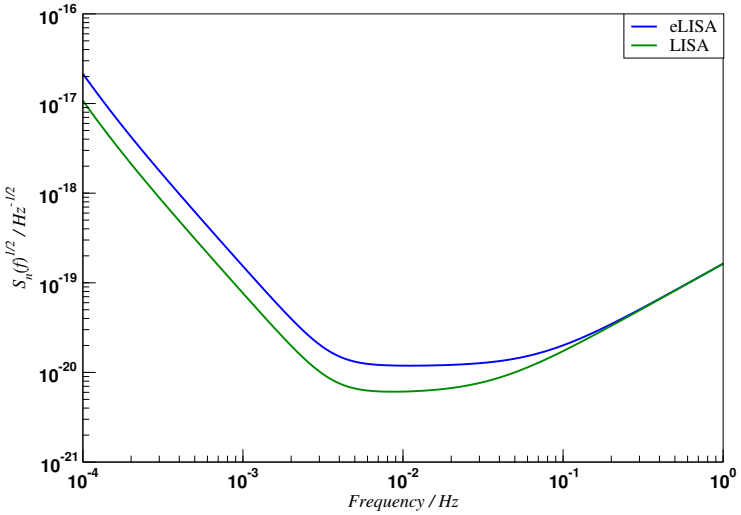

Figure 2.8: Designed sensitivity for eLISA and LISA [32]

.

Chapter 3

Astrophysical sources of GWs

In this chapter, we will describe the stellar formation scenarios that lead to the formation of binary systems composed of compact objects. These sources are one of the main sources of gravitational waves for both ground-based and space-based gravitational wave detectors. In the first section, we will review isolated stellar evolution for low and high mass stars. In the second section, we will investigate the compact objects that are formed at the end of stellar evolution of single stars. Finally, we will describe stellar evolution in binaries, and how it leads to the formation of compact binaries.

3.1 Single star evolution

Stars are formed in cold clouds of interstellar gas and dust mainly composed of hydrogen. Four physical mechanisms play a major role in these clouds during star formation: gravity which pulls matter together, thermal pressure which opposes gravity and pushes matter apart, magnetic pressure which also opposes gravitation and radiation which controls temperature.

Under certain conditions, gravity manages to overcome local thermal pressure in a region of the molecular cloud, hence leading to gravitational collapse of the region. This phenomenon is known as Jeans instability and leads to the formation of a so-called protostar. Gravitational collapse is possible when the total mass of a region of the molecular cloud is superior to the Jeans mass M_J given by,

$$M_J \sim \left(\frac{k_B}{Gm}\right)^{3/2} \left(\frac{T^3}{\rho}\right)^{1/2}, \tag{3.1}$$

where k_B is Boltzmann's constant, G is the gravitational constant, ρ and T are the mean density and temperature of the region and m is the mean particle mass inside the region. Equivalently, the Jeans length L_J gives us the value of the smallest size of the region that can collapse under gravity inside the molecular cloud,

$$L_J \sim \left(\frac{k_B T}{G\rho m}\right)^{1/2}. \tag{3.2}$$

During this collapse, gravitational energy is converted into thermal energy that heats the gases in the interior of the protostar. Gravitational contraction is for the moment the only source of heat for the protostar since the temperature of the core of the protostar is to low to ignite nuclear reactions. At first, the heat inside the protostar is evacuated towards the surface through both convection and radiation, owing to the fact that the core of the protostar is transparent. However, at some point the density in the interior becomes so high that the core becomes opaque and does not radiate anymore. The pressure in the interior then builds up leading to an increase in core temperature. The temperature keeps increasing as the protostar accretes material, up until a point where the temperature in the core reaches the temperature needed to start the fusion of hydrogen, $T \approx 5 \times 10^6 K$, and the protostar becomes a main sequence star. The timescale for a star to go from the protostar to the main sequence state is highly dependent on the initial total mass of the collapsed region of the cloud gas. As an example, for an initial $30 M_\odot$ protostar, the time to reach the main sequence lasts around 10^4 years, while for smaller initial mass of $1 M_\odot$, the process is longer and can last for 10^6 years.

For a main sequence star, the nuclear reactions inside the core generate heat and internal thermal pressure that is capable of balancing gravity. At this point, the star is then both in hydrostatic and

thermal equilibrium that can be written for a small shell of material of a spherical star at radius r as,

$$\frac{\mathrm{d}P(r)}{\mathrm{d}r} = -\frac{Gm(r)}{r^2}\rho, \tag{3.3}$$

$$\frac{\mathrm{d}F(r)}{\mathrm{d}r} = 4\pi r^2 \rho q(r), \tag{3.4}$$

where P is the pressure, $m(r)$ is the mass of the star comprised between its center and radius r, F is the heat flux of energy and $q(r)$ is the heat generated by nuclear burning. Under these conditions, it is possible to derive a simple power-law between the mass and luminosity of a main sequence star as [33, 34],

$$\frac{L}{L_\odot} \sim \left(\frac{M}{M_\odot}\right)^a \text{ with } \begin{cases} a = 2.3 & \text{for} \quad M < 0.43 M_\odot, \\ a = 4 & \text{for} \quad 0.43\ M_\odot < M < 2\ M_\odot, \\ a = 3.5 & \text{for} \quad 2\ M_\odot < M < 20\ M_\odot, \\ a = 1 & \text{for} \quad M > 20\ M_\odot. \end{cases} \tag{3.5}$$

The luminosity of the star can also be related to the surface temperature of the star T through the Stefan-Boltzmann law as,

$$L = 4\pi R^2 \sigma T^4. \tag{3.6}$$

A common graph used to describe stellar evolution is the Hertzprung-Russel (HR) diagram where the state of a star is represented in terms of its luminosity and surface temperature in a log-log plot. As the exponent a in the mass-luminosity relationship from Eq. (3.5) does not vary much, the main sequence stars are located on a diagonal line on the HR diagram, as shown in Figure 3.1.

From the main sequence stage of stars, we have two different stellar evolution scenarios depending on the mass M of the main sequence star,

- low-mass stellar evolution for stars with masses between $0.3\ M_\odot < M < 8\ M_\odot$
- high-mass stellar evolution for stars with masses $M > 8\ M_\odot$

In the following sections, we will briefly describe the main steps of these two stellar evolution scenarios

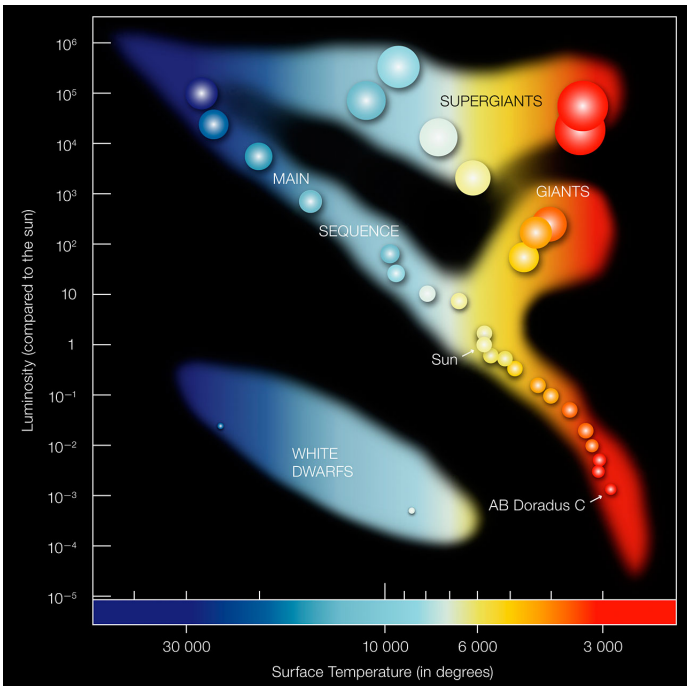

Figure 3.1: Hertzsprung Russel diagram (ESO, [35])

3.1.1 Low-mass stellar evolution

PP cycle			Triple alpha process		
${}^1_1\mathrm{H} + {}^1_1\mathrm{H}$	$\longrightarrow$	${}^2_1\mathrm{D} + e^+ + \nu_e$	${}^4_2\mathrm{He} + {}^4_2\mathrm{He}$	$\longrightarrow$	${}^8_4\mathrm{Be}$
$e^+ + e^-$	$\longrightarrow$	2γ	${}^4_8\mathrm{Be} + {}^4_2\mathrm{He}$	$\longrightarrow$	${}^{12}_6\mathrm{C}$
${}^2_1\mathrm{D} + {}^1_1\mathrm{H}$	$\longrightarrow$	${}^3_2\mathrm{He} + \gamma$			
${}^3_2\mathrm{He} + {}^3_2\mathrm{He}$	$\longrightarrow$	${}^4_2\mathrm{He} + 2{}^1_1\mathrm{H}$	${}^{12}_6\mathrm{C} + {}^4_2\mathrm{He}$	$\longrightarrow$	${}^{16}_8\mathrm{O} + \gamma$

Table 3.1: (left)Table presenting the main set of nuclear reactions to transform hydrogen in helium for low mass stars, PP cycle. (right) Table presenting the main set of nuclear reactions to transform helium into carbon and oxygen for low mass stars, triple alpha process.

For this range of masses and temperature, the main reaction that converts hydrogen into helium in the core is the proton-proton chain reaction, or PP cycle. The nuclear reactions of the PP cycle are presented in Table 3.1. As long as there is enough hydrogen in the core, the star produces helium and stays on main sequence. As an example, the typical timescale required for a $1M_\odot$ star, like the Sun, to entirely convert its hydrogen to helium is around 10^{10} years [36].

When the hydrogen is depleted in the core, the nuclear reactions stop and there is no thermal pressure to counteract gravity. In fact, the temperature in the core of low mass stars is not sufficient at this stage to trigger the nuclear reaction of helium. As a consequence, the star is formed with an inert helium core that is not in thermal equilibrium, and starts to collapse. This gravitational collapse starts to heat the helium core and through thermal convection, sufficient heat is transported towards the shells of the stars to have hydrogen burning in the exterior of the star. These nuclear reactions are responsible for a huge expansion of the envelope of the star. As an example, the radius of the Sun is expected to increase by a factor of about 250 during its expansion. In addition, the surface temperature of the star decreases and the emission spectrum of the star is shifted to the red part of the visible spectrum. In this stage of the evolution, the stars are said to be red giants and are represented in the HR diagram of Figure 3.1.

At some point in the red giant phase, the gravitational collapse of the core heats up the core to a temperature sufficient to trigger helium burning. The main nuclear reaction for helium is the triple alpha process that converts three helium atoms into a single carbon atom. An additional reaction also creates an oxygen atom when a carbon atom fuses with another helium atom as detailed in Table 3.1. For stars with initial main sequence masses between 0.8 and $2.0M_\odot$, these nuclear reactions start a violent thermal runaway nuclear reaction that is referred as to helium flash [37]. Now that the core has a new source of nuclear reaction, the gravitational collapse stops and the star starts to regain hydrostatic and thermal equilibrium. This phase of the star is called the horizontal branch.

The triple alpha process is much less efficient than the PP cycle in terms of energy creation. This means that the helium core of the low mass star is depleted much faster than in the case of hydrogen, with a typical timescale of about 10^8 years for a $1M_\odot$ mass. When the helium core is depleted, the star starts to collapse once again because the temperature is not high enough to trigger the nuclear reactions for carbon or oxygen. Similarly to the red giant phase, the heat generated by gravitational collapse triggers the reactions of helium in the shells, causing the star to expand once again. In this state, the star is said to be in its asymptotic giant branch.

In this case, the gravitational collapse is not enough to trigger the next set of nuclear reactions. The star is then composed of an inert contracting carbon-oxygen core with an expanding envelope of helium and hydrogen. The expanding envelope is then ejected by stellar winds and forms a planetary nebula. The process is fast and lasts only around 10^4 years. At the end, the matter in the carbon-oxygen becomes degenerate and the core stops contracting, leading to the formation of a white dwarf.

3.1.2 High-mass stellar evolution

In the case of high mass stars, the main set of nuclear reaction for hydrogen burning during the main sequence is the carbon-nitrogen-oxygen cycle (CNO) detailed in Table 3.2. The typical timescale needed to deplete the hydrogen core is faster than for low mass stars and is around 10^7 years.

The departure from the main sequence for high mass stars is quite similar to the one detailed before for less massive stars. When hydrogen is depleted in the core, the core is heated up by gravitational contraction and heat is transported towards the shells of the star. This triggers hydrogen nuclear reactions and causes the expansion of the envelope of the star. This expansion is even larger than in the case of low mass stars, and the star becomes a red supergiant. The size of a red supegiant can be as large as 1500 times the size of the Sun [38]. When the temperature inside the core is sufficient, helium nuclear reactions are triggered and helium starts to be converted into oxygen and carbon. The star is back to thermal and hydrostatic equilibrium and is denoted as a blue supergiant.

When the helium core is depleted, the star once again experiences a gravitational collapse that heats the inert carbon-oxygen core. Unlike low mass stars, the temperature inside the star becomes sufficient to trigger the next set of nuclear reactions, namely the fusion of carbon into neon and magnesium. The nuclear reactions are much faster than for hydrogen and helium and the typical timescale of nuclear reaction for the carbon core is around 600 years for a $25M_{\odot}$ star. At the end of this phase, the star has an inert core of oxygen, neon and magnesium.

The inert core will again be heated through gravitational collapse to trigger another sets of nuclear reactions. This process is repeated a number of times with the successive nuclear reactions of neon, oxygen and silicon. For each process, the timescales is faster as presented in Table 3.3. For the last nuclear reaction of silicon, the end product is iron. However, the nuclear fusion of iron is an endothermic reaction that does not produce heat in the core. When this happens, the star starts to collapse in a violent process called type II supernova [39]. During this process, the outer part of the core collapses towards the center of the star with high velocity and bounces off the core. This leads to a sharp heat increase of the core, where a variety of nuclear reactions start to occur. Among the end products of these nuclear reactions, we find neutrinos that are created by inverse beta decay. Since the neutrinos are weakly interacting with the matter, they carry most of the energy away from the core which leads to the acceleration of the star collapse.

There are different possible outcomes after a Type II supernova if the star is not disrupted in the process. For stars with masses between 8 and about $18M_{\odot}$, the remnant of the core is a neutron star. When the mass is superior to $18M_{\odot}$, the remnant of the core is a stellar mass black hole.

CNO cycle		
${}^{12}_{6}\mathrm{C} + {}^{1}_{1}\mathrm{H}$	$\longrightarrow$	${}^{13}_{7}\mathrm{N} + \gamma$
${}^{13}_{7}\mathrm{N}$	$\longrightarrow$	${}^{13}_{6}\mathrm{C} + e^{+} + \nu_e$
${}^{13}_{6}\mathrm{C} + {}^{1}_{1}\mathrm{H}$	$\longrightarrow$	${}^{14}_{7}\mathrm{N} + \gamma$
${}^{14}_{7}\mathrm{N} + {}^{1}_{1}\mathrm{H}$	$\longrightarrow$	${}^{15}_{8}\mathrm{O} + \gamma$
${}^{15}_{8}\mathrm{O}$	$\longrightarrow$	${}^{15}_{7}\mathrm{N} + e^{+} + \nu_e$
${}^{15}_{7}\mathrm{N} + {}^{1}_{1}\mathrm{H}$	$\longrightarrow$	${}^{12}_{6}\mathrm{C} + {}^{4}_{2}\mathrm{He}$

Table 3.2: Table presenting the main set of nuclear reactions to transform hydrogen in helium for high mass stars, CNO cycle.

3.2 Compact objects

In the previous section, we have seen that the end of stellar evolution leads to compact objects such as white dwarfs, neutron stars or stellar mass black holes. We will briefly describe these compact objects in this section.

3.2.1 White dwarfs (WD)

White dwarfs are the end products of low mass stellar evolution where the core is inert without producing thermal pressure through nuclear reactions. These compact stars have typical masses around $0.6M_{\odot}$ and radius of $0.01R_{\odot}$ [40]. In this case, gravity is balanced by degenerate pressure, that we briefly explain here.

Reaction	Temperature / K	Time scale
H burning	4×10^7	11×10^6 years
He burning	2×10^8	2×10^6 years
C burning	8×10^8	2000 years
Ne burning	1.6×10^9	8 months
O burning	1.9×10^9	3 years
Si burning	3.3×10^9	18 days
Fe burning	$> 7.1 \times 10^9$	1 second

Table 3.3: Example of the cycle of nuclear reactions with their temperature and time scale for a $15 M_\odot$ star [39].

The density of matter inside a WD is extremely high with typical density between 10^7 and 10^{11}kg m^{-3}. As gravity tends to pull material together, the electrons begin to oppose gravity because of the Pauli exclusion principle. As electrons are fermions, they are described by Fermi-Dirac statistics which give us the distribution of electrons over the energy states of a system as

$$f(p) = \frac{1}{\exp\left[(\epsilon_p - \mu)/kT + 1\right]}, \tag{3.7}$$

where μ is the chemical potential and ϵ_p is the kinetic energy of the electron particle with momentum p

$$\epsilon_p = (m_e^2 c^4 + p^2 c^2)^{1/2} - m_e c^2, \tag{3.8}$$

and m_e is the mass of the electron. In the limit of low temperature or high density ("cold gas"), the Fermi-Dirac distribution reduces to a simple expression

$$f(p) = \begin{cases} 1 \quad \text{if} \quad p \leq p_F \quad \text{or} \quad \epsilon \leq \epsilon_F, \\ \\ 0 \quad \text{if} \quad p > p_F \quad \text{or} \quad \epsilon > \epsilon_F, \end{cases} \tag{3.9}$$

where p_F is the Fermi momentum and ϵ_F is the associated Fermi energy. We can derive an expression for the Fermi momentum as,

$$p_F = \left(\frac{3h^3 \rho}{8\pi m_H \mu_e}\right)^{1/3}, \tag{3.10}$$

where μ_e is the mean electron mass per hydrogen nucleus and m_H is the mass of one hydrogen atom. The resulting degenerate pressure from the electrons can be expressed in terms of the Fermi momentum as,

$$P = \frac{1}{3} \int_0^{p_F} \left(\frac{8\pi p^2}{h^3}\right) pv \mathrm{d}p. \tag{3.11}$$

This last expression has different forms depending on if the electrons are relativistic ($v \approx c$), or non-relativistic ($v = p/m_e$). The following expressions for the degenerate pressure are,

$$P_{NR} \sim \left(\frac{\rho}{\mu_e}\right)^{5/3} \text{ if relativistic} \tag{3.12}$$

$$P_R \sim \left(\frac{\rho}{\mu_e}\right)^{4/3} \text{ if non-relativistic} \tag{3.13}$$

Given the expressions for degenerate pressure, one can then derive a relationship between the WD radius and its mass. Considering that gravitational and internal energies balance each other in equilibrium, we can write

$$\frac{GM^2}{R} \sim PV \sim PR^3 \tag{3.14}$$

where P is the degenerate pressure of electrons. If the electrons are non-relativistic, we can use Eq. (3.13) to write,

$$P \sim n_e^{5/3} \sim M^{5/3} R^{-5}. \tag{3.15}$$

Putting all together, we find the relationship between the radius and mass of the white dwarf is,

$$R \sim M^{-1/3} \tag{3.16}$$

This equation is a bit counter-intuitive since it states that the more massive a white dwarf, the smaller it is. In addition, the white dwarf has an upper limit for its mass that is given by the Chandrasekhar mass, M_{Ch} [41],

$$M_{Ch} \sim \left(\frac{hc}{G}\right)^{3/2} \frac{1}{m_P^2} \frac{1}{\mu_F^2} \tag{3.17}$$

Above the Chandrasekhar mass, the pressure from degenerate electrons is not sufficient to counteract gravitational collapse. As an example, the mass of a white dwarf accreting matter in binary system can increase above M_{Ch} and the white dwarf is then disrupted in a Type Ia supernova.

3.2.2 Neutron stars (NS)

As we have seen, neutron stars are the remnants of high mass stellar evolution for masses M comprised between 8 and about $18M_\odot$. The typical value for the mass of the neutron star is about $1.4M_\odot$ for a radius of $R \sim 10$ km. As a consequence the density of the matter inside a neutron star is extremely high with values around the nuclear density, $\rho \sim 10^{17}$ kg / m^3. At such densities, protons and electrons can not move freely and form neutrons via the following nuclear reaction,

$$p + e^- \longrightarrow n + \nu_e. \tag{3.18}$$

In this condition of matter, the neutrons are degenerate and the degenerate pressure of neutrons counteract gravity.

The structure of the neutron stars is thought to be divided into four parts, each of them having specific states of matter in its interior as shown in Figure 3.2. The outer crust is thought to be made of a lattice of heavy nuclei inside a sea of electrons, which is close to the same state of matter found in WD. When going deeper towards the inner crust, the pressure increases and the nuclear reaction between protons and electrons into neutrons and neutrinos in Eq. (3.18) is favored. In the inner crust, the state of matter in the neutron star is then formed of free electrons and neutrons, along with neutron-rich nuclei. In the outer core, the pressure is so high that the remaining nuclei are not stable anymore and are transformed into a free fluid of neutrons. It is believed that the fluid of neutrons in the interior should exhibit exotic properties such as superconductivity and superfluidity. Finally when going even deeper in the interior of the neutron star, we have the inner core, where we think that the matter might be on the form of a plasma of quarks and gluons. However, understanding the equation of state of the matter inside a neutron star is still an active subject of research [42].

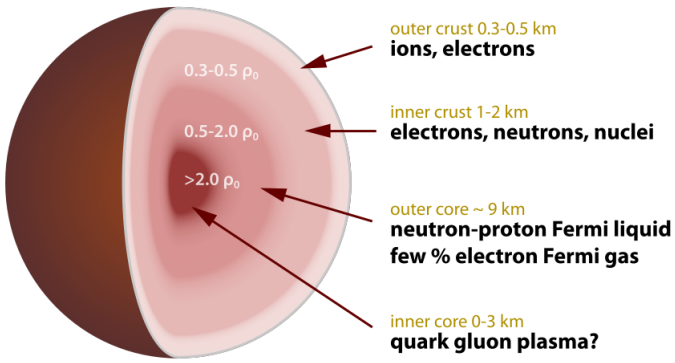

Figure 3.2: Illustration of the model of the interior of a neutrons star [43].

As a consequence of their small radius and angular momentum conservation, neutron stars rotate rapidly with period of rotation lying in the range of one millisecond to a few tens of seconds with intense magnetic fields $B \sim 10^{12}$G. The combination of these two physical properties is of particular interest for the study of neutron star. In fact, the rotating magnetic field produces a strong electric field at

the surface acting on the free charged particles of the outer crust. As a result of this, radio waves are radiated from the magnetic poles to outer space. Most of the time the axes of rotation and magnetic fields (beam directions) are not aligned, and it is then possible to observe a pulsating radiation coming from the neutron star. This is the reason why these objects are referred to as pulsars. The variation of the period of pulsations extremely small and can be used as a tool for time measurement. During its lifetime, the period of rotation-powered pulsars slowly increases owing to energy loss through emission of electromagnetic waves.

As for WDs, neutron stars have an upper mass limit corresponding to the limit where degenerate neutron pressure cannot counteract gravity, which is known as the Tolma-Oppenheimer-Volkoff limit [44, 45]. The value of the limiting mass range between 1.5 and $3M_\odot$ depending on the mass of the progenitor [46]. For now, the highest mass observed for a neutron star is $2.01M_\odot$ [47], but we are hoping to use gravitational wave observations in order to observe what is the transition between neutron star and stellar mass black holes [48].

3.2.3 Stellar mass black holes (BH)

Stellar black holes represent the last stage of evolution possible for high mass stars with masses superior to $18M_\odot$. In this case, gravitational collapse is not balanced by other forces such as electron or neutron degeneracy pressure. Under its own weight, the star collapses to a point and becomes a singularity in spacetime. The range of masses for stellar mass black holes are between 5 and a few tens of solar masses [49, 50, 13].

A Schwarzschild black hole is described in terms of a singularity of spherical event horizon and a singularity of spacetime at its center. To have a particle stand still at the event horizon of the black hole, its velocity needs to be equal to the speed of light c. And if the particle crosses the event horizon, it has no possibility to escape the gravitational attraction of the black hole. The radius of the spherical horizon is given by the Schwarzschild radius

$$R_{Sch} = \frac{2GM_\bullet}{c^2}, \tag{3.19}$$

where $M_\bullet$ is the mass of the Schwarzschild black hole. As an example, the Sun, if it could be converted into a black hole, would have a Schwarzschild radius that is about 3 km.

If we now consider a massive particle orbiting the Schwarzschild black hole, the minimal radius for which the particle can orbit stably around the black hole is given by,

$$r_{lso} = \frac{6GM_\bullet}{c^2}, \tag{3.20}$$

where r_{lso} is referred as to radius of the last stable circular orbit If we take a photon particle, the radius then becomes,

$$r_{lso} = \frac{3GM_\bullet}{c^2}, \tag{3.21}$$

The spherical region defined by this radius is called the photosphere.

A Kerr black hole is a black hole that is rotating with spin a. In this case, the radius for the event horizon R_h is expressed as,

$$R_h = \frac{R_{Sch}}{2}\left[1 + \sqrt{1 - \left(\frac{Jc}{GM_\bullet^2}\right)^2}\right], \tag{3.22}$$

where $a = J/(M_\bullet c)$ is the angular momentum of the black hole. As in the case of Schwarzschild black hole, any particle that crosses this event horizon can not escape the black hole gravitational attraction. However, unlike Schwarzschild black hole, Kerr black holes also possess a second event horizon or outer horizon with a radius,

$$R_o = \frac{R_{Sch}}{2}\left[1 + \sqrt{1 - \left(\frac{Jc\cos(\theta)}{GM_\bullet^2}\right)^2}\right], \tag{3.23}$$

where θ is the azimuthal angle. The region delimited by these two horizons is called the ergosphere as represented in Figure 3.3. In this region, the particles are forced to rotate along with the Kerr black hole. As a result, energy is transferred to the particle, and the Kerr black hole loses energy in the process.

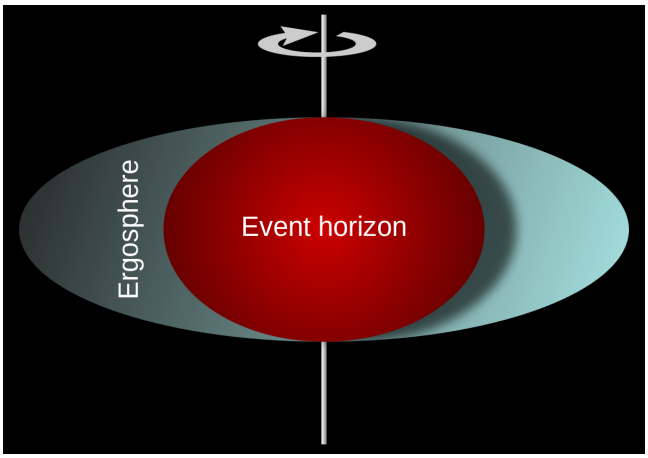

Figure 3.3: Illustration of the horizon and ergosphere of a Kerr black hole [51].

3.3 Binary star evolution

In this section, we will describe the stellar evolution for binary stars. We will see how low mass binaries can form close binary white dwarf systems. In the case of high mass binaries, we will see that these systems can form binaries composed of neutron stars and black holes.

3.3.1 Low-mass binaries

The standard evolution of binary system with low mass components is shown on Figure 3.4. This diagram is not exhaustive but is a good picture of the probable evolutionary process of such binaries. We start the evolution from two main sequence stars in a binary. The more massive star will evolve first and enter the red giant phase. During the expansion of the red giant star, at some point the star causes mass transfer by a phenomenton called Roche lobe overflow. This corresponds to a situation where the outer envelope of a star starts to expand into the region where the gravitational attraction of the other star in the binary is dominant. This results in a mass transfer from a donor star to a companion star. In this case, the donor is the red giant and the companion is the main sequence star. When the mass of the donor is superior to the mass of the companion, which is the case here, the mass transfer is unstable and the binary enters a phase known as a common envelope.. During this phase, the two stars share the same envelope and the distance between the two stars decreases owing to friction during the orbital motion in the envelope.

At the end of the common envelope, the system is then composed of a white dwarf, that was the originally more massive star, and a main sequence star. If the orbital separation of the binary is small and the white dwarf has a low mass, then the main sequence star can overflow its Roche lobe leading to mass transfer from the main sequence star to the white dwarf. These systems are referred as to cataclysmic variables. If the mass transfer raises the mass of the white dwarf close to the Chandrasekhar limit, it triggers a Type Ia supernova that disrupts the white dwarf

Now, in the case where the orbital separation is large at the end of the first common envelope phase, the main sequence star starts to evolve to its red giant phase. At some point, the red giant star starts to fill its Roche lobe and we have unstable mass transfer that leads to a second common envelope phase. If the core of the companion star is not degenerate when entering common envelope, the system is composed of a WD with a helium star. Once again, a stable mass transfer from the helium star to the white dwarf can happen which can lead to a Type Ia supernova if the mass of the white dwarf is above the Chandrasekhar limit.

In the case where the core of the companion mass is degenerate, after the outflow of the common envelope, assuming that the two stars did not merge, the binary is formed with two WDs. The two compact objects are very close to each other owing to angular momentum loss during common envelope. Close binary WD evolution is mainly governed by angular momentum loss from gravitational waves and chemical composition of the compact objects. If two WDs merge, they can either form a neutron star (accretion induced collapse) or explode as a type Ia supernova. When one of the WDs has a helium core, the system can become a cataclysmic-like variable.

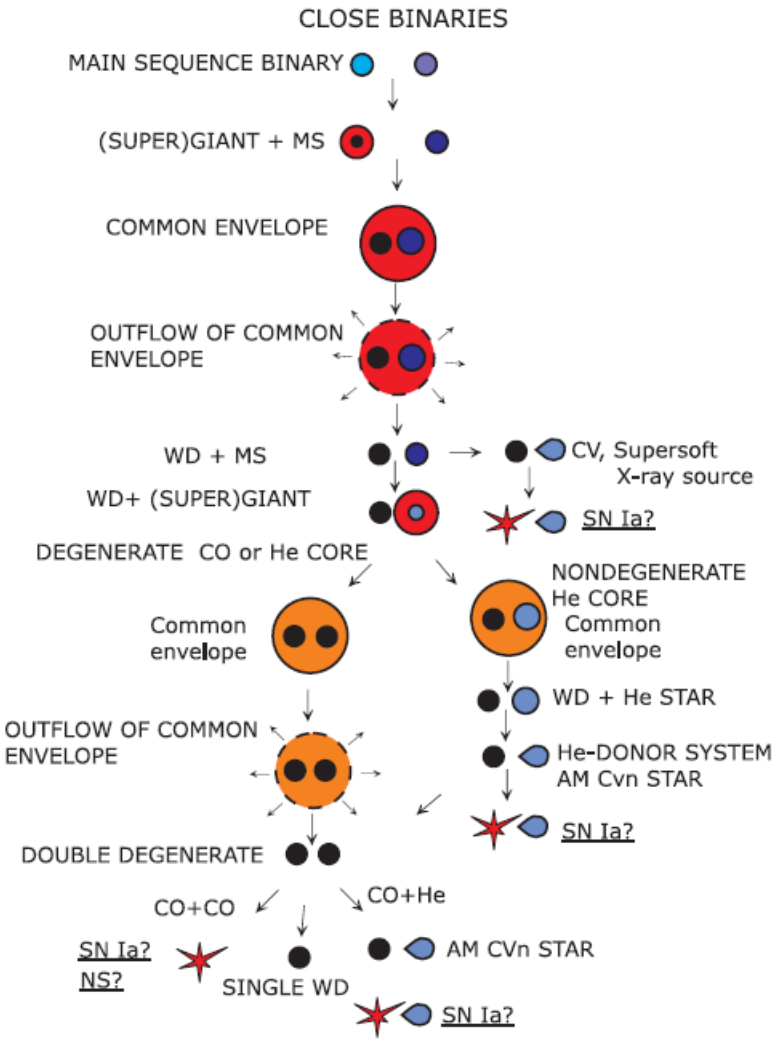

Figure 3.4: Diagram presenting the typical stellar evolution of low mass binaries [52]

3.3.2 High-mass binaries

As in previous section we give a description of the most common scenario for high mass binary stellar evolution, which are summarized on Figure 3.5. Starting with two high mass main sequence stars, the more massive star of the binary evolves faster to a red supergiant phase and overfills its Roche lobe. During the mass transfer most of the hydrogen envelope of the more massive star is transferred to the companion. Deprived of its hydrogen atmosphere, the more massive star is composed of a helium rich core and is called a Wolf-Rayet star [53]. At the end of its evolution, the Wolf-Rayet star explodes as a core-collapse supernova whose remnant could be either a neutron star or a stellar mass black hole. Recalling that the upper mass limit of a neutron star is around 3 $M_\odot$ and that the average mass of a stellar mass black hole is between 5 $M_\odot$ and tens of solar masses, it is possible that the other main sequence star is more massive than the compact object. As a consequence, the companion star starts to evolve to its red giant phase and overfills its Roche lobe.

In this condition, the mass transfer is unstable, and the compact object and the helium core of the companion star shares the same envelope. Two different outcomes for this common envelope phase are possible depending on the loss of angular momentum during common envelope. If the two stars merge together, we are left with compact object inside an outer envelope that is known as a Thorne-Zytkow object [54]. If the two stars did not merge together when the envelope is ejected, we are left with the binary system composed of a compact object with a Wolf-Rayet star. As before, later in its stellar evolution the Wolf-Rayet star explodes as a supernova. Two possible outcomes are then possible for the binary. If the kick velocity of the supernova remnant is too strong, the binary is disrupted and the system

is left as two single compact objects. In the case where the kick velocity is not sufficient to disrupt the binary, we are left with a binary composed of two compact objects that are either neutron stars or stellar mass black holes.

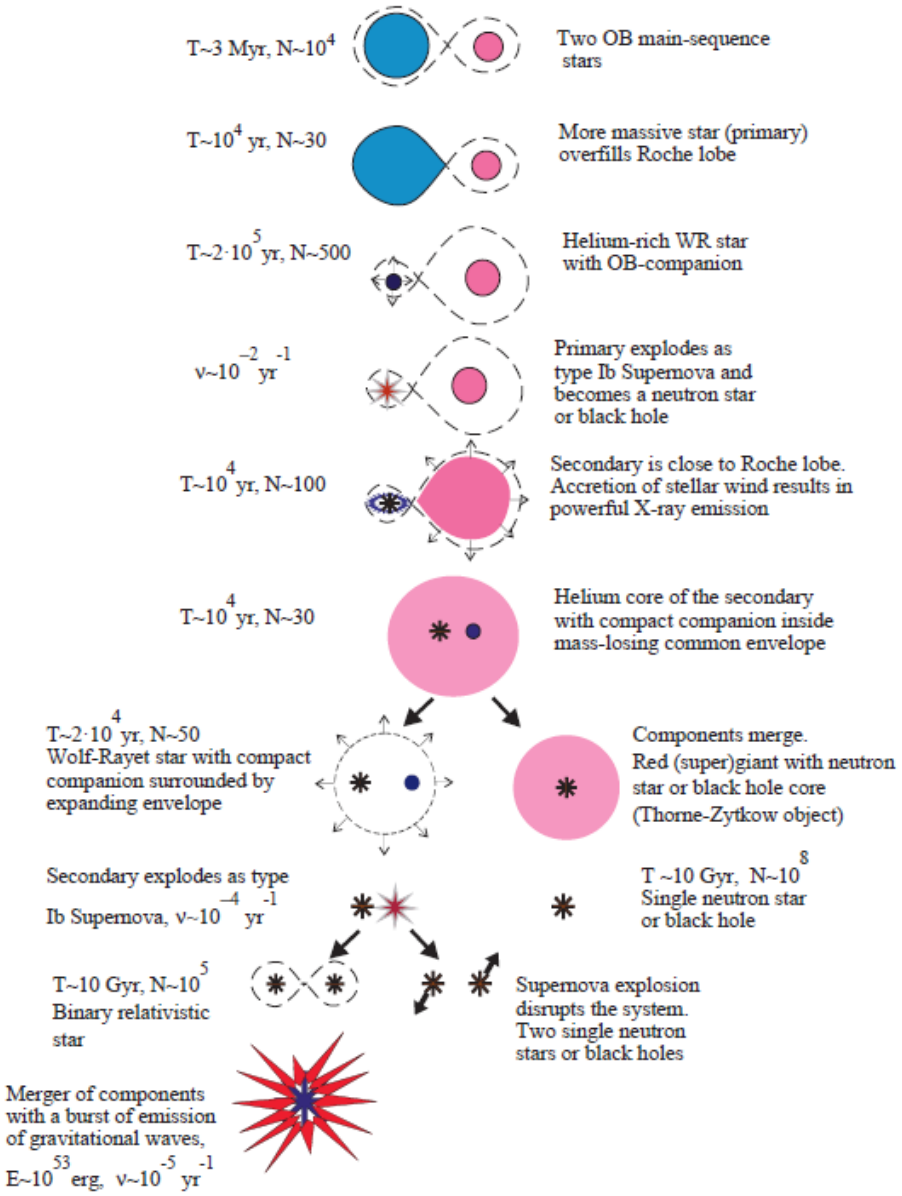

Figure 3.5: Diagram presenting the typical stellar evolution of high mass binaries into neutron star/black hole binary systems (from [52])

Chapter 4

Gravitational wave data analysis

A gravitational wave detector measures a time domain strain, $s(t)$, that is the combination of the noise $n(t)$ with the gravitational wave signal $h(t)$

$$s(t) = h(t) + n(t), \tag{4.1}$$

where we assume that the noise is stationary and Gaussian. The first aspect of gravitational wave data analysis is the detection problem, or, how can we infer the presence of a gravitational wave signal $h(t)$ from the detector output $s(t)$? To answer this question, we can search for the gravitational signal $h(t)$ in the detector output using theoretical models, or templates, of gravitational waves that are predicted by general relativity. These templates $h(t;\lambda^\mu)$ depend on a number of parameters λ^μ that describe the physics of the source, such as the masses of two compact objects, but also parameters related to the location such as the angles in the sky α and δ. By phase matching the template with the signal, we can assess if the detector output effectively contains a gravitational wave signal represented by the template $h(\lambda^\mu, t)$.

Even though the noise is reduced to a very small level, the amplitude of the gravitational wave signal $h(t)$ remains in most cases small compared to the level of the noise $n(t)$. This means that we need to use data analysis methods that give us the most efficient way to phase-match our theoretical templates $h(\lambda^\mu, t)$ with the noisy signal $s(t)$. An efficient solution for this problem is to use matched filtering, which is the optimal linear filter for weak signals buried in noise [55]. If we move our problem from the time domain to the Fourier domain, then matched filtering gives us an analytical expression for the filter $K(t)$ in the Fourier domain that is a combination of the Fourier transform of the theoretical templates $h(\lambda^\mu, t)$ and the one-sided noise power spectral density of the noise $S_n(f)$. All the details on this approach are given in the first section of this chapter where we first review Fourier analysis and then derived the optimal filter inferred by matched filtering.

Thanks to matched filtering, we have a way to search for the gravitational wave signals. Now, let us assume that we have detected a gravitational wave signal represented by a template and a set of parameters λ^μ_0, we are still left with a crucial aspect of parameter estimation. We want to evaluate what are the errors for the parameters λ^μ as inferred from our signal. An effective way to tackle this problem is to use Bayesian inference. We will describe in this chapter how this approach is applied, by first reviewing aspects of Bayesian analysis and then describe how to estimate the probability distributions of the parameters λ^μ.

4.1 Fourier analysis and matched filtering

4.1.1 Fourier analysis

In this section, we will review aspects of Fourier analysis in the continuous case. Since the detector output $s(t)$ will not be continuous but a collection of discrete samples, we will then provide the expressions of the Fourier transform in the discrete case and highlight some caveats related to signal sampling. Finally, we will give a short example to illustrate the usefulness of Fourier analysis.

4.1.1.1 Continuous Fourier analysis

If we consider a continuous signal, $h(t)$, we write the Fourier transform of $h(t)$ with the following convention

$$\tilde{h}(f) = \int_{-\infty}^{\infty} h(t) \mathrm{e}^{2\pi i f t} \mathrm{d}t. \tag{4.2}$$

The inverse Fourier transform is then expressed as,

$$h(t) = \int_{-\infty}^{\infty} \tilde{h}(f) \mathrm{e}^{-2\pi i f t} \mathrm{d}f. \tag{4.3}$$

We are often interested in quantifying the frequency content of the signal. To do this, we use the power spectrum of the signal,

$$\mid \tilde{h}(f) \mid^2 = \tilde{h}(f)\tilde{h}^*(f), \tag{4.4}$$

where $\tilde{h}^*(f)$ is the complex conjugate of $\tilde{h}(f)$. The power spectrum can be interpreted as the power content of the signal at a given frequency f. Parseval's theorem states that the total energy content of the signal is the same whether we compute it in the time or frequency domain,

$$\int_{\infty}^{\infty} \mid h(t) \mid^2 \mathrm{d}t = \int_{\infty}^{\infty} \mid \tilde{h}(f) \mid^2 \mathrm{d}f. \tag{4.5}$$

To have an indication of the similarity between two signals, we can compute the cross-correlation of these two signals. The cross-correlation $C(\tau)$ between two signals in the time domain, $h_1(t)$ and $h_2(t)$, for a time lag τ is given by

$$C(\tau) = \int_{-\infty}^{\infty} h_1(t) h_2(t+\tau) \mathrm{d}t. \tag{4.6}$$

The Fourier transform of the cross-correlation in Eq. (4.6) is simply the product of the two Fourier transforms of the signals,

$$\begin{aligned} \tilde{C}(f) &= \int_{-\infty}^{\infty} C(\tau) \mathrm{e}^{2\pi i f \tau} \mathrm{d}\tau, & (4.7) \\ &= \tilde{h}_1(f)\tilde{h}_2^*(f). & (4.8) \end{aligned}$$

With regards to the detector noise, we assume here that the noise is stationary and results from a Gaussian distribution with zero mean and variance σ^2. Under these assumptions, the noise can be described in terms of its one-sided noise power spectral density [56],

$$\langle \tilde{n}(f)\tilde{n}(f^{'}) \rangle = \frac{1}{2} S_n(f) \delta(f - f^{'}), \tag{4.9}$$

where $\langle . \rangle$ denotes an ensemble average and δ is the Dirac delta function. Since the noise is stationary, the ergodic principle states that the ensemble average can be computed by integrating over time for a single realisation of the noise. $S_n(f)$ can be seen as an extension of the notion of power spectrum for stochastic process. Finally, under the assumpation of Gaussianity and stationarity, the noise is fully described by [57]

$$\begin{aligned} \langle n(t) \rangle &= 0, & (4.10) \\ \langle n^2(t) \rangle &= \int_0^{\infty} S_n(f) \mathrm{d}f. & (4.11) \end{aligned}$$

In reality, the assumption of Gaussianity and stationarity break down with a real detector where we observe glitches in the data. In this case, we need to treat the data with more complex techniques but we assume in this work that the noise satisfies these conditions.

4.1.1.2 Discrete Fourier analysis and the sampling theorem

In reality, the detector measures a signal $h(t)$ with discrete samples $h_j = h(t_j)$. Every time sample is recorded with a sampling frequency f_s that defines a constant time interval between each sample

$\Delta t = 1/f_s$. If we consider a time series of N samples, h_j, with total observation time, $T_{obs} = N\Delta t$, the discrete Fourier transform of the series is expressed as,

$$\tilde{h}_k = \sum_{j=0}^{N-1} h_j \mathrm{e}^{-2\pi i j k/N}, \tag{4.12}$$

where the frequency associated with $\tilde{h}_k$ is $f_k = k/T_{obs}$.

Since we now approximate the continuous signals by a discrete version, we need to be certain that we do not lose information in the sampling process. The Nyquist-Shannon sampling theorem states that if a continuous signal contains no frequencies higher than f_{max}, then this signal can be completely reconstructed by its discrete version where the sampling frequency is at least $f_s = 2f_{max}$. In other words, for a fixed sampling frequency, the maximum frequency that can be properly reproduced by a sampled signal is given by the Nyquist frequency, $f_s/2$. Intuitively, this theorem can be understood by considering the fact that a discrete cosine function should have at least two samples per cycle in order to be properly represented.

We can illustrate the sampling theorem with an example, where we sample from a cosine function $h(t) = \cos(2\pi f_0 t)$ with frequency $f_0 = 5$ Hz. In Figure 4.1, we plot the time samples h_j and the power spectrum $\mid \tilde{h}_k \mid$ where $\tilde{h}_k$ are the Fourier components associated with the time samples h_j. We consider two different scenarios: in the first scenario we use $f_s = 10$ Hz, satisfying the Nyquist-Shannon sampling theorem (top row) and in the second scenario we undersample the original signal with $f_s = 8$ Hz (bottom row). When we look at the results of the first scenario, we see that the power spectrum indicates a dominant frequency at 5 Hz, that is the true frequency of the original signal f_0. In the second case where $f_s = 8$ Hz, we observe that the power spectrum now indicates a dominant frequency at 3 Hz. If we plot the resulting cosine associated with this frequency, we observe that it effectively encompasses all sample points (black crosses), even though they were sampled from a cosine with higher frequency f_0. This phenomenon is called aliasing and indicates that two different original signals become indistinguishable when they are sampled, in this case the two cosines with frequency 3 and 5 Hz when using $f_s = 8$ Hz.

In practice, this implies that the signal output from the gravitational wave detector has to be sampled with a sampling frequency that is higher than twice the value of the maximum frequency that can be measured given the detector noise. For the LIGO detectors, the sampling frequency is $f_s = 16384$ Hz [6] and for Virgo we have $f_s = 20000$ Hz [5].

4.1.1.3 Example of Fourier analysis

We present here an example that highlights the usefulness of Fourier analysis in our case. Let us consider the heuristic case where we consider a wave $h(t)$ that is a simple sine function $h(t) = A\sin(2\pi f_0 t)$ with amplitude $A = 10^{-21}$ and frequency $f_0 = 50$ Hz. Now let us assume that we measure a signal $s(t)$ that is a combination of $h(t)$ and noise affecting our measurement, $n(t)$. In this example, we consider a special case where the noise is the result of a white Gaussian process given by its constant one-sided noise power spectral density $S_n(f) = 10^{-43}\mathrm{Hz}^{-1}$.

On the left hand side of Figure 4.2, we present the time evolution of the three signals $s(t)$, $n(t)$ and $h(t)$ for a total observation time $T_{\mathrm{obs}} = 1$ s. Since the amplitude of the noise $n(t)$ is much larger than the amplitude of $h(t)$, we can not infer the presence of the wave $h(t)$ from our measurement $s(t)$ just by looking at the time domain data.

Now if we look at the problem in the frequency domain, by taking the Fourier transform of the three signals, $\tilde{h}(f)$, $\tilde{n}(f)$ and $\tilde{s}(f)$, we can compute the power spectrum of these signals as shown on the right hand side of Figure 4.2. For $|\tilde{h}(f)|$ we observe a single peak at frequency f_0 which is what we expect since this signal is a pure sine with frequency f_0. In the case of just noise, $|\tilde{n}(f)|$ is spread across the spectrum with an almost constant amplitude, as expected for a white Gaussian process. For the spectrum of the measured signal $s(t)$, we observe a peak at 50 Hz indicating that the buried signal $\tilde{h}(f)$ stands above the noise. And intuitively, that means that the frequency content of the signal $\tilde{h}(f)$ can be inferred just by looking at the spectrum of the measured signal $|\tilde{s}(f)|$.

4.1.2 Matched filtering

Matched filtering [55] is a data analysis technique that is well suited for the analysis of signals buried in noise. We will describe in this section how it can be applied in the case of GWs [56, 58, 59].

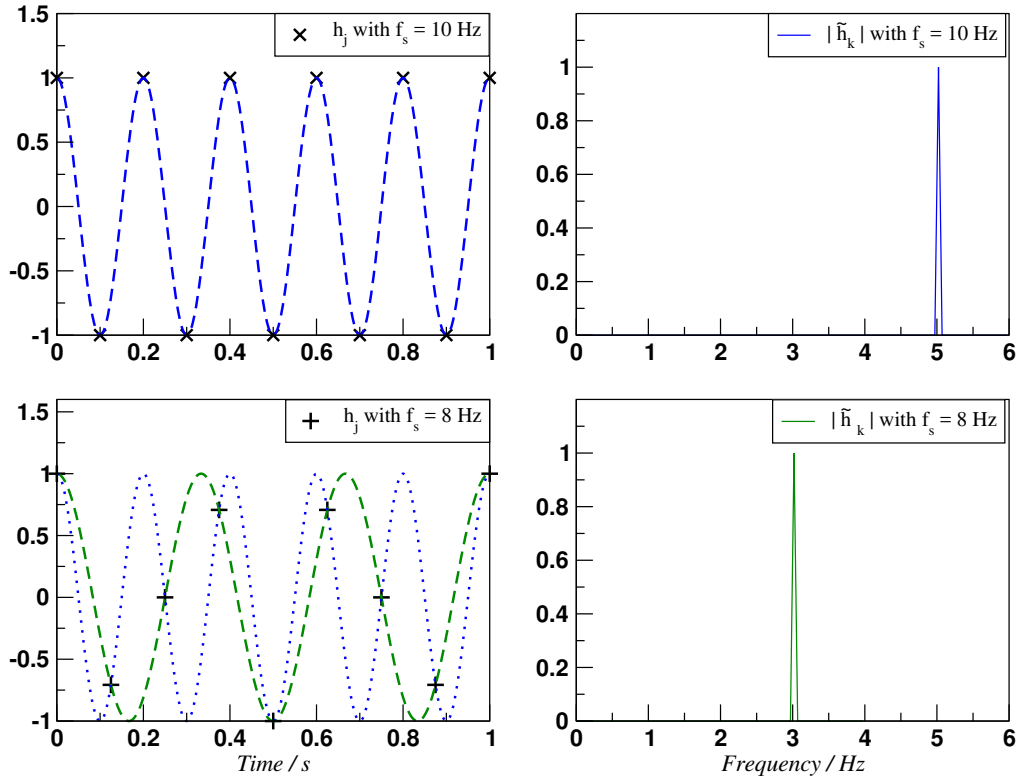

Figure 4.1: Illustration of the sampling theorem for a cosine signal $h(t) = \cos(2\pi f_0 t)$ with $f_0 = 5$ Hz. We give two practical cases: one where the sampling frequency is adapted, $f_s = 10$ Hz leading to good signal reproduction (top row) and the other where we under sample the original signal, $f_s = 8$ Hz, leading to aliasing (bottom row).

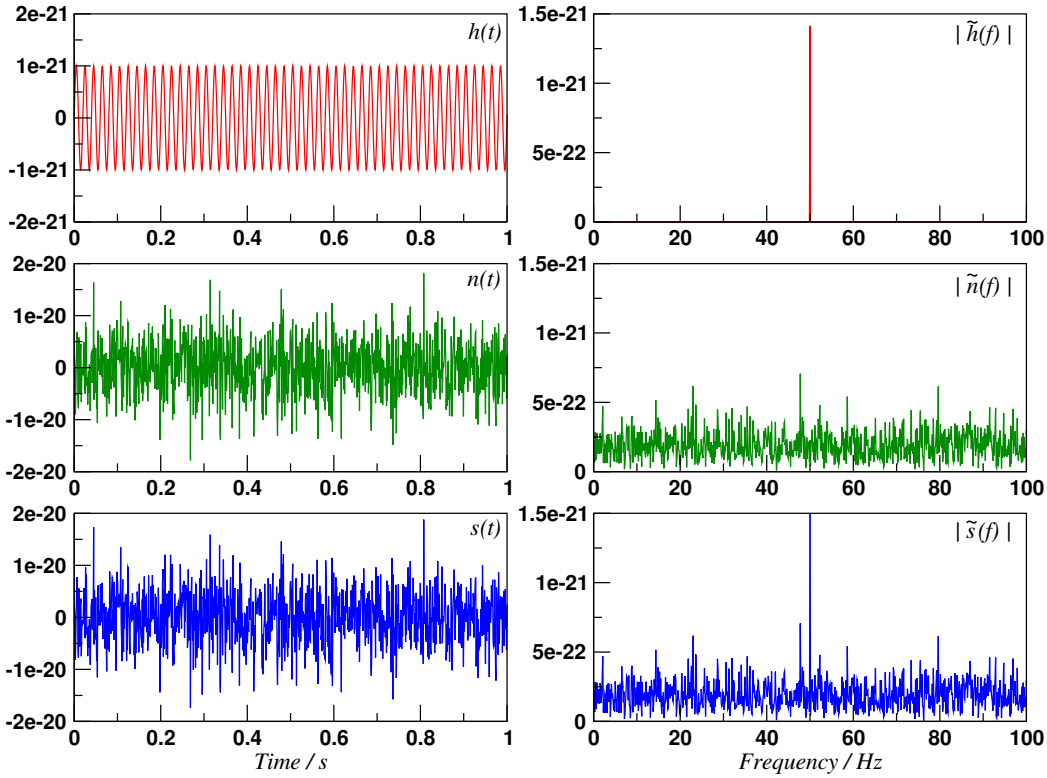

Figure 4.2: Illustration of the use of Fourier analysis to search for a heuristic wave $h(t)$ buried in the detector noise $n(t)$ of a measured signal $s(t)$.

Let us consider a filter $K(t)$ and compute the cross-correlation $C(\tau)$ between the output of the detector $s(t)$ and this filter for a time lag τ,

$$C(\tau) = \int_{-\infty}^{\infty} s(t)K(t+\tau)\mathrm{d}t, \tag{4.13}$$

$$= \int_{-\infty}^{\infty} \tilde{s}(f)\tilde{K}^*(f)\mathrm{e}^{-2\pi i f\tau}\mathrm{d}f, \tag{4.14}$$

where we used the Fourier transform of the cross-correlation from Eq. (4.8) in Eq. (4.14). The idea of matched filtering is to assess how well the filter matches the detector output statistically. To do that, we compute the signal-to-noise ratio, or SNR, ρ, as

$$\rho = \frac{S}{N}, \tag{4.15}$$

where S is the statistical mean of the cross-correlation $C(\tau)$ in the presence of the gravitational wave signal $h(t)$ and N is the statistical standard deviation of $C(\tau)$ in the absence of gravitational wave signal. For S, we have,

$$S = \langle C(\tau)\rangle = \int_{-\infty}^{\infty} \langle\tilde{s}(f)\rangle\tilde{K}^*(f)\mathrm{e}^{-2\pi i f\tau}\mathrm{d}f, \tag{4.16}$$

$$= \int_{-\infty}^{\infty} \tilde{h}(f)\tilde{K}^*(f)\mathrm{e}^{-2\pi i f\tau}\mathrm{d}f, \tag{4.17}$$

where we used the fact that the statistical mean of the noise is zero in Eq. (4.11). Now, for N, we have,

$$N^2 = \left[\langle C^2(\tau)\rangle - \langle C(\tau)\rangle^2\right]_{h=0}, \tag{4.18}$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \langle\tilde{s}(f)\tilde{s}^*(f^{'})\rangle\tilde{K}(f)\tilde{K}^*(f^{'})\mathrm{e}^{2\pi i\tau f}\mathrm{e}^{-2\pi i\tau f^{'}}\,\mathrm{d}f\mathrm{d}f^{'}, \tag{4.19}$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \langle\tilde{n}(f)\tilde{n}^*(f)\rangle\tilde{K}(f)\tilde{K}^*(f)\mathrm{e}^{2\pi i\tau(f-f^{'})}\mathrm{d}f\mathrm{d}f^{'}, \tag{4.20}$$

$$= \int_{-\infty}^{\infty} \frac{1}{2}S_n(f) \mid \tilde{K}(f) \mid^2 \mathrm{d}f, \tag{4.21}$$

where we used the fact that the variance of the detector output is equal to the variance of the noise in the absence of signal in Eq. (4.21) and the definition of the one-sided noise power spectral density from Eq. (4.9) in Eq. (4.21). Putting everything together we find that,

$$\rho = \frac{\displaystyle\int_{-\infty}^{\infty} \tilde{h}(f)\tilde{K}^*(f)\mathrm{e}^{-2\pi i f\tau}\mathrm{d}f}{\left[\displaystyle\int_{-\infty}^{\infty} \frac{1}{2}S_n(f) \mid \tilde{K}(f) \mid^2 \mathrm{d}f\right]^{1/2}} \tag{4.22}$$

Let us introduce the inner product [58] as,

$$\langle h|g\rangle = 2\int_0^{\infty} \frac{\tilde{h}(f)\tilde{g}^*(f) + \tilde{h}^*(f)\tilde{g}(f)}{S_n(f)}\mathrm{d}f, \tag{4.23}$$

and rewrite the signal to noise ratio as,

$$\rho = \frac{\left\langle \frac{1}{2}S_n(f)\tilde{K}(f)\mathrm{e}^{-2\pi i f\tau}\middle|\tilde{h}(f)\right\rangle}{\left\langle \frac{1}{2}S_n(f)\tilde{K}(f)\middle|\frac{1}{2}S_n(f)\tilde{K}(f)\right\rangle}. \tag{4.24}$$

We can say that $\tilde{K}(f)$ is the optimal filter if it maximises the signal to noise ratio in Eq. (4.24). In this case, the optimal filter is given by

$$\tilde{K}(f) = \frac{\tilde{h}(f)\mathrm{e}^{-2\pi i f\tau}}{S_n(f)}. \tag{4.25}$$

Note that we still have the presence of the delay that can be maximised on by choosing an appropriate delay τ. When doing so, the signal-to-noise ratio is given by,

$$\rho = \frac{\langle \tilde{s}(f) \mid \tilde{h}(f) \rangle}{\sqrt{\langle \tilde{h}(f) \mid \tilde{h}(f) \rangle}}, \tag{4.26}$$

and the optimal signal-to-noise ratio (i.e. when $s(t) = h(t)$) is

$$\rho_{opt} = \sqrt{\langle \tilde{h}(f) \mid \tilde{h}(f) \rangle}. \tag{4.27}$$

4.2 Detection and parameter inference

As we have seen in the previous section, the SNR is maximised when we use the optimal filter that is the true gravitational wave signal weighted by the noise. In reality we use general relativity to build gravitational waveforms templates $\tilde{h}(f; \lambda^\mu)$ that depend on a number of parameters λ^μ inherent from the model. Finding the best value for the SNR is then a question of finding the best set of parameters for which the template is as close as possible to the true gravitational wave signal. This raises the problem of how can we detect signals using a set of templates, such that we both have enough templates in the set to recover all the signals without missing them, and not too many templates to make the problem computationally affordable.

Taking the point of view of differential geometry , the set of parameters λ^μ defines a manifold on which the templates $\tilde{h}(\lambda^\mu, f)$ are considered as vectors [60]. The norm of a vector is then given in the usual way as,

$$\mid \tilde{h} \mid = \sqrt{\langle h \mid h \rangle}. \tag{4.28}$$

If we now consider two nearby vectors on the manifolds separated by $\Delta\lambda^\mu$, we can express the distance between these two points as,

$$ds^2 = \mid \widetilde{h}(f, \lambda^\mu + \Delta\lambda^\mu) - \widetilde{h}(f, \lambda^\mu) \mid^2 . \tag{4.29}$$

By expanding the quantity in the inner product and keeping only the first order terms, the previous equation can be rewritten as,

$$ds^2 = \Big| \frac{\partial \tilde{h}}{\partial \lambda^\mu} \Delta\lambda^\mu \Big|^2 = \Big\langle \frac{\partial \tilde{h}}{\partial \lambda^\mu} \Big| \frac{\partial h}{\partial \lambda^\nu} \Big\rangle \Delta\lambda^\mu \Delta\lambda^\nu. \tag{4.30}$$

The latter expression suggests that we introduce the following metric on our manifold

$$\Gamma_{\mu\nu} = \Big\langle \frac{\partial \tilde{h}}{\partial \lambda^\mu} \Big| \frac{\partial \tilde{h}}{\partial \lambda^\nu} \Big\rangle. \tag{4.31}$$

As required for a metric, this tensor is symmetric and is known as the Fisher Information matrix (FIM). We should highlight the fact that, as in general relativity, this metric is only local and depends on the point we are at on the manifold. If we now compute the inner product between two nearby templates separated by $\Delta\lambda^\mu$ and expand it at the quadratic order, we can compute the overlap between the two templates $\mathcal{O}$ [61],

$$\mathcal{O} = \Big\langle \tilde{h}(\lambda^\mu) \Big| \tilde{h}(\lambda^\mu + \Delta\lambda^\mu) \Big\rangle = 1 - \frac{1}{2} \Gamma_{\mu\nu} \Delta\lambda^\mu \Delta\lambda^\nu. \tag{4.32}$$

The reason why this complementary differential geometry approach is fruitful is because we have translated the original problem of maximising the signal-to-noise ratio into a distance minimisation problem between the true signal and a template. One possible solution to solve this problem is to use a grid of waveform templates, $h_i(\lambda_i^\mu)$, spread over the parameter space λ^μ. The number and layout of these templates is crucial since it will impact both the efficiency and the computational time needed for the search. A possible solution for the grid is to require that the overlap between two adjacent template of the grid should be superior to a given value. In some cases, the number of templates required to build a proper grid is not computationnaly affordable [62], and we then need to use other optimization algorithms to find the closest template as we will see later on in this work.

Let us now consider the problem from a statistical point of view where λ^μ are treated as random variables. In this framework, we can define the likelihood, $\mathcal{L}(\lambda^\mu)$, to be the probability of observing the

detector output $s(t)$ for a given set of parameters λ^μ. Under the assumptions that the noise is stationary and Gaussian, the likelihood is given by [58]

$$\mathcal{L}(\lambda^\mu) = \exp\left[-\frac{1}{2}\langle s - h(\lambda^\mu)|s - h(\lambda^\mu)\right], \tag{4.33}$$

where $\langle|\rangle$ is the inner product defined previously in Eq. (4.23). To derive the latest expression, recalling that the noise is Gaussian, the combination $s - h(\lambda^\mu)$ is a realisation of the noise if a gravitational wave signal is measured by the detector and is perfectly represented by the template $h(\lambda^\mu)$. If we expand the inner product and take the logarithm of the likelihood in Eq. (4.33), we obtain

$$\ln \mathcal{L}(\lambda^\mu) = \langle s \mid h(\lambda^\mu)\rangle - \frac{1}{2}\langle h(\lambda^\mu) \mid h(\lambda^\mu)\rangle - \frac{1}{2}\langle s \mid s\rangle. \tag{4.34}$$

The last term in the previous equation, $\langle s \mid s\rangle$, is a constant that does not depend on the template parameters. Since in most of our analysis we compute ratios of likelihood, we use instead without loss of generality the reduced log-likelihood,

$$\ln \mathcal{L}_R(\lambda^\mu) = \langle s \mid h(\lambda^\mu)\rangle - \frac{1}{2}\langle h(\lambda^\mu) \mid h(\lambda^\mu)\rangle \tag{4.35}$$

In the statistical framework, the Fisher Information matrix from Eq. (4.31) is defined in terms of the log-likelihood as,

$$\Gamma_{\mu\nu} = -\mathbb{E}\left[\frac{\partial^2 \ln \mathcal{L}}{\partial\lambda^\mu\lambda^\nu}\right] = \left\langle \frac{\partial\tilde{h}}{\partial\lambda^\mu}\middle|\frac{\partial\tilde{h}}{\partial\lambda^\nu}\right\rangle. \tag{4.36}$$

where $\mathbb{E}$ is an ensemble average over the signal realisations. If we now assume that we are in a situation where the signal to noise ratio ρ is high and assume that the distribution of the parameters is a multivariate Gaussian distribution around the maximum of the posterior distribution, then the inverse of the Fisher matrix gives us the variance-covariance of the Gaussian distribution of the parameters, $C_{\mu\nu} = \Gamma^{-1}_{\mu\nu}$, where the standard deviation is then $\sigma = \sqrt{C_{mu\mu}}$. Under these assumptions, the distribution of the parameter errors $\Delta\lambda^\mu$ is a Gaussian distribution given by,

$$p(\Delta\lambda^\mu) \approx A \exp\left[-\frac{1}{2}\Delta\lambda^\mu\Gamma_{\mu\nu}\Delta\lambda^\nu\right]. \tag{4.37}$$

where A is a normalisation constant.

The Fisher Information matrix is defined as the negative expectation value of the Hessian matrix of the log-likelihood. But we have seen previously that the Fisher Information matrix can be seen as the metric tensor on the parameter space. In Eq. (4.36), this dual interpretation between the previous differential geometry approach and the statistical approach is made clear by the fact that the Fisher matrix is connected to the curvature of the log-likelihood.

The Fisher matrix has been used in a number of gravitational wave data analysis studies as a tool for parameter estimation. The main advantage of this approach is the fast computation time. However, there are a number of caveats with this method that make it unusable in practical parameter estimation.

- The first difficulty is the interpretation of the Fisher matrix that is dependent on whether the problem is tackled from a Frequentist or Bayesian approach, as outlined in [63]. In our case, the Fisher Information matrix is seen from a Bayesian point of view and gives a measure of the uncertainty on the parameters. But the matrix does not use the actual data from the detector $s(t)$ and relies only on the models for the GW templates, and the one-sided noise spectral density.

- Moreover, as we have seen, the Fisher matrix assumes that the parameters have a Gaussian distribution around the peak of the likelihood. However, in gravitational wave data analysis we are dealing with multi-modal distributions that can not be approximated with Gaussian distributions. In Chapter 1, we have seen that the localisation of the source in the sky inferred by the beam pattern functions and the time delays produce two solutions in the sky with a network of three detectors. And even in the case where the distribution of the parameters is unimodal, we can have large deviations from Gaussianity.

- Another issue with the Fisher Information matrix comes from the fact that the FIM "assumes" that the manifold of the parameter is a subset of $\mathbb{R}^n$ where n is the dimension of the parameter space[64]. But we know that this is not the case because the sky angles and spins are defined on 2-spheres

S^2 and the manifold defined by the parameters has then a structure $S^2 \times \mathbb{R}^{n-2}$ (non-spinning) and $S^2 \times S^2 \times S^2 \times \mathbb{R}^{n-6}$ (spinning). Thus in some cases, the error estimates on the sky returned by the FIM can be larger than the 2 sphere.

- Another problems with the FIM comes from the fact that it is not gauge invariant. It assumes that the distribution of the parameters are Gaussian regardless of the parametrisation.

- Last but not least, in a number of situations, the matrix is found to be either singular or ill-conditioned, making the inversion process very difficult and untrustworthy. In addition to that, the condition number of the Fisher Information matrix is very sensitive to the parametrisation chosen λ^μ as detailed in [64]

In order to have reliable parameter estimation, it is then essential to use a full Bayesian analysis. Bayesian analysis require much more computation time and power, but produces more trustworthy results. In the next section, we will review some aspects of probability theory and introduce Bayesian analysis in the context of gravitational wave data analysis.

4.3 Bayesian analysis for parameter estimation

In the previous section, we have described the process to search for a gravitational wave signal buried in noise using gravitational wave templates, $h(\lambda^\mu)$, that depend on a set of parameters λ^μ. Ultimately the objective of parameter estimation is to use the measurements of the detector output, $s(t)$, to constrain the values of λ^μ. But before getting in the details of Bayesian inference, we review some aspects of probability theory

4.3.1 Probability theory

We consider that our set of N template parameters λ^μ are outcomes of the associated set of random variables Λ^μ. These random variables are fully characterised through their probability density or distribution. Let us consider for now a single random variable Λ^μ from the set with $\mu \in [1, N]$. The probability density $p(\lambda^\mu)$ is defined such that for that for any interval $I^\mu \in S^\mu$, where S^μ is the total range in which λ^μ takes values, we have

$$\mathbb{P}\left[\Lambda^\mu \in I^\mu\right] = \int_{I^\mu} p(\lambda^\mu) \mathrm{d}\lambda^\mu, \tag{4.38}$$

where $\mathbb{P}\left[\Lambda^\mu \in I^\mu\right]$ is the probability that the outcome of Λ^μ is in the interval I^μ. This probability density function needs to satisfy the normalisation condition,

$$\mathbb{P}\left[\Lambda^\mu \in S^\mu\right] = \int_{S^\mu} p(\lambda^\mu) \mathrm{d}\lambda^\mu = 1. \tag{4.39}$$

When doing parameter estimation, we assess the probability distribution not on a single random variable but for the whole set of parameters at the same time. The associated probability distribution is called the joint-distribution, $p(\lambda^1, ..., \lambda^N)$, and is a generalisation of Eq. (4.38) as,

$$\mathbb{P}\left[\Lambda^1 \in I^1, ..., \Lambda^N \in I^N\right] = \int_{I^1} ... \int_{I^N} p(\lambda^1, ..., \lambda^N) \mathrm{d}\lambda^1 ... \mathrm{d}\lambda^N \tag{4.40}$$

Even if we have access to the joint-distribution, we are most of the time interested in the individual probability densities for each parameter. The transition between the two probability densities is not trivial because we need to take into account the correlations between the parameters. One example are the angles in the sky that are correlated through the beam pattern functions $F^+(\theta, \phi, \psi)$ and $F^\times(\theta, \phi, \psi)$. In the context of probability, this correlation is defined with the use of conditional probability density. For a given parameter λ^μ, the conditional probability $p(\lambda^\mu | \lambda^1, ..., \lambda^{\mu-1}, \lambda^{\mu+1}, ..., \lambda^N)$ characterises the probability distribution for λ^μ given fixed values of all the other parameters. This conditional probability can be related to the joint distribution through the product rule,

$$p(\lambda^1, ..., \lambda^N) = p(\lambda^\mu | \lambda^1, ..., \lambda^{\mu-1}, \lambda^{\mu+1}, ..., \lambda^N) p(\lambda^1, ..., \lambda^{\mu-1}, \lambda^{\mu+1}, ..., \lambda^N). \tag{4.41}$$

Applying the normalisation of Eq. (4.39) and using the expression of conditional probabilities in Eq. (4.41), we can integrate the joint distribution other all the range of parameters except λ^μ to have access to the marginal distribution for λ^μ

$$p(\lambda^\mu) = \int_{S^1} ... \int_{S^{\mu-1}} \int_{S^{\mu+1}} ... \int_{S^N} p(\lambda^1, ..., \lambda^N) \mathrm{d}\lambda^1 ... \mathrm{d}\lambda^{\mu-1} \mathrm{d}\lambda^{\mu+1} ... \mathrm{d}\lambda^N. \tag{4.42}$$

4.3.2 Bayesian analysis for gravitational wave data analysis

Our objective is to derive an expression for the joint distribution of parameters in Eq. (4.40). The distribution depends on two things

- the output of the detector $s(t)$,
- the model M we use to compute the gravitational waveform template $h(\lambda^\mu)$ and the one-sided noise power spectral density $S_n(f)$.

As a consequence, the joint distribution is written in this context as $p(\lambda^\mu|s, M)$ and is referred to the posterior distribution. However, deriving a direct analytical expression of the posterior distribution written in this form is not an easy task. This is where Bayesian inference is extremely powerful because the posterior density can be defined by Bayes' theorem as,

$$p(\lambda^\mu|s, M) = \frac{p(s|\lambda^\mu, M)p(\lambda^\mu|M)}{p(s|M)}, \tag{4.43}$$

where $p(s|\lambda^\mu, M) = \mathcal{L}(\lambda^\mu)$ is the likelihood defined by Eq. (4.33), $p(\lambda^\mu|M)$ the prior distribution and $p(s|M)$ the evidence.

The prior density $\pi(\lambda^\mu) = p(\lambda^\mu|M)$ is a reflection of the a priori knowledge that we have on the template parameters. These can be astrophysically motivated or can be provided by results of other experiments conducted in the past. If we do not have any prior information or choose not to constrain the initial knowledge on the template parameters, we can use an uninformed flat or uniform prior, and the prior density is in this case only a constant value independent of λ^μ .

The evidence $p(s \mid M)$ is the marginal density obtained by integrating the likelihood multiplied by the prior over the entire parameter space,

$$p(s \mid M) = \int \mathcal{L}(\lambda^\mu)\pi(\lambda^\mu)\mathrm{d}\lambda^\mu. \tag{4.44}$$

Hence, it acts as a constant that scales the posterior density such that the probability sums to one. This number is of importance when testing different models M. However, we will not do Bayesian model selection in this work, and hence will not take into account the evidence by only considering the posterior distribution up to a multiplicative constant.

4.4 Description of Markov Chain Monte Carlo methods

From the expression of the log-likelihood in Eq. (4.33) and given a prior probability, we can effectively compute the value of the posterior distribution for a set of parameters λ^μ. But in the end, what we are interested in is to find the maximum of the posterior distribution along with the distribution of the parameters. Once again we could use a grid just as described in the previous section, but the number of posterior distribution evaluations needed for good accuracy on results might require too much computational time.

This issue is a generic problem that has been studied for a long time. One method that was developed to tackle these situations is the Monte Carlo method. The idea is to use a stochastic approach instead of a deterministic grid. These methods are very powerful and have been used in a number of applications. In the first section, we illustrate the concept of Monte Carlo method with a simple example to estimate the value of π. Then in a second section, we will present the rejection sampling algorithm that is an example of algorithm based on Monte Carlo. Finally, we will introduce some concepts of Markov Chain theory and see how we can use Markov Chains in Monte Carlo applications.

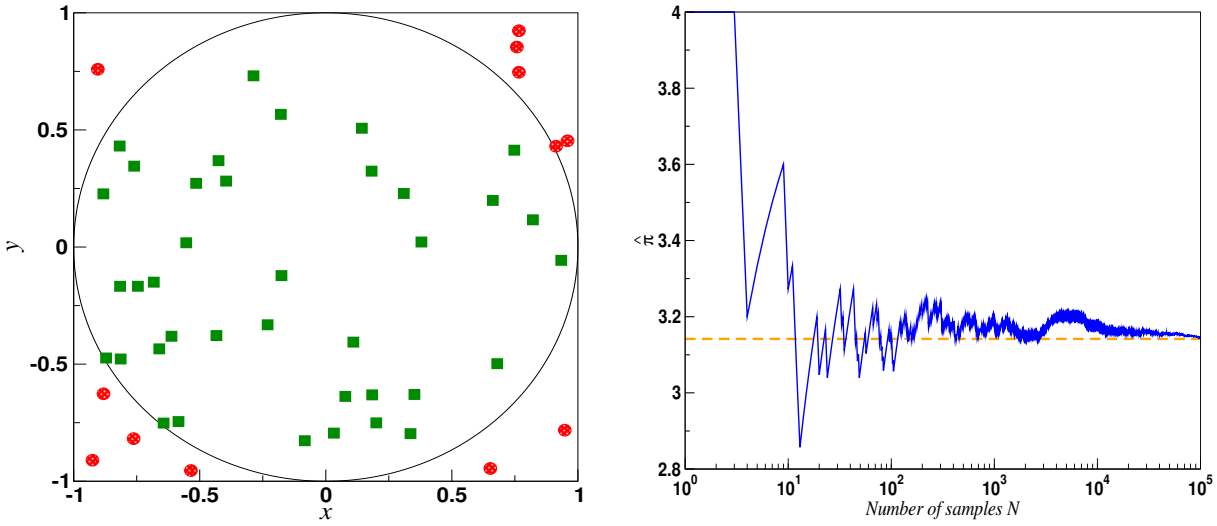

Figure 4.3: (left)Illustration of the Monte Carlo algorithm with $N = 50$ samples, and where 38 samples have been generated inside the circle (green squares) and 12 outside (red circles). The following value for the estimator of π is $\hat{\pi} = 3.04$. (right) Value of the estimator $\hat{\pi}$ as a function of the number of samples generated N plotted in solid line. The true value of π is indicated in dashed line

4.4.1 Monte Carlo principle

To illustrate how Monte Carlo methods work, let us say that we want to find a stochastic method to approximate the value of π. If we consider a unit circle, such that we have $x^2 + y^2 \leq 1$, and a square bounding the circle defined by $\{-1 \leq x \leq 1, -1 \leq y \leq 1\}$. If we define uniform random variables (X,Y) that take values inside the square as $X \sim \mathcal{U}[-1, 1]$ and $Y \sim \mathcal{U}[-1, 1]$, the probability that the point (x, y) generated by (X, Y) is inside the circle is given by,

$$\mathbb{P}((x, y) \text{ is inside the circle}) = \frac{\text{area of circle}}{\text{area of square}} = \frac{\pi}{4} \tag{4.45}$$

To estimate the probability in the previous equation, we can draw a number of N independent samples from the distributions of X and Y. If N_{int} points are generated inside the circle, then from Eq. (4.45) the quantity $\hat{\pi}$ is an estimator of π such that,

$$\hat{\pi} = 4 \times \frac{N_{int}}{N} \tag{4.46}$$

On the left hand side of Figure 4.3, we show an example of such an experiment with $N = 50$ samples and $N_{int} = 38$, giving an estimation $\hat{\pi} = 4{\times}38/50 = 3.04$. Since the samples are statistically independent, the law of large numbers tells us that the estimator $\hat{\pi}$ converges to the value of π for large N. Regarding the speed of convergence, the central limit theorem states that the speed of the convergence is of order $N^{-1/2}$. On the right hand side of Figure 4.3, we plot the value of the estimator of $\hat{\pi}$ as a function of the number of samples generated. We observe that the estimator converges towards the true value of π and provides a better estimation of π with 10^5 samples, $\hat{\pi} = 3.14568$. We highlight here once again the importance of the assumption that the samples are independent. As we will see in the next section, we will derive a similar theorem when the samples are not independent but form a Markov Chain.

4.4.2 Rejection sampling

Monte Carlo methods assure us that we can approximate the posterior distribution using a set of samples. Practically, the main difficulty is to find a method to generate a sample from the probability density. Since $p(\lambda^\mu)$ is a complicated function of λ^μ, there is no analytical method to directly generate samples from the posterior distribution. The main idea of rejection sampling [65] is to use another probability distribution $q(x)$ from which we know how to generate samples in order to build samples for $p(x)$. This additional distribution is what we call the proposal distribution.

We can describe in a general context how rejection sampling works. Given a random variable X with probability density $p(x)$ that we want to sample from, a proposal distribution with probability density

$q(x)$ from which we know how to sample and assuming that there is a real $M \in \mathbb{R}$ such that $f(x) < Mq(x)$ for all x, then we can generate a sample for $p(x)$ as follows

1. Draw a sample X_i from proposal distribution q,
2. Generate a uniform draw, $u \in \mathcal{U}[0, 1]$,
3. If $u < \dfrac{f(X_i)}{Mg(X_i)}$, accept the sample, otherwise reject it.

With this method we can generate samples from the distribution $p(x)$ even if $p(x)$ is only known up to a multiplicative constant [65]. An example of the rejection sampling is the example given in the previous section where the proposal distribution were the uniform samples $\mathcal{U}[-1, 1]$ bounding the inner circle.

The main drawback of this method is the fact that the value of M can be quite large in order to satisfy the condition that $f(x) < Mq(x)$, especially in high dimensions since M has to be set globally. And this high value of M can lead to large rejections for the samples since the probability of accepting a sample is proportional to,

$$\mathbb{P}(\text{X accepted}) \sim \frac{1}{M} \tag{4.47}$$

However, rejection sampling is a good illustration of the fact that we can sample from a probability distribution by using another proposal distribution. As we will see, this will be used in other more efficient Monte Carlo algorithms that are based on Markov Chains.

4.4.3 Markov Chain for Monte Carlo

In this section, we present the different properties of Markov Chains. We define the set of random variables X_t that take values in S, where S represents the parameter space. A Markov Chain $\{X_t, t \in T\}$, where t can be interpreted as fictitious time and $T \subset \mathbb{N}$, is a discrete stochastic process that satisfies the Markov Property

$$\mathbb{P}\left(X_t | X_0, ..., X_{t-1}\right) = \mathbb{P}\left(X_t | X_{t-1}\right). \tag{4.48}$$

In other words, the future state of the random process is only dependent on the current state and not the whole history of the Markov Chain. This is a very strong property from which we can derive the fact that we can describe the full Markov Chain by considering only the initial distribution for X_0 and the probability of transition from the current position X_t to the next state X_{t+1}. The latter probability is expressed in terms of a probability distribution called the transition kernel as,

$$\mathbb{P}\left[X_{t+1} | X_t = x_t\right] = \int K(x_t, x_{t+1}) \mathrm{d}x_{t+1}. \tag{4.49}$$

If now consider a given probability distribution f, we say that f is the invariant distribution of a Markov Chain with transition kernel K if it satisfies the condition that for all $y \in S$,

$$f(y) = \int_S f(x) K(x, y) \mathrm{d}x. \tag{4.50}$$

Intuitively the last equation tells us that if at some point a time realisation of the process is drawn according to the distribution f then the next realisation will also be drawn from the same distribution, and iteratively it will be the case for all future realisations. In other words, the transition probability K leaves the distribution f invariant. We say that a Markov kernel K is in detailed balance with a distribution f if for all x, y we have

$$f(x)K(x, y) = f(y)K(y, x). \tag{4.51}$$

If we have detailed balance between the kernel K and the distribution f, then one can demonstrate that f is the invariant distribution of the Markov Chain. This is one of the key property used to build algorithm based on Markov Chains. In fact, if we know the invariant distribution f, we have a way to characterize what is the long-term behavior of the Markov Chain. But to define a true convergence in distribution to f, we must require that the Markov Chain is both aperiodic and recurrent The aperiodicity states that the chain does not exhibit periodic behaviors where the same points in the parameter space are visited after a given period. The recurrence of a Markov Chain ensures that for an infinite time process, the Markov Chain explores the entire parameter space.

Under these properties, we say that the Markov chain is ergodic. The ergodic theorem states that an ergodic Markov Chain converges to its invariant distribution f when the chain is simulated with a large number of iterations. As we will see later on in this chapter, the number of iterations required for the convergence of the Markov Chain is a non trivial problem.

4.5 MCMC algorithms

In this section, we will present three examples of Markov Chain Monte Carlo algorithms: the Metropolis-Hastings algorithm, Differential Evolution (DE) and Differential Evolution Markov Chain (DEMC).

4.5.0.1 Metropolis-Hastings algorithm (MHMC)

As we have seen in Section 4.4.2, rejection sampling is limited because of the choice of global proposal distribution on the sample space. The idea of the Metropolis-Hastings algorithm is to use the structure of Markov Chain to build a rejection-like algorithm that takes advantage of local proposal updates with the transition kernel. This algorithm was first introduced by Metropolis [66] for a specific probability distribution and generalised for all distributions by Hastings [67]. In its initial formulation, the algorithm was designed to compute the equation of state of a substance composed of many particles whose behavior is given by the Boltzmann distribution. Since then the algorithm has been extended to a very large number of physical applications. In the field of gravitational waves, this method has been used in a number of studies for ground-based [68, 69, 70] and space-based [71, 72, 64, 73, 74, 75] observations. The reason for the popularity of this algorithm is its versatility and simple formulation for parameter estimation but also for searches.

The Metropolis-Hastings algorithm takes the ideas developed for rejection sampling and adapts them in the framework of Markov Chains. Here we will build a Markov Chain λ_t^μ in the template parameter space to sample from the posterior distribution $p(\lambda^\mu)$. This algorithm is the following 3-step process for a realisation t ,

1. Using a proposal distribution $q(.|\lambda_t^\mu)$ that depends on the current position, draw a test sample x^μ
2. Evaluate the Metropolis-Hasting ratio

$$\alpha(x^\mu|\lambda_t^\mu) = \min\Big\{1, \frac{\pi(x^\mu)\mathcal{L}(x^\mu)q(\lambda_t^\mu|x^\mu)}{\pi(\lambda_t^\mu)\mathcal{L}(\lambda_t^\mu)q(x^\mu|\lambda_t^\mu)}\Big\}, \tag{4.52}$$

3. Set $\lambda_{t+1}^\mu = x^\mu$ with probability α, otherwise set $\lambda_{t+1}^\mu = \lambda_t^\mu$

We are still left with choosing a proposal distribution $q(.|.)$ for the jump transition. Even though we do not have the constraints of rejection sampling on q, the efficiency and acceptance rate of the algorithm can greatly vary depending on the choice of q. A common choice is to use a random walk Metropolis-Hastings algorithm where we use multivariate Gaussian distribution for the jumps, i.e.

$$q(x|\lambda_t^\mu) = \lambda_t^\mu + \mathcal{N}(0, \Sigma). \tag{4.53}$$

But even in this case, we still need to define a good covariance matrix Σ to have an efficient algrorithm. Note that in this case, the proposal distribution is symmetric and the terms depending on the proposal distribution in the Metropolis-Hastings ratio from Eq. (4.52) cancel out, which is equivalent to the original Metropolis algorithm.

4.5.0.2 Differential evolution (DE)

Differential evolution (DE) was introduced by Storn and Price in 1995 [76]. Their goal was to find an algorithm capable of finding global extremum of complicated and multimodal functions.

The principle of the algorithm is to evolve a population of N_p candidate solutions, or parameter vectors, in the parameter space of a given $\mathcal{D}$-dimensional fitness function. In our case the candidate solutions are the template parameters and the fitness function is the likelihood function. The ensemble of the N_p parameters form a so-called generation G. Each member of this generation is evolved separately to create another generation $G+1$. The evolution process can be separated into three different steps: mutation, crossover and selection.

Given a candidate solution of generation G, $X^i(G)$, a first mutant vector $V^i(G+1)$ is created using three other distinct members of generation G, by adding the weighted difference between two of them to the third one

$$V^i(G+1) = X^j(G) + \gamma\left[X^k(G) - X^l(G)\right], \tag{4.54}$$

where $i \neq j \neq k \neq l \in [1, .., N_p]$. This means that a minimum of four particles is needed for DE to work. The differential weight $\gamma \in [0, 2]$ is a real, constant factor that controls the amplification of the

differential vector $\left[X^k(G) - X^l(g)\right]$. This step is called the mutation because features of three candidate solution are mixed together to create another candidate.

Using this mutant solution, a crossover solution, called a trial vector $V^i(G+1)$ is created by combining elements of the target solution $X^i(g)$, with the mutated solution $V^i(G+1)$, to produce a trial vector $U^i(g+1)$ as,

$$U_j^i(g+1) = \begin{cases} V_j^i(G+1) & \text{if } \alpha <= \text{CR} \quad \text{or } j = \beta \\ X_j^i(G) & \text{if } \alpha > \text{CR}, \end{cases} \tag{4.55}$$

where $j = 1..D$ stands for the parameter index, $\text{CR} \in [0, 1]$ is called the crossover constant, α a random number drawn uniformly between 0 and 1 and β is a randomly chosen index between 1 and D. The crossover constant controls the rate of crossover and must be selected before. Thus this crossover process also features elements of evolutionary algorithms that can be found in other algorithms such as a genetic algorithm.

Finally the trial vector is accepted with probability α, where α is the Metropolis-Hastings criterion comparing the fitness of the trial and target vectors. Strictly speaking, the Differential Evolution algorithm is not a Markov Chain algorithm. In fact, the Markovian property in Eq. (4.48) is not respected since we use various points from the history of the chain in Eq. (4.54). However, we can show that the differential evolution chain is asymptotically Markovian and can be used for parameter estimation purposes.

4.5.0.3 Differential evolution Markov Chain (DEMC)

In the Metropolis-Hastings algorithm described earlier, the acceptance rate of the jumps can be sometimes quite small (around 30%) and with poor mixing of the chain. One option used to increase the acceptance rate is to combine Differential Evolution with the Metropolis-Hastings method in an algorithm called Differential Evolution Monte Carlo (DEMC).

The idea is to change the jump transition between two adjacent points of the Markov Chain by a differential move transition. As explained before, Differential Evolution evolves a generation of particles to the next generation. However, in the case of Markov Chain, we only have one chain. That is why the concept of generation needs to be replaced by something else. In a first part of the algorithm, we run a usual Metropolis-Hastings algorithm but we keep every nth point of the chain in a vector $\overline{x}_i$. This trimmed history vector of the chain can then be used as the equivalent of a generation G. Thus the next point of the Markov Chain x_{i+1} can be built in a DE-like move as :

$$x_{i+1} = x_i + \gamma\left(\overline{x}_j - \overline{x}_k\right) \tag{4.56}$$

where i,j and k are mutually different and γ is the differential weight. The optimal value of this weight is given by $\gamma = 2.38/\sqrt{2D}$ where D is the dimension of the problem [77, 78]. This new proposed state of the chain can then be tested against the previous position with the probability α in Eq. (4.52) in order to decide if the jump is accepted or not. Once again, we do not construct a Markov Chain here since we introduce a history and do not respect the Markovian property as in Eq. (4.48). However, since we accumulate more and more points of the history of the chain to make the differential jumps and because we mix the DE steps with MHMC steps, the DEMC chain is asymptotically Markovian and converges to the target distribution.

4.6 Discrete statistical analysis of the posterior distribution

The MCMC algorithm produces a set of N samples λ_t^μ, where $t \in [1, N]$, representing the posterior distribution for the parameters λ^μ. We define in this section, some of the statistical quantities that we have used in this work to characterize the distribution of the template parameters. The sample mean $\overline{\lambda^\mu}$ is given by,

$$\overline{\lambda^\mu} = \frac{1}{N}\sum_{t=1}^{N} \lambda_t^\mu, \tag{4.57}$$

and the variance $\text{Var}(\lambda^\mu)$,

$$\text{Var}(\lambda^\mu) = \frac{1}{N}\sum_{t=1}^{N} \left(\lambda_t^\mu - \overline{\lambda^\mu}\right)^2. \tag{4.58}$$

If we take the square root of the variance, we obtain the standard deviation σ^μ,

$$\sigma^\mu = \sqrt{\mathrm{Var}(\lambda^\mu)}. \tag{4.59}$$

The skewness γ_1^μ is a measure of the asymmetry of the distribution of the parameter λ^μ around the mean $\overline{\lambda^\mu}$,

$$\gamma_1^\mu = \frac{1}{N}\sum_{t=1}^{N}\left(\frac{\lambda_t^\mu - \overline{\lambda^\mu}}{\sigma^\mu}\right)^3. \tag{4.60}$$

For a symmetric distribution around the mean, the skewness is zero, $\gamma_1^\mu = 0$. The kurtosis κ^μ is expressed as,

$$\kappa^\mu = \frac{1}{N}\sum_{t=1}^{N}\left(\frac{\lambda_t^\mu - \overline{\lambda^\mu}}{\sigma^\mu}\right)^4. \tag{4.61}$$

Practically, an excess of kurtosis will point towards some peculiar behavior in the distribution and can also be interpreted as a measure of the flatness or peakedness of the distribution. As a reference, we highlight that the kurtosis of a Gaussian distribution is equal to 3. Excess kurtosis of a distribution is then given by $\kappa_{ex} = \kappa - 3$, such that κ_{ex} for a Gaussian distribution is zero.

The mean and variance are useful but sometimes can be misleading especially when the distribution is asymmetric. This is the reason why, we often use the median of the distribution m^μ,

$$\mathbb{P}\left[\lambda^\mu \leq m^\mu\right] = \frac{1}{2}, \tag{4.62}$$

and the credible interval $\mathcal{C}^\mu$,

$$\int_{\mathcal{C}^\mu} p\left(\lambda^\mu|s\right) d\lambda^\mu = 1 - \alpha. \tag{4.63}$$

where for a 99% credible interval, $\alpha = 0.01$.

4.7 Convergence of the MCMC

For a chain produced by any of the MCMC algorithms described in the previous section, the convergence theorem assures us that we have a convergence to the invariant distribution for an infinitely long chain. In reality, we only run the code for a finite amount of time. In this case, we do not have certainty that the chain has converged. Knowing when to stop a Markov Chain Monte Carlo method is not a trivial problem and no definitive criterion exists to tell us when we have convergence.

To illustrate the problem of convergence of a MCMC algorithm, we take the same example that was presented in Section 4.1.1.3. We consider the heuristic wave model given by

$$h(A, f_0, t) = A\cos(2\pi f_0 t). \tag{4.64}$$

We assume that our data is a measured signal $s(t)$ that is the combination of a wave signal $h(t)$ with $f_0 = 50$ Hz and $A = 10^{-21}$ and noise that is the result of a white Gaussian noise process with constant one-sided noise power spectral density, $S_n(f) = 10^{-43}$.

We want to estimate the marginal posterior distribution for the parameters (A, f_0). To do that, we generate a Markov Chain X_i using the Metropolis-Hastings algorithm as defined in Section 4.5.0.1 with a total number of $N = 10^6$ iterations. For the parametrisation, we use $X = (\ln A, f_0)$, and for the proposal distribution we use a multivariate Gaussian distribution where the covariance matrix Σ is given by the diagonal matrix,

$$\Sigma = \begin{pmatrix} \sigma^2_{\ln A} & 0 \\ 0 & \sigma^2_{f_0} \end{pmatrix}. \tag{4.65}$$

By computing the inverse of the Fisher matrix $C_{\mu\nu} = \Gamma^{-1}_{\mu\nu}$ at the true signal position, we can have an estimation of the values of the variances as detailed in Section 4.2,

$$\begin{cases} \sigma^2_{\ln A} &= C_{00}, \\ \sigma^2_{f_0} &= C_{11}. \end{cases} \tag{4.66}$$

Once again, we highlight the fact that the Fisher matrix is only used to build the proposal distribution, and not to provide parameter estimation for A and f_0.

In order to have an idea of the convergence of the chain, we can first look at the value of the instantaneous mean of the chains. In Figure 4.4, we plot the instantaneous mean of the chains for A and f_0 as a function of iteration number for 10^6 iteration chain. Between 10 and 10^4 iterations of the chain, we see that the instantaneous mean has large oscillation. For more than 10^4 chain points, we observe that the mean starts to converge indicating that the chain is converging. Finally, the value of the instantaneous mean is almost constant after 10^5 chain points.

Another interesting feature that we can look at is the posterior distribution for A and f_0 using different numbers of chain points. In Figure 4.5, we plot the histograms of the distribution of A (top) and f_0 (bottom) for 10^3, 10^4, 10^5 and 10^6 chain points. In the case where we use 10^3 points for the distributions, we see that the posterior distribution is not well defined and present a large number of peaks. For 10^4 chain points, we see that some of the previous artifacts have disappeared and the distribution starts to being smooth. Finally for 10^5 chain points, the distribution is very smooth and we do not observe significant differences when we increase the number of points up to 10^6.

Finally, to test for convergence we also want to look at the auto-correlation function of the chain $\rho(\tau)$ at lag τ,

$$\rho(\tau) = \frac{\sum_{i=1}^{N-\tau} \left(X_i - \overline{X}\right)\left(X_{i+\tau} - \overline{X}\right)}{\sum_{i=1}^{N} \left(X_i - \overline{X}\right)^2}, \tag{4.67}$$

and the integrated autocorrelation length L (IAL) [79],

$$L = 1 + 2\sum_{\tau=1}^{\tau_{max}} \rho(\tau), \tag{4.68}$$

where τ_{max} is the maximum lag. To set a value for τ_{max}, we evaluate when the contribution of the autocorrelation is negligible and the autocorrelation becomes noisy. From the autocorrelation length, one can compute the number of statistically independent samples N_{ind} of the chain as,

$$N_{ind} = \left\lfloor \frac{N}{L} \right\rfloor \tag{4.69}$$

where $\left\lfloor . \right\rfloor$ represents the floor of the quantity. In Figure 4.6, we plot the value of the autocorrelation of the chain for A and f_0. We observe that in this example, the autocorrelation quickly decreases for both parameters, and for lags superior to 20, the autocorrelation becomes noisy and starts oscillating close to 0. If we compute the autocorrelation length and stop when the autocorrelation becomes noisy, we find that $L = 3$ for both parameters which then sets the number of statistically independent samples of the chain to be equal to $N_{ind} = 33333$.

We should highlight here that the previous example is a simple model with two uncorrelated parameters. In this case, as we have seen the MCMC algorithm quickly converges. For GW data analysis, we need to deal with problems that involve both higher dimensions and parameters that are highly correlated. In this case, the convergence of the MCMC is much slower and requires a much larger number of iterations.

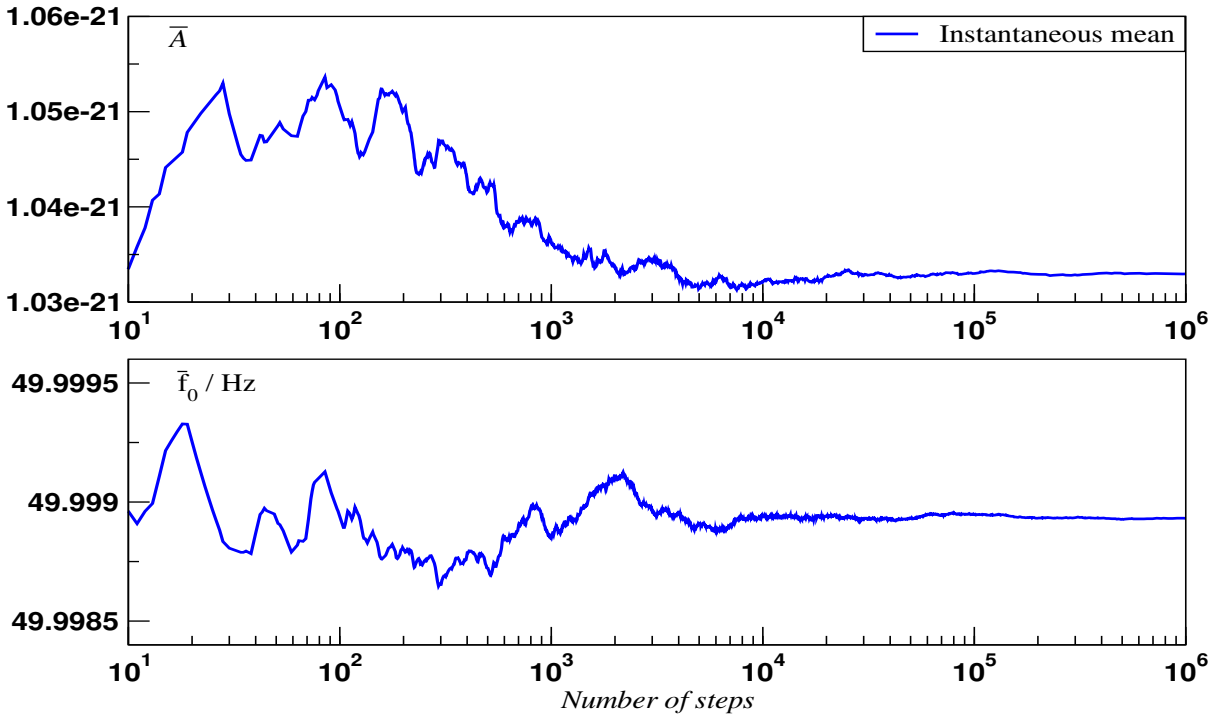

Figure 4.4: Value of the instantaneous mean $\bar{A}$ and $\bar{f}_0$ as a function of the number of chain points. The true values are not plotted on this graph.

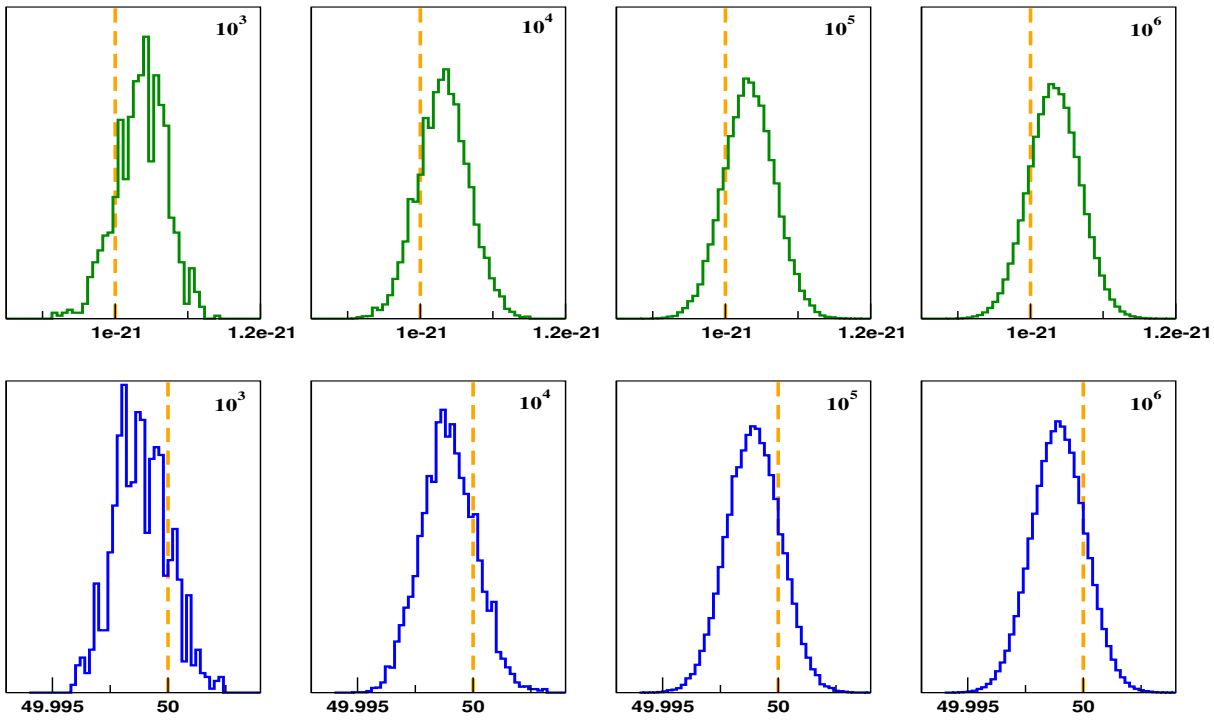

Figure 4.5: Posterior distribution for A (top) and f_0 (bottom) as inferred from the first 10^3, 10^4, 10^5 and 10^6 iterations of the chain. True values are represented by dashed lines.

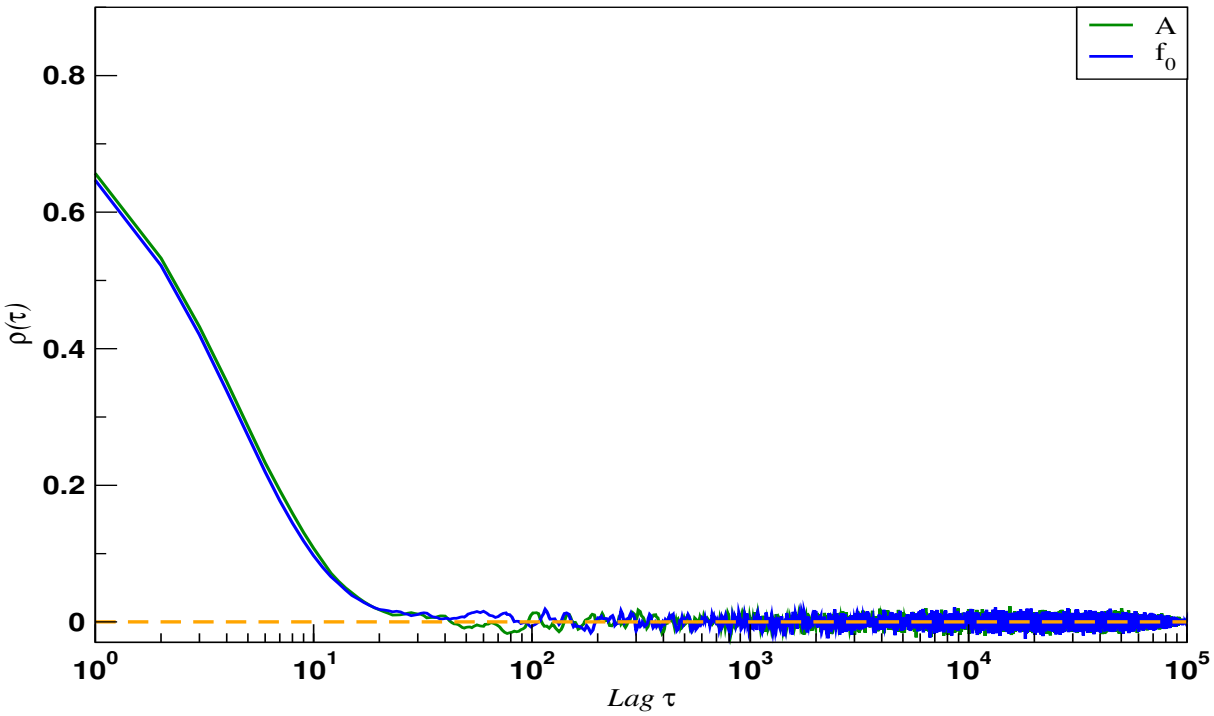

Figure 4.6: Autocorrelation function of the chain for A (top) and f_0 (bottom) as a function of the lag τ. Zero is represented by a dashed line and the autocorrelation drops to zero at lag $\tau = 15$.

Chapter 5

Gravitational wave model and detector network response

In this chapter, we will describe the form of the gravitational waves emitted during the coalescence of a binary formed of two compact objects. In the first section, we will describe the models used to compute the waveform in the source frame using the Post-Newtonian (PN) approximation. In the second section, we will detail how this GW waveform is observed by a network of detectors.

5.1 Introduction

We consider here a binary composed of two compact objects with masses m_1 and m_2, where we set $m_1 > m_2$ by convention. As it will be useful later on, we define also a number of parameters that are expressed as combination of these two masses ,

$$m = m_1 + m_2, \tag{5.1}$$

$$\mathcal{M}_c = \frac{(m_1 m_2)^{3/5}}{(m_1 + m_2)^{1/5}}, \tag{5.2}$$

$$\mu = \frac{m_1 m_2}{m_1 + m_2}, \tag{5.3}$$

$$\eta = \frac{m_1 m_2}{m_1 + m^2}, \tag{5.4}$$

where m is the total mass, $\mathcal{M}_c$ is the chirp mass, μ is the reduced mass and η is the symmetric mass ratio. The first three parameters have units of masses while the symmetric mass ratio is dimensionless and is bound in the interval $\eta \in (0, 1/4]$.

As we have seen in Chapter 1, the GW polarisations $h^+(t)$ and $h^\times(t)$ emitted by a binary system at luminosity distance D_L towards an observer can be written in the source frame as,

$$h^+(t) = A_+ \left(1 + \cos^2(\iota)\right) \cos(2\phi(t)), \tag{5.5}$$

$$h^\times(t) = A_\times \cos(\iota) \sin(2\phi(t)), \tag{5.6}$$

where $\phi(t)$ is the orbital phase, A^+ and $A^\times$ are the amplitudes of the polarisations and ι is the inclination angle defined to be the angle between the total angular momentum vector, $\hat{L}$, and a unit line of sight vector, $\hat{k}$, from the source to observer, i.e. $\cos\iota = \hat{L}.\hat{k}$. In the case where the orbit is circular, the orbital angular frequency is given by,

$$\omega_{orb}(t) = \frac{\mathrm{d}\phi(t)}{\mathrm{d}t}, \tag{5.7}$$

where ω_{orb} is related to the orbital separation r using Kepler's law as,

$$\omega_{orb}^2 = \frac{Gm}{r^3}. \tag{5.8}$$

As the orbital separation decreases, we see from Eq. (5.8) that the orbital frequency, and thus the gravitational waves frequency, increases. The coalescence of a compact binary can be split into three

different phases: inspiral, merger and ringdown. Since there is no analytical solution for the two-body problem in general relativity, the waveform associated with the coalescence needs to be computed either by using approximation techniques or by solving Einstein's equations numerically.

During the inspiral, the two compact objects orbit each other and the gravitational wave can be approximated with a post-Newtonian approximation, i.e. an expansion in powers of (v/c). As the distance between the two objects decreases through emission of GWs, the frequency of the gravitational waves increases in time. This increase in frequency is referred to as a chirp. The post-Newtonian approximation holds during the inspiral until the separation between the two objects reaches a limiting value. In the case of binary neutron star, the limit is the given by the last stable orbit r_{LSO},

$$r_{lso} = \frac{6Gm}{c^2}, \tag{5.9}$$

where m is in kg. The corresponding value for the gravitational wave frequency at the last stable orbit frequency f_{lso} is then,

$$f_{lso} = \frac{c^3}{6^{3/2}G\pi m}. \tag{5.10}$$

At this point, the two objects are so close that they start to interact and merge. In this phase, the dynamics are highly non-linear and we need to compute the gravitational waveform using numerical relativity simulations. At the end of the merger, the two compact objects merge into a single object. If the two objects are two white dwarfs, the result of the merger can be a neutron star. If we have two neutron stars, they can form an unstable hypermassive neutron star before collapsing to a black hole [80]. For systems composed of a neutron star and a black hole, or two black holes, the end product of the merger is a black hole.

In the case where the end product of the merger is a black hole, the black hole formed after the merger is initially highly distorded and is in an unstable phase. During this so-called ringdown phase, the black hole oscillates and emits gravitational waves through quasi normal modes. The frequency and amplitudes of these modes can be computed using black hole perturbation theory [81].

In this work, we have only used the gravitation waveform from the inspiral part of the coalescence where PN approximation is valid. This choice is motivated by the fact that the frequency at the last stable orbit is in the high frequency band of the current aLIGO and aVirgo, where the noise power spectral density is dominated by photon shot noise from the laser. As an example, the frequency of the last stable orbit for a typical binary neutron star with $m_1 = m_2 = 1.4M_\odot$ is, $f_{LSO} = 1570$ Hz. This means that the merger and ringdown part of the frequency will have a small contribution to the overall SNR of the signal. In addition, we do not have for the moment gravitational waveforms that take into account the internal equation of state of the neutron star and properly model the tidal interaction with the neutron stars. Thus using point-particle approximate for the waveform during the merger and the ringdown will be very inaccurate, and lead to incorrect results and conclusions.

In the next section, we will introduce the Taylor F2 waveform model that we have used to describe the gravitational waveform in the inspiral phase. This model describes the waveform directly in the Fourier domain, which is what we want for matched filtering approach. But before that, we will describe the Taylor T2 model that is the time model from which the Taylor F2 is derived.

5.2 Waveform models

The current collaboration aLIGO/aVirgo has developed a set of template waveforms included in the LAL library [68]. However, since we are using simple waveform and our study was an exploratory study, we developed the analysis outside of the current Ligo/Virgo collaboration codes.

As we said before, during the inspiral phase of the coalescence, orbital energy E is lost through an energy flux of gravitational waves $\mathcal{F}$, and the orbit is no longer completely circular. However, in the adiabatic limit, we can assume that the orbit evolves slowly in the sense that the fractional change of orbital velocity over an orbital period is small, $\dot{\omega}/\omega^2 \ll 1$. In this approximation, we can consider that the orbit is quasi-circular and can be describe by Kepler's equations at any orbital time t. In addition, we can also consider that the energy flux of gravitational wave balance the change in orbital energy averaged over one period. This is the energy balance equation that is written as,

$$\mathcal{F} = -m\frac{\mathrm{d}E}{\mathrm{d}t}, \tag{5.11}$$

The expressions for the orbital energy and energy flux of GWs have been derived in the post-Newtonian approximation. The post-Newtonian approximation is a perturbative expansion from Newton theory that is expressed using a small parameter that is often taken to be the characteristic velocity of the binary v,

$$v^3 = 2\pi f_{orb}\frac{mG}{c^3} = \pi f_{GW}\frac{mG}{c^3}. \tag{5.12}$$

where f_{orb} is the orbital frequency and f_{GW} is the GW frequency. The relevant quantities for the dynamics of the orbital motion are then approximated by sums that depend on the power of the expansion parameter v. The order of the PN expansion is denoted by half the last power of v in the sum, meaning that an expansion in v^7 is said to be a 3.5 PN approximation. We give here the expressions for the PN expansions for E at 3 PN order [82, 83, 84, 85, 86, 87],

$$\begin{aligned} E(v) &= -\frac{1}{2}\eta v^2\left[1-\left(\frac{3}{4}+\frac{1}{12}\eta\right)v^2-\left(\frac{27}{8}-\frac{19}{8}\eta+\frac{1}{24}\eta^2\right)v^4-\right. \\ &\left.\left\{\frac{675}{64}-\left(\frac{34445}{576}-\frac{205}{96}\pi^2\right)\eta+\frac{155}{96}\eta^2+\frac{35}{5184}\eta^3\right\}v^6\right], \end{aligned} \tag{5.13}$$

and $\mathcal{F}$ at 3.5 PN order [88, 89, 90, 91, 92],

$$\begin{aligned} \mathcal{F}(v) &= \frac{32}{5}\eta^2 v^{10}\left[1-\left(\frac{1247}{336}+\frac{35}{12}\eta\right)v^2+4\pi v^3-\left(\frac{44711}{9072}-\frac{9271}{504}\eta-\frac{65}{18}\eta^2\right)v^4-\left(\frac{8191}{672}+\frac{583}{24}\eta\right)\pi v^5\right. \\ &+\left\{\frac{6643739519}{69854400}+\frac{16}{3}\pi^2-\frac{1712}{105}\gamma+\left(\frac{41}{48}\pi^2-\frac{134543}{7776}\right)\eta-\frac{94403}{3024}\eta^2-\frac{775}{324}\eta^3-\frac{856}{105}\ln(16v^2)\right\}v^6 \\ &\left.-\left(\frac{16285}{504}-\frac{214745}{1728}\eta-\frac{193385}{3024}\eta^2\right)\pi v^7\right]. \end{aligned} \tag{5.14}$$

If we put everything together, the orbital dynamics of the system are described in terms of the expansion parameter v and the following set of differential equations,

$$\frac{\mathrm{d}\phi}{\mathrm{d}t}-\frac{v^3}{m} = 0, \tag{5.15}$$

$$\frac{\mathrm{d}v}{\mathrm{d}t}+\frac{\mathcal{F}}{mE'(v)} = 0, \tag{5.16}$$

where the first equation is Kepler's law from Eq. (5.8) and the second one is derived from the energy balance in Eq. (5.11). We can also express these equations in their integral form as,

$$t(v) = t_{ref}+m\int_v^{v_{ref}}\frac{E'(v)}{\mathcal{F}(v)}\mathrm{d}v, \tag{5.17}$$

$$\phi(v) = \phi_{ref}+\int_v^{v_{ref}}v^3\frac{E'(v)}{\mathcal{F}(v)}\mathrm{d}v, \tag{5.18}$$

where ϕ_{ref} and t_{ref} are integration constants and v_{ref} is an arbitrary reference velocity. We often set $\phi_{ref}=\phi_c$ and $t_{ref}=t_c$ where ϕ_c and t_c are the orbital phase and time at coalescence.

5.2.1 Taylor T2

The Taylor T2 model is based on the set of equations in their integrated form given in Eq. (5.17) and Eq. (5.18) [93, 94]. The idea is to express the ratio $\mathcal{F}(v)/E'(v)$ in the integrals as a Taylor series to express the orbital phase and orbital time. At the 3.5 PN order, they are expressed in terms of the characteristic velocity as [95],

$$\begin{aligned} \phi_{3.5}(v) &= \phi_c-\frac{1}{32\eta v^5}\left[1+\left(\frac{3715}{108}+\frac{55}{12}\eta\right)v^2-10\pi v^3+\left(\frac{15293365}{1016064}+\frac{27145}{1008}\eta+\frac{3085}{144}\eta^2\right)v^4\right. \\ &+\left(\frac{38645}{672}-\frac{65}{8}\eta\right)\ln\left(\frac{v}{v_{lso}}\right)+\left\{\frac{12348611926451}{18776862720}-\frac{160}{3}\pi^2-\frac{1712}{21}\gamma+\left(\frac{2255}{48}\pi^2-\frac{15737765635}{12192768}\right)\eta\right. \\ &\left.\left.+\frac{76055}{6912}\eta^2-\frac{127825}{5184}\eta^3-\frac{856}{21}\ln(16v^2)\right\}v^6+\left(\frac{77096675}{2032128}+\frac{378515}{12096}\eta-\frac{74045}{6048}\eta^2\right)\pi v^7\right], \end{aligned} \tag{5.19}$$

and,

$$t_{3.5}(v) = t_c - \frac{5m}{256\eta v^8}\left[1+\left(\frac{743}{252}+\frac{11}{3}\eta\right)v^2 - \frac{32}{5}\pi v^3 + \left(\frac{3058673}{508032}+\frac{5429}{504}\eta+\frac{617}{72}\eta^2\right)v^4 - \left(\frac{7729}{252}-\frac{13}{3}\eta\right)\pi v^5 + \left\{-\frac{10052469856691}{23471078400}+\frac{128}{3}\pi^2+\frac{6848}{105}\gamma+\left(\frac{3147553127}{3048192}-\frac{451}{12}\pi^2\right)\eta - \frac{15211}{1728}\eta^2+\frac{25565}{1296}\eta^3+\frac{3424}{105}\ln(16v^2)\right\}v^6 + \left(-\frac{15419335}{127008}-\frac{75703}{756}\eta+\frac{14809}{378}\eta^2\right)\pi v^7\right] \quad (5.20)$$

The main problem with the T2 model comes from the fact that we can compute $\phi(v)$ and $t(v)$, but we are interested in $\phi(t)$ to compute the gravitational waveform, and solving for $\phi(t)$ is quite expensive in terms of computation time.

5.2.2 Taylor F2

The Taylor F2 model is a PN approximant where the gravitation waveform is directly expressed in the Fourier domain [96]. To compute the Fourier transform, we can use the stationary phase approximation that allows us to write the waveform as,

$$\tilde{h}(f) = \frac{a(t_f)}{\sqrt{\dot{F}(t_f)}}\exp\left[\Psi(t_f)-\frac{\pi}{4}\right], \quad (5.21)$$

where f_{GW} is the gravitational wave frequency, $a(t_f)$ is related to the amplitude of the GW and Ψ is defined as,

$$\Psi(t_f) = 2\pi f t_f - 2\phi(t). \quad (5.22)$$

The stationary phase approximation tells us that the time t_f is the time when the frequency of the gravitational wave F is equal to the Fourier frequency f. As for the Taylor T2 model, we can use the integral expressions in Eq. (5.17) to express the value of $\psi(t_f)$ as,

$$\psi(t_f) = 2\pi f t_c - \phi_c + 2\int_v^{v_{ref}}(v_f^3-v^3)\frac{E'(v)}{\mathcal{F}(v)}\mathrm{d}v. \quad (5.23)$$

To solve the integrals, we can use the Taylor series expansion used in Taylor T2 for the ratio $\mathcal{F}(v)/E'(v)$. By doing so, we can rewrite the expression for the gravitational waveforms in the Fourier domain as measured by a detector as,

$$\tilde{h}(f) = \sqrt{\frac{5}{24}}\left(\frac{\mathcal{M}_c G}{c^3}\right)^{5/6}\frac{c}{D_L}\frac{\mathcal{Q}}{\pi^{2/3}}f^{-7/6}\mathrm{e}^{i\Psi_{3.5}(f)}, \quad (5.24)$$

where $\mathcal{Q}$ is a function depending on the detector and inclination of the source as,

$$\mathcal{Q} = \left[\left(\frac{1}{2}(1+\cos^2(\iota))F^+(\alpha,\delta,\psi)\right)^2 + \left(\cos(\iota)F^\times(\alpha,\delta,\psi)\right)^2\right]^{1/2}, \quad (5.25)$$

and Ψ is the gravitational waveform phase given in the Taylor F2 approximation by [95],

$$\Psi_{3.5}(f) = 2\pi f t_c - \phi_c - \pi/4 + \frac{3}{128\eta v^5}\left[1+\left(\frac{3715}{756}+\frac{55}{9}\eta\right)v^2 - 16\pi v^3 + \left(\frac{15293365}{508032}+\frac{27145}{504}\eta+\frac{3085}{72}\eta^2\right)v^4 + \pi\left(\frac{38645}{756}-\frac{65}{9}\right)\left\{1+3\log\left(\frac{v}{v_{lso}}\right)\right\}v^5 + \left\{\frac{1158323123653}{4694215680}-\frac{640}{3}\pi^2-\frac{6848}{21}(\gamma+\log(4v))+\left(-\frac{15737765635}{3048192}+\frac{2255}{12}\pi^2\right)\eta + \frac{76055}{1728}\eta^2-\frac{127825}{1296}\eta^3\right\}v^6 + \pi\left(\frac{77096675}{254016}+\frac{378515}{1512}\eta-\frac{74045}{756}\right)\right]. \quad (5.26)$$

In this study, we have decided to fix the value of t_c to the value of the chirp timeτ that corresponds to the time it takes for the binary to get from an initial frequency f_{low} to the end frequency of the inspiral

f_{lso}, and is expressed at 3.5 PN order as [97],

$$\begin{aligned}\tau_{3.5} &= \frac{5m}{256\eta v_{low}^8}\left[1+\left(\frac{743}{252}+\frac{11}{3}\eta\right)v_{low}^2-\frac{32}{5}\pi v_{low}^3+\left(\frac{3058673}{508032}+\frac{5429}{504}\eta+\frac{617}{72}\eta^2\right)v_{low}^4\right.\\ &+\left(\frac{13}{3}\eta-\frac{7729}{252}\right)\pi v_{low}^5+\left\{-\frac{10052469856691}{23471078400}+\frac{128}{3}\pi^2+\frac{6848}{105}(\gamma+\ln(4v))\right.\\ &+\left(-\frac{451}{12}\pi^2+\frac{3147553127}{3048192}\right)\eta-\frac{15211}{1728}\eta^2+\frac{25565}{1296}\eta^3\Big\}v_{low}^6\\ &\left.+\left(-\frac{15419335}{127008}-\frac{75703}{756}\eta+\frac{14809}{378}\eta^2\right)\pi v_{low}^7\right],\end{aligned} \tag{5.27}$$

where $v_{low}=\left(\pi f_{low} mG/c^3\right)^{1/3}$.

5.3 Detector Network Response to GWs

As we have seen, the response of a single detector to a GW is expressed as,

$$h(t)=h_+F^+ + h_\times F^\times \tag{5.28}$$

In this section we will describe how a network of detectors responds to a GW [98].

GW can be written in the TT gauge in the source frame coordinates $(\mathbf{X},\mathbf{Y},\mathbf{Z})$ as,

$$h_{ij}=h^+(\mathbf{e}_+)_{ij}+h^\times(\mathbf{e}_\times)_{ij}, \tag{5.29}$$

where we define the polarisation tensors $(\mathbf{e}_+)_{ij}$ and $(\mathbf{e}_\times)_{ij}$ as,

$$(\mathbf{e}_+)_{ij} = (\mathbf{X}\otimes\mathbf{X}-\mathbf{Y}\otimes\mathbf{Y})_{ij}, \tag{5.30}$$

$$(\mathbf{e}_\times)_{ij} = (\mathbf{X}\otimes\mathbf{Y}+\mathbf{Y}\otimes\mathbf{X})_{ij}. \tag{5.31}$$

where $\otimes$ is the tensor product. The vector $\mathbf{Z}$ of the source-frame is then taken such that $\mathbf{Z}=\mathbf{X}\wedge\mathbf{Y}$ points from the source towards the detector, where $\wedge$ is the vector product.

We now introduce a coordinate system fixed at the center of the earth that we express in terms of latitude and longitude (ϕ,λ), referenced from the prime meridian (Greenwich). The x-axis is taken such that it passes through the point $(\phi=0^\circ,\lambda=0^\circ)$, the y-axis through the point $(\phi=0^\circ,+\lambda=90^\circ)$ and the z-axis through $(\phi=+90^\circ,0^\circ)$. To locate the source in the sky, we use the spherical polar coordinates (θ,ϕ) that are measured with respect to the previous fixed frame. The relationship between these angles and the position of the source in terms of right ascension α and declination δ are

$$\alpha = \phi+\mathrm{GMST}, \tag{5.32}$$

$$\delta = \pi/2-\theta, \tag{5.33}$$

where GMST is the Greenwich mean sidereal time of arrival of the signal. In addition to that, we define the polarisation angle ψ to be the angle of rotation in the transverse plane from the X-axis in the source-frame to the x-axis in the Earth frame. We then have the following relationships,

$$\mathbf{X} = (\sin\phi\cos\psi-\sin\psi\cos\phi\cos\theta)\,\mathbf{i}-(\cos\phi\cos\psi+\sin\psi\sin\phi\cos\theta)\,\mathbf{j}+(\sin\psi\sin\theta)\,\mathbf{k}, \tag{5.34}$$

$$\mathbf{Y} = (-\sin\phi\sin\psi-\cos\psi\cos\phi\cos\theta)\,\mathbf{i}-(\cos\phi\sin\psi-\cos\psi\sin\phi\cos\theta)\,\mathbf{j}+(\cos\psi\sin\theta)\,\mathbf{k}, \tag{5.35}$$

where $\mathbf{i}$, $\mathbf{j}$ and $\mathbf{k}$ are unit vectors pointing in the (x,y,z) direction. Using the previous expression, we can then express the waveform in the Earth fixed frame of reference, and then project the GW onto the detector that is located on the surface of the Earth.

The response of an interferometer on the Earth's surface to an incoming gravitational wave can be expressed as,

$$D_{ij}=\frac{1}{2}\left(\mathbf{n}^x\otimes\mathbf{n}^x-\mathbf{n}^y\otimes\mathbf{n}^y\right)_{ij}, \tag{5.36}$$

where $\mathbf{n}^x$ and $\mathbf{n}^y$ are unit vectors pointing towards the x and y arms of the interferometer. The response to a gravitational wave signal by an interferometer A can then be expressed as,

$$h^A=\sum_{i,j=1}^{3}D_{ij}^A h_{ij}, \tag{5.37}$$

and the beam pattern functions F_+^A and $F_\times^A$ can then be found by comparing the previous expression with

$$h^A = F_+^A h^+ + F_\times^A h^\times. \tag{5.38}$$

In order to determine the tensor D_{ij}, we need to find a way to express the basis vectors of the interferometer in the Earth centered frame we have defined earlier. The position of the interferometer can be expressed in terms of the WGS-84 earth model [99], where the Earth is modeled as an oblate ellipsoid with semi-major axis $a = 6378137$ m and a semi-minor axis $b = 6356752.314$ m. In this model, the position of the detector in the Earth frame reference is written,

$$\mathbf{x} = x\mathbf{i} + y\mathbf{i} + y\mathbf{i} \tag{5.39}$$

where its coordinates are given in terms of longitude, latitude and evelation h,

$$\begin{aligned} x &= [R(\phi) + h]\cos\phi\cos\lambda, & (5.40)\\ y &= [R(\phi) + h]\cos\phi\sin\lambda, & (5.41)\\ z &= \left[\left(\frac{b^2}{a^2}\right)R(\phi) + h\right]\sin\phi. & (5.42) \end{aligned}$$

In the previous expression, $R(\phi)$ is the local radius of the Earth given by $R(\phi) = a^2(a^2\cos\phi + b^2\sin\phi)^{-1/2}$. We can then introduce the following basis for the detector at the position x,

$$\begin{aligned} \mathbf{e}_\lambda &= -\sin\lambda\mathbf{i} + \cos\lambda\mathbf{j}, & (5.43)\\ \mathbf{e}_\phi &= -\sin\phi\cos\lambda\mathbf{i} - \sin\phi\sin\lambda\mathbf{j} + \cos\phi\mathbf{k}, & (5.44)\\ \mathbf{e}_h &= \cos\phi\cos\lambda\mathbf{i} + \cos\phi\sin\lambda\mathbf{j} + \sin\phi\mathbf{k}, & (5.45) \end{aligned}$$

where $\mathbf{e}_\lambda$, $\mathbf{e}_\phi$, and $\mathbf{e}_h$ are respectively pointing East, North and up. In this basis, we can express the directions of the x and y arms $n^{x,y}$ using the angles $(\psi_{x,y}, \omega_{x,y})$ that act like azimuthal and polar angles in the bases previously defined,

$$\begin{aligned} n^x &= \cos\omega_x\cos\psi_x\mathbf{e}_\lambda + \cos\omega_x\sin\psi_x\mathbf{e}_\phi + \sin\omega_x\mathbf{e}_h, & (5.46)\\ n^y &= \cos\omega_y\cos\psi_y\mathbf{e}_\lambda + \cos\omega_y\sin\psi_y\mathbf{e}_\phi + \sin\omega_y\mathbf{e}_h. & (5.47) \end{aligned}$$

We have now everything we want to compute the expression of the detector tensor D_{ij} in the earth fixed frame reference and find the expressions for the beam pattern functions.

As GWs travel with the speed of light c, in terms of a detector network, this means that the GW will arrive at different times at the various detectors depending on the sky localisation of the source. We can define the time delays between two detectors A and B, $\Delta_{A/B}$, where we use B as the reference detector as,

$$\Delta_{A/B} = a_1\cos(GMST - \alpha)\cos\delta + a_2\sin(GMST - \alpha)\cos\delta + a_3\sin\delta \tag{5.48}$$

where a_1, a_2 and a_3 depend on the positions of the detectors A and B.

These time delays will induce a change in the phase of the waveform. If we consider a three detector network using aLIGO and aVirgo, we can take one detector as a reference, for instance aLIGO at Hanford, and write the expressions of the phase of the waveforms detected by each detectors as,

$$\begin{aligned} \Psi_H &= \Psi & (5.49)\\ \Psi_L &= \Psi - 2\pi f\Delta_{L/H} & (5.50)\\ \Psi_V &= \Psi - 2\pi f\Delta_{V/H} & (5.51) \end{aligned}$$

where $\Delta_{L/H}$ is the time delay of arrival at Livingston, $\Delta_{V/H}$ is the time delay of arrival at Virgo and Ψ_H, Ψ_H, Ψ_V are the phases of the waveforms measured respectively at Hanford, Livingston and Virgo.

From the point of view of data analysis, we can write the expression of the signal to noise ratio ρ^d and reduced log-likelihood $\ln\mathcal{L}^d$ for each detector d as,

$$\begin{aligned} \rho_d &= \frac{\langle s^d \mid h^d\rangle}{\sqrt{\langle h^d \mid h^d\rangle}}, & (5.52)\\ (\ln\mathcal{L})_d &= \langle s^d \mid h^d\rangle - \frac{1}{2}\langle h^d \mid h^d, \rangle & (5.53) \end{aligned}$$

where s^d is the signal measured at the detector d and h^d is the template used at the detector. Since we have a coherent detection by the network, we can express the network SNR and log-likelihood for a network of N detectors as,

$$\rho_{net} = \sqrt{\sum_{d=1}^{N} (\rho^d)^2}, \tag{5.54}$$

$$(\ln \mathcal{L})_{net} = \sum_{d=1}^{N} (\ln \mathcal{L})_d \,. \tag{5.55}$$

Chapter 6

Differential Evolution Monte Carlo for BNS sources

6.1 Introduction

Binary neutron stars (BNS) are expected to be a major source of GWs for ground-based detectors. However, due to uncertainties in their formation mechanisms (e.g. common envelope interaction, natal kicks from supernovae, etc.), the event rate for BNSs is still quite uncertain. While no BNSs were detected during aLIGO's first science run, a 90% upper credible interval of $\leq 12,000\,Gpc^{-3}yr^{-1}$ was placed on the event rate for BNSs [100]. Even if this is still quite large, this is already an order of magnitude improvement over previous rate estimates from initial LIGO/Virgo [101]. While not enough information was gathered during the first science run to rule out astrophysical models of BNS formation, a detection (or indeed, non-detection) of a BNS would begin to constrain astrophysical models.

The goal of this study is the development of a Bayesian inference algorithm for binary neutron star (BNS) systems. These particular sources were chosen for a number of different reasons. Firstly, as we do not expect to see the merger-ringdown of these systems, we can use a simplified waveform model, such as the TaylorF2 waveform described in Chapter 5. Secondly, and with an eye of future 3G GW parameter estimation, BNS systems are long lived sources in the detector. The BBH sources that have been detected in aLIGO's O1 and O2 science runs all lasted between 200 ms and 1.6 seconds in the detector [102, 11, 12]. On the other hand, and assuming a lower frequency cutoff for the detector of 40 Hz, a $(1.4, 1.4)\,M_{\odot}$ system will take $\sim$25 seconds to reach the last stable orbit frequency. If 3G detectors manage to descend to a low frequency cutoff of say 5 Hz, this timescale increases to 1.75 hours. The answer to how parameter estimation can be carried out for such sources is not trivial [103]. Thirdly, our primary goal here is to develop an algorithm for the LVC-O3 science run, which should begin in late fall, 2018. It is unclear what the lower frequency cutoff for the detectors will be for this run, but it will have an increase in the chirptime for the source (for reference, the chirptime will be 55 seconds for a low frequency cutoff of 30 Hz, and 2.6 minutes for 20Hz).

For our test sources, we took a sample of BNS systems from the study on the first two years of electromagnetic follow-up with advanced LIGO and Virgo[1] [104]. We required our test sources to have a network SNR greater than a threshold of $\rho = 8$. We should again point out that due to the fact that we are carrying out a feasibility study, and due to the simplicity of the TaylorF2 waveform, all of our codes were developed outside of the LALInference framework [68]. As a consequence, any run-times quoted below assume constant sampling rates for the waveform generation. LALInference provides much faster ways of likelihood calculation, so the reader should take our run-times as a worst case scenario, and keep in mind that the important quantity will be the factor of acceleration in the convergence, rather than the actual clock time.

[1] http://www.ligo.org/scientists/first2years/

6.2 DEMC for BNS parameter estimation.

6.2.1 Parameterisation of the search space.

While the parameters of the GW template are given in Chapter 5, we run our algorithm on a parameter space with a different parameterisation. As with any study in physics, the choice of coordinate systems is crucial to the success of the study. In running a Markov chain, the choice of parameterisation of the parameter space can make the difference between a chain with a high acceptance rate, and one that never moves.

One of the most important choices to be made regards the choice of masses. While the two individual masses (m_1, m_2) are astrophysically the most interesting quantities, a parameter space defined by these two quantities is highly degenerate as any point in this space has a 1-2 mapping (i.e. a system with $(m_1, m_2) = (1.4, 10)\, M_\odot$ will produce exactly the same waveform as a system with $(m_1, m_2) = (10, 1.4)\, M_\odot$). There are other combinations of mass that can be used instead of the individual masses, for example, the total mass $m = m_1 + m_2$ and the symmetric mass ratio $\eta = m_1 m_2/m^2$. While the total mass would work perfectly well, the symmetric mass ratio has a physical cutoff at $\eta = 1/4$, when $m_1 = m_2$. This places an unphysical boundary on the search space that our algorithm is unaware of, and would hence need special treatment. Instead of η, one could also use the mass ratio $q = m_2/m_1$. While this also has a natural boundary of $q = 1$ for equal mass systems, it is easier to deal with. In past studies, the choice of chirp-mass $\mathcal{M}_c = m\eta^{3/5}$ and reduced mass $\mu = m\eta$ has proved to be an adequate choice for our combination of mass coordinate [105, 106].

As was seen in Chapter 4, our DEMC will evaluate the Metropolis-Hastings ratio to evaluate each proposal in parameter space. While we will discuss our choice of parameter priors a little later, we should point out that we can simplify our choice of priors by making certain parameter choices in the beginning. As we will not know how our inclinations are distributed, and as we expect our sources to be isotropically distributed over the sky, it makes sense here to choose $\cos\iota$ and $\sin\theta$ as two coordinates in parameter space, rather than ι and θ.

As we will see below, we can increase the efficiency of the chain mixing by moving in eigendirections rather than coordinate directions. One way of doing this is to use the local metric (i.e. the FIM) in defining the directionality and scale of the jump proposals. Due to the large dynamical ranges involved (i.e. some of our parameters have scales on the order of unity, while D_L, for example, has a scale on the order of 10^{24}), we can ensure that our FIM is numerically stable by using $\{\ln D_L, \ln \mathcal{M}_c, \ln\mu, \ln t_c\}$ instead of $\{D_L, \mathcal{M}_c, \mu, t_c\}$. This also ensures that we add equal weight to each decade in the parameter range. Putting this all together, we define our parameter space coordinates as

$$\lambda^\mu = \{\cos\iota, \phi_c, \psi, \ln D_L, \ln\mathcal{M}_c, \ln\mu, \sin\theta, \phi, \ln t_c\}. \tag{6.1}$$

6.2.2 Range of parameter priors.

For many of our prior distributions, we choose uninformative (flat) priors. We choose our mass priors such that they are flat in $(\ln\mathcal{M}_c, \ln\mu)$ corresponding to individual masses in the range $m_i \in [1, 2.3]\, M_\odot$. Our distance prior is flat in $\ln D_L$ corresponding to $D_L \in [10^{-6}, 200]$ Mpc, where the lower bound corresponds to the distance to M31, and the upper bound is defined by the design sensitivity BNS range of aLIGO. For the chirp-time, our prior is flat in $\ln t_c$, such that we have $t_c \in [t_c - 5, t_c + 5]$ seconds. Finally, for the priors in our angular parameters, we choose flat priors in $(\cos\iota, \sin\theta) \in [-1, 1]$, $(\phi, \varphi_c) \in [0, 2\pi]$ and $\psi \in [0, \pi]$.

6.2.3 Run setup.

Our runs use a three detector network based on the two aLIGOs and aVirgo (HLV). For our signal, we inject a TaylorF2 waveform into stationary, Gaussian noise. All of our chains are run for 10^6 iterations. While we know that this is not long enough for full parameter estimation, the goal here is to do an apples-to-apples comparison with the Hamiltonian Monte Carlo chain. Our runs also start from the true answer, as we are not interested in conducting a search phase, and assume the source has been already found. As an aside, in our algorithm we use simulated annealing to get the chain moving. We have run tests where we have started the chains at a distance from the true solution, and have always converged to the correct solution before the end of the simulated annealing phase. The main stages of the algorithm are as follows :

BNS	C#	ι/deg	ϕ_c/deg	ψ/deg	D_L/Mpc	$m_1/M_\odot$	$m_2/M_\odot$	α / deg	δ / deg	t_c / secs	ρ_{HLV}
1	5384	46	105	315	43	1.23	1.21	216.4	-77.7	31.92	49.01
2	699	40	333	108	41	1.34	1.23	223.9	51.3	29.36	86.01
3	899	26	139	118	84	1.36	1.25	99.9	-30.8	28.58	51.77
4	1135	149	162	342	57	1.43	1.24	168.9	9.0	27.61	34.04
5	1281	38	324	254	72	1.43	1.20	64.9	42.2	28.38	23.74
6	2608	153	305	215	46	1.36	1.35	106.1	16.7	26.80	64.95
7	2704	34	201	289	87	1.32	1.30	345.2	58.7	28.35	34.73
8	3015	176	327	115	68	1.43	1.30	277.7	-19.8	26.53	50.05
9	3123	155	81	307	77	1.46	1.23	121.0	70.4	27.33	39.44
10	3249	145	110	141	83	1.31	1.31	77.8	-25.9	28.35	50.28

Table 6.1: Source information for the BNS test-sources. The BNS number will be used as our source reference during this thesis, while $C\#$ is the reference number of each source in the 2 year EM follow-up study. The SNR, ρ_{HLV}, is the three detector network SNR retrieved using a template with the true parameter values.

- Calculate the FIM, $\Gamma_{\mu\nu}$, and from that, the eigenvalues E_μ and eigenvectors $V_{\mu\nu}$ of $\Gamma_{\mu\nu}$. We construct multi-variate jumps that use a product of normal distributions in each eigendirection of $\Gamma_{\mu\nu}$. The standard deviation in each eigendirection is given by $\sigma_\mu = 1/\sqrt{DE_\mu}$, where D is the dimensionality of the search space (in this case $D = 9$) and the factor of $1/\sqrt{D}$ ensures an average jump of $\sim 1\sigma$. In general, this means that we take jumps in the parameters that have the form

$$x^\mu \to x^\mu + A^\mu \sqrt{\Delta x^\mu}. \tag{6.2}$$

  Here, A^μ are scaling amplitudes given by

$$A_\mu = \frac{\mathcal{N}(0,1)}{\sqrt{DE_\mu}}, \tag{6.3}$$

  and the small displacement in the eigendirection of each parameter, Δx^μ, are given by

$$\Delta x^\mu = \sum_{\nu=1}^{D} \frac{V_{\mu\nu}^2}{E_\nu}. \tag{6.4}$$

  Our transition kernel $q(.|.)$ then has the form

$$q(.|.) = \prod_{\mu=1}^{D} \sqrt{DE_\mu} \exp\left(-\frac{D}{2} E_\mu A_\mu A_\mu\right). \tag{6.5}$$

- We choose to run a burn-in phase of 10^5 iterations, where we use simulated annealing [72, 107] to ensure that the chain moves from its starting point, and mixes well. Simulated annealing is an efficient mechanism for ensuring widespread exploration of a parameter space. By heating the likelihood surface, one can lower and fatten high peaks on the likelihood surface. In Figure 6.1, we plot the simplified example of a1D reduced likelihood as a function of m_1, keeping all other parameters constant. The solid (blue) curve represents the reduced likelihood for an annealing temperature of $T = 1$. We can see that as the waveform goes in and out of phase with the signal, we obtain secondary peaks around the main mode (It was shown in [72] that in a D-dimensional space, we actually have island chains of secondary solutions). We can see that at each side of the main peak, we have secondary solutions with reduced likelihoods of ~ 200. It is possible that a normal Markov chain could spend the majority of its runtime investigating one of these peaks rather than the main solution. We also plot the same likelihood with temperatures of $T = 5$ (orange-dashed) and $T = 10$ (magenta-dot-dashed). We can see that not only does a higher temperature reduce the amplitude of the peaks, it also fattens them such that it is easier to walk across the likelihood surface. In this way, we can approach the main peak in an accelerated fashion. A problem with simulated annealing is knowing how high the initial temperature should be. If it is too high, it flattens all features on the surface and the chain random walks around parameter space, wasting computational cycles. If is not high enough, we risk getting the chain stuck on a secondary solution.

  Another reason for using simulated annealing is that we will use accepted points in parameter space to construct a history for the DE moves. To ensure the efficiency of the chain, this means that we

have to make sure that samples from all modes are included in the history. Simulated annealing ensures that the chain will explore most of the given parameter space. To ensure that our initial temperature is high enough for exploration, but not too high, we ran some preliminary runs, tying the initial temperature to the SNR of a potential signal. We settled on an initial temperature of $T_{ini} = 50$, and set a power law dependent annealing scheme as [107]

$$T = 10^{T_0(1-i/T_s)}, \tag{6.6}$$

where $T_0 = \log_{10}(T_{ini}) = \log_{10}(50)$ is the heat index, i is the chain iteration number, and T_s is the cooling schedule which is set to 10^5 iterations.

The main effect of the annealing is that the eigenvalues of the FIM are rescaled according to

$$E_\mu^T = E_\mu/T, \tag{6.7}$$

which means that the amplitudes of our jumps, defined above, now become

$$A_\mu^T = \frac{\mathcal{N}(0,1)}{\sqrt{DE_\mu^T}} = \mathcal{N}(0,1)\sqrt{\frac{T}{DE_\mu}}. \tag{6.8}$$

This also means that our multivariate proposal distribution changes to

$$\begin{aligned} q(.|.) &= \prod_{\mu=1}^{D} \sqrt{\frac{DE_\mu}{T}} \exp\left(-\frac{D}{2T} E_\mu^T A_\mu^T A_\mu^T\right), & (6.9)\\ &= \prod_{\mu=1}^{D} \sqrt{\frac{DE_\mu}{T}} \exp\left(-\frac{D}{2T} \frac{E_\mu}{T} \mathcal{N}^2(0,1) \frac{T}{DE_\mu}\right), & (6.10)\\ &= \prod_{\mu=1}^{D} \sqrt{\frac{DE_\mu}{T}} \exp\left(-\frac{1}{2T} \mathcal{N}^2(0,1)\right). & (6.11) \end{aligned}$$

We further split this phase into two parts : in the first 5×10^4 iterations, we use only the MHMC proposals, where every accepted proposal is recorded to form a history for future DE moves. Furthermore, to ensure that we are moving between modes of the solution, every 10^3 iterations, we draw a random value from a uniform distribution $\beta \in \mathcal{U}[0,1]$. If $\beta \geq 0.5$, we propose a mode-hop of the form $(\iota \to \pi - \iota, \psi \to \pi - \psi)$. In the final 5×10^4 iterations of the burn-in, we use two DE proposals for every MHMC proposal, again recording every accepted point in parameter space for future DE proposals. The final proposal used in this phase is, every 10^2 DE proposals, we set the factor $\gamma = 1$. This move can also encourage the chain to move between modes of the solution.

- Once the heat has dropped to $T = 1$, we continue with the combination of DE-MHMC proposals as described above, as well as the mode-hop proposals.

6.3 Results

Before presenting the results of our DEMC chains, we should highlight that there is no single criterion for testing the convergence of a Markov chain. Some statistical methods do exist, but many of them require the running on multiple chains []. One of the most powerful methods of testing for convergence is still carrying out a visual inspection of the chains. Below, we present a number of different visual tests, that while still not conclusive, help with analysing the performance of the chains.

6.3.1 Convergence of the DEMC chains.

While we present results for all ten BNS sources in Appendices A and C, we will focus our attention on two test systems in particular : BNS1 and BNS5. BNS1 is composed of two low mass NSs, making it the longest lasting source in our sample, lasting ~ 31 seconds in the detector band, and is almost an equal mass system ($\eta = 0.24998$). BNS5 has the largest mass ratio in the sample($\eta = 0.2481$), and last approximately 28 seconds in the detector.

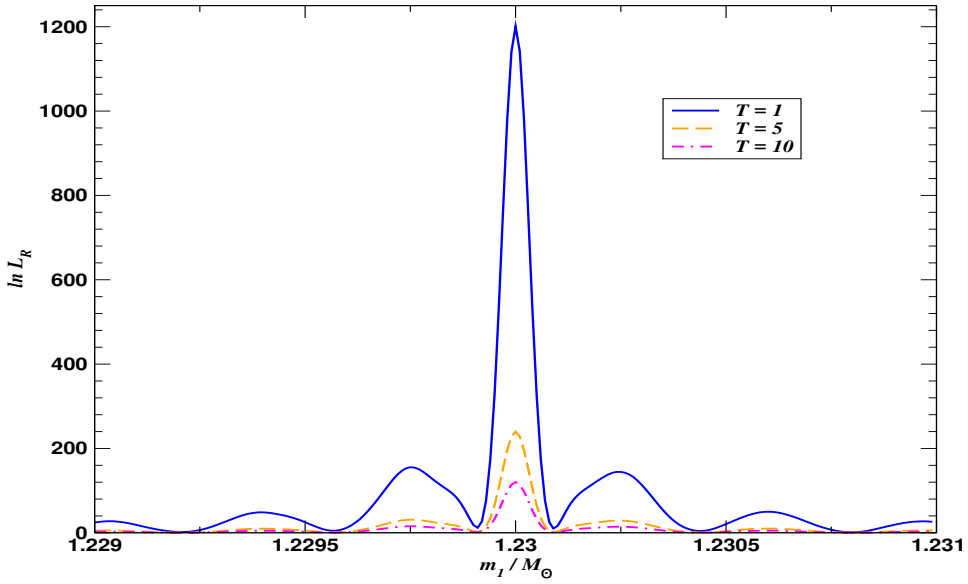

Figure 6.1: The evolution of the reduced log-likelihood, $\ln L_R$, as a function of simulated annealing temperature, T, for a simple 1D problem where all waveform parameters except m_1 are kept constant.

6.3.1.1 Exploration of the posterior distribution

One way of examining the performance of the DEMC chain is to look at its exploration in the difficult parts of parameter space. We know for BNS sources that the solution is bimodal in the (i, D_L) plane, multimodal in the ψ direction, and unimodal in all other coordinates. As a consequence, in Figures 6.2 and 6.3, we plot snapshots of the DEMC exploration of the posterior distribution in the (ι, D_L) plane, at instances of $10^2, 10^3, 10^4, 10^5$ and 10^6 iterations, for BNS1 and BNS5 respectively. In the case of BNS1, we see that up to 10^3 iterations, as expected for this type of algorithm, we have very localised exploration of the posterior. At 10^4 iterations, we observe that we are now beginning to explore both modes of the solution. This trend continues, and we can see at 10^6 iterations that we are exploring both branches of the posterior, but are only just beginning to explore the bridge between the modes. For BNS5, we observe a similar pattern of behaviour, where we only begin to explore the second mode after a few thousand iterations. While the bridge has been completed between the two modes by 10^6 iterations, it is unclear at this point whether or not we have full exploration of the posterior.

6.3.1.2 Convergence of the marginalised posterior distribution

Another quantity that can be examined is the convergence of the marginalised posterior distribution, as a function of iteration number. In Figures 6.4 and 6.5, we plot the posterior distribution for BNS1 and BNS5, based on 10^3 (orange), 10^4 (red), 10^5 (blue) and 10^6 (black) iterations for the set of parameters $\{\iota, D_L, \mathcal{M}_c, \mu, \theta, \phi\}$. We ignore, for now, the convergence of (ϕ_c, ψ) as they are not of specific astrophysical interest, and t_c as it is measured so accurately that the spread in the posterior distribution is tight to begin with.

If we first focus on BNS1. As expected, with only 10^3 samples, the distribution is highly peaked, and in the case of inclination, is concentrated on one mode of the solution. At 10^4 iterations, we can see that the distributions still display a high level of peakedness for all parameters, with suggestions of bi-modality in a number of parameters. However, it is only from 10^5 iterations onwards that we observe the first real signs that we are beginning to converge to the target distribution for $\{\mathcal{M}_c, \mu, \theta, \phi\}$ (as when compared to the posterior using 10^6 iterations). For $\{\iota, D_L\}$, we see that our distribution is still very peaked, and also still quite shifted from the distribution obtained with the 10^6 samples. It is clear that the exploration in these two parameters is slow compared to the others, thus leading to slow convergence of the marginalised posterior.

For BNS5, we see a slightly different situation. In this case, there is no clear convergence for any of the parameters. If we focus on the sky position, for example. For BNS1, there is not much visual difference between the posterior distributions at 10^5 and 10^6 iterations. For BNS5, however, the distributions are quite different. This does not give us very much confidence that we have converged to the target

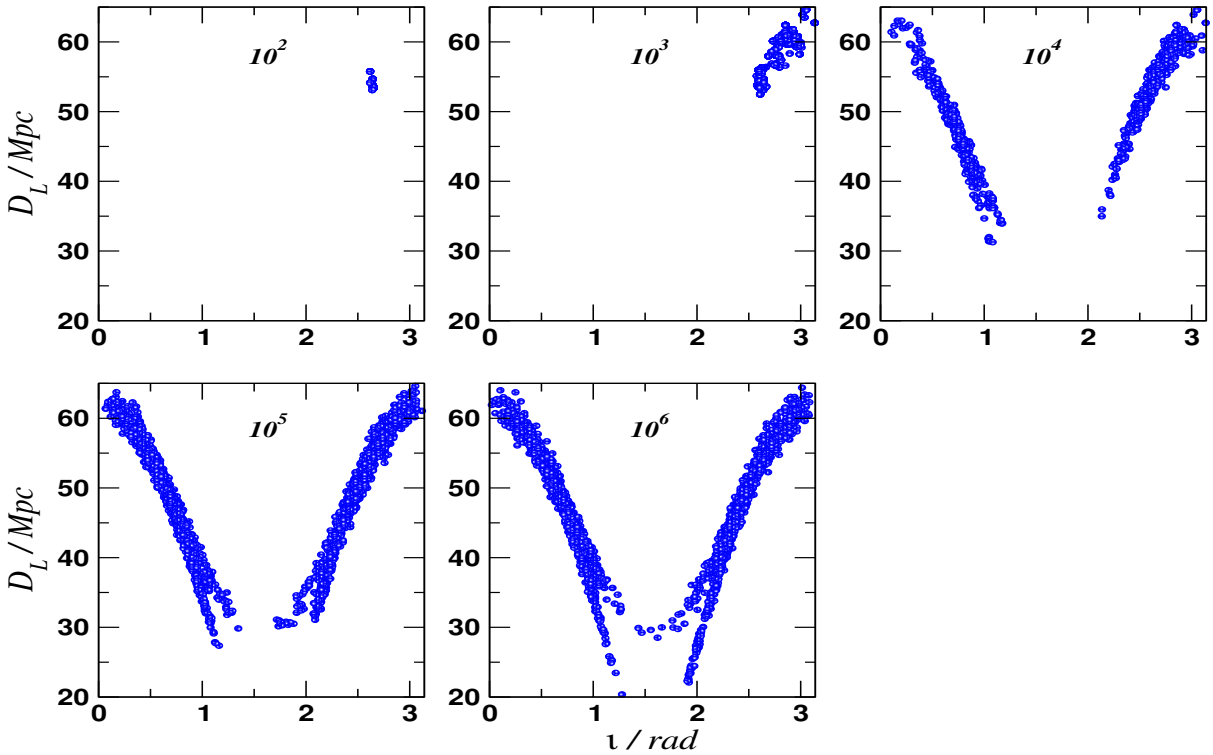

Figure 6.2: Snapshots of the exploration of the 2D (ι, D_L) posterior distribution for BNS1 using a DEMC chain in order of magnitude steps for $10^2 - 10^6$ iterations.

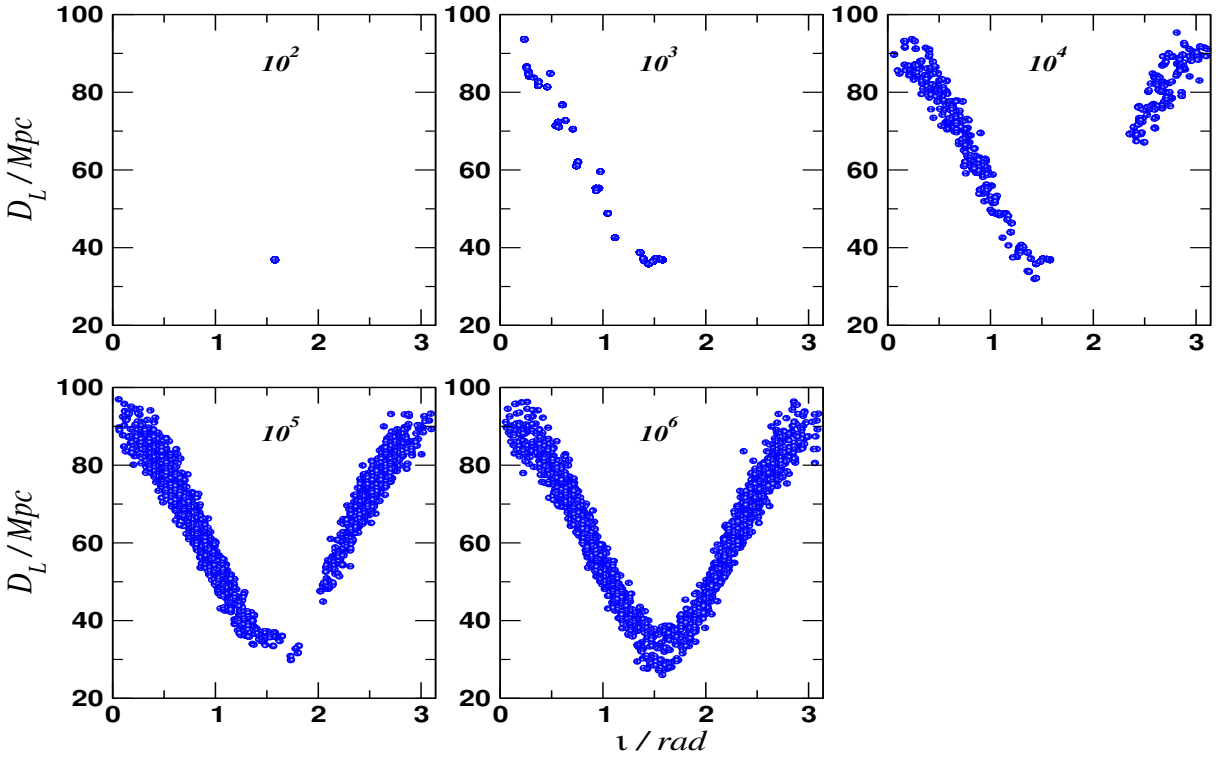

Figure 6.3: Snapshots of the exploration of the 2D (ι, D_L) posterior distribution for BNS5 using a DEMC chain in order of magnitude steps for $10^2 - 10^6$ iterations.

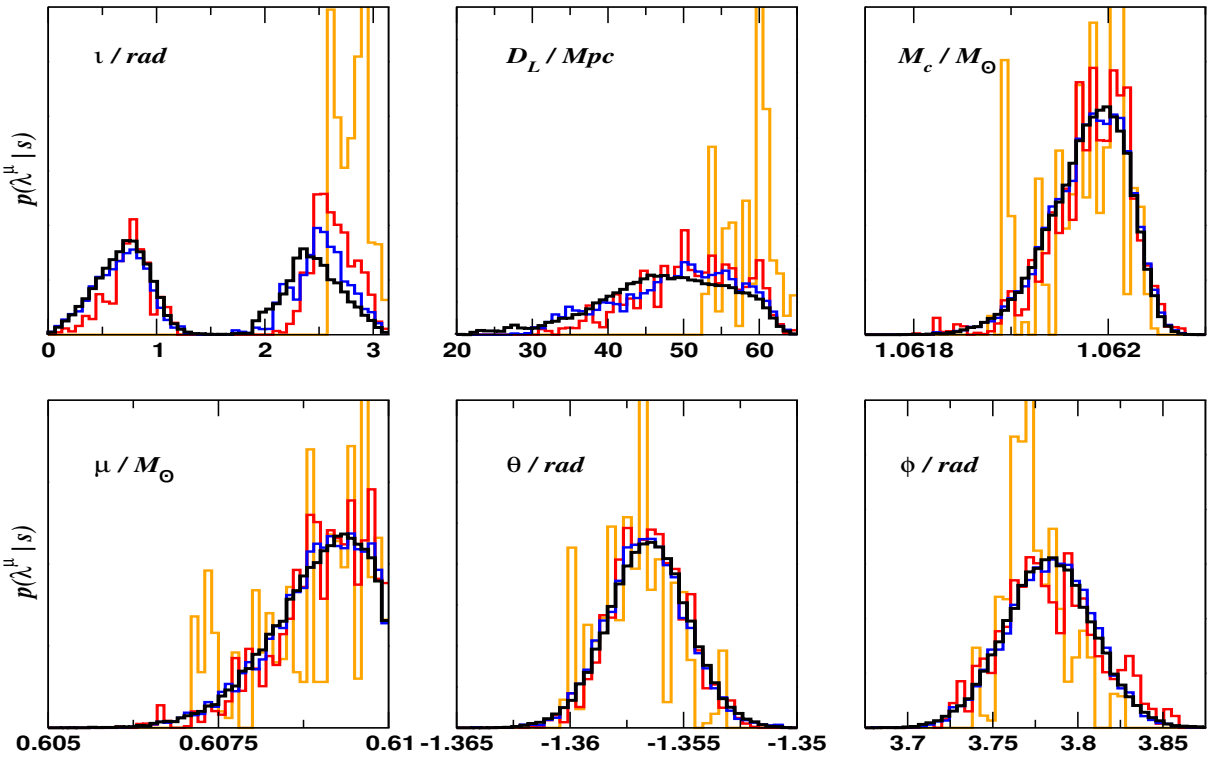

Figure 6.4: Evolution of the marginalised posterior distribution for BNS1, using a DEMC chain for 10^3 (orange), 10^4 (red), 10^5 (blue) and 10^6 (black) iterations, for the parameters $\{\iota, D_L, M_c, \mu, \theta, \phi\}$.

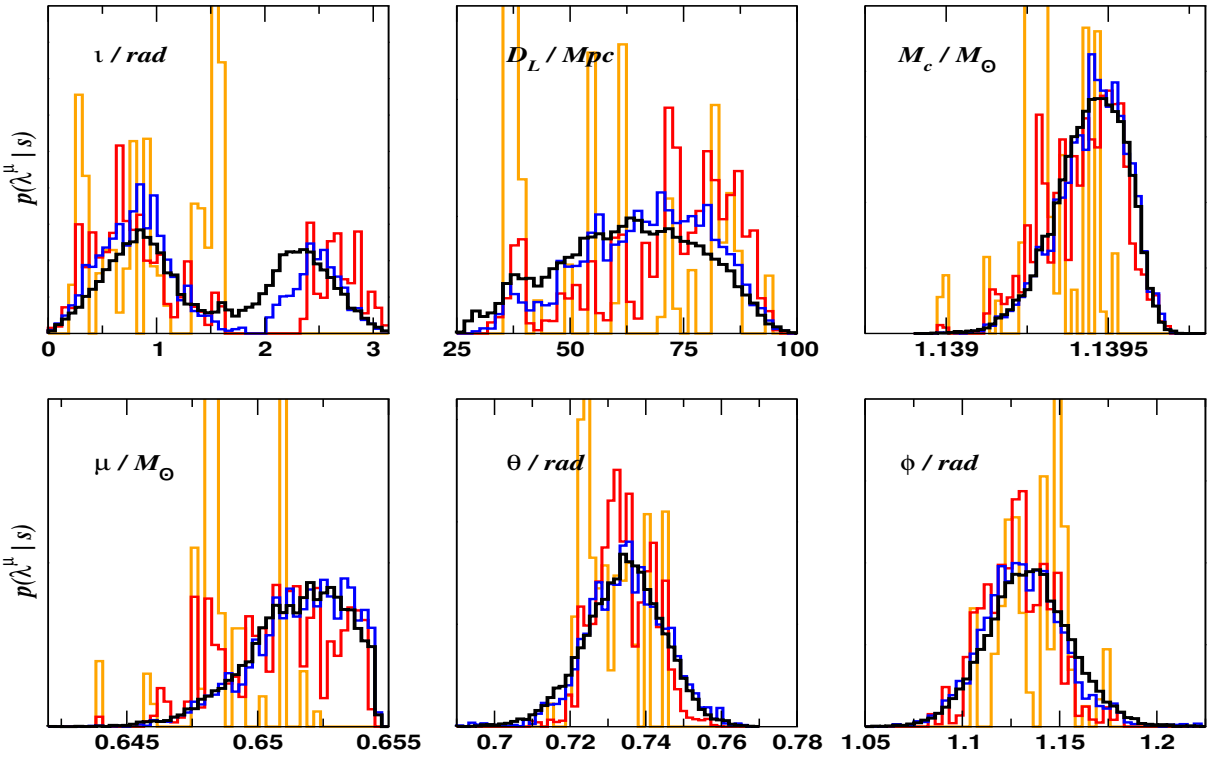

Figure 6.5: Evolution of the marginalised posterior distribution for BNS5, using a DEMC chain for 10^3 (orange), 10^4 (red), 10^5 (blue) and 10^6 (black) iterations, for the parameters $\{\iota, D_L, M_c, \mu, \theta, \phi\}$.

distribution even after 10^6 iterations. This conclusion is further supported by looking at $\{\mathcal{M}_c, \mu\}$, but is most obvious looking at $\{\iota, D_L\}$. This is quite interesting, because if had only looked at Figure 6.3, we would have assumed that as we seemed to be exploring the parameter space quite well, that we would be converging to the target distribution faster in this case.

6.3.1.3 Convergence of the instantaneous median

A powerful way of examining the convergence of a Markov chain is to investigate the instantaneous median of the chain. While we wait for the chain to become statistically independent, we expect large oscillations in the value of the median in the early stages of the chain. As we converge towards the target density, we expect these large deviations to die off, and for the mean to tightly oscillate around the true value.

If we look at Figure 6.6, we can see that it is difficult to say that the DEMC chains have converged for BNS1 after 10^6 iterations. It is clear that looking at ι, D_L, ψ, the chains are still converging as we still have very large oscillations in the instantaneous medians. For the other parameters, they seem to have converged to a particular value at around $\sim 10^5$ iterations, but it is not clear by 10^6 iterations if they have finished moving around as there are still visible oscillations in the curves, or some of the curves have non-zero slopes.

For BNS5 (Figure 6.7), we see a similar pattern. While $\{\iota, D_L\}$ clearly have not converged, again, all other parameters are only showing signs of convergence at a few times 10^5 iterations. And once again, it is not clear if the chains have converged by the end of the run.

We should point out that we have opted not to plot the true values in the plots as we are more interested in the magnitude of the oscillations in convergence. While, as we stated before, this is not a definitive measure of convergence, it does show that 10^6 iterations is not enough to guarantee the convergence of the DEMC chains.

6.3.1.4 Autocorrelation and Integrated Autocorrelation TIme

A final method that we will use as a convergence diagnostic is to use the autocorrelation and integrated autocorrelation time, defined in Chapter 4 by

$$\rho(\tau) = \frac{\sum_{i=1}^{N-\tau} \left(X_i - \overline{X}\right)\left(X_{i+\tau} - \overline{X}\right)}{\sum_{i=1}^{N} \left(X_i - \overline{X}\right)^2}, \tag{6.12}$$

where N are the total number of samples, $\overline{X}$ is the sample mean, and τ is the lag. The integrated autocorrelation time L (ACT) is given by,

$$L = 1 + 2\sum_{\tau=1}^{\tau_{max}} \rho(\tau), \tag{6.13}$$

where τ_{max} is the maximum lag that we choose to use. If a chain is mixing well, we expect the autocorrelation to drop to zero quickly as a function of the lag between samples. If we define τ_{zac} as the zero-autocorrelation lag, a small τ_{zac} infers a good chain mixing and quick convergence to the target density. By contrast, a large τ_{zac} infers inefficient mixing and slow convergence. In theory, we expect the autocorrelation to fall off exponentially as a function of lag. In practice, we do indeed see this type of fall-off, but it is accompanied by a lot of numerical noise in the autocorrelation after the zero crossing lag. This means that the calculation of the ACT is numerically unstable at high lags. In order to ensure that we have numerically stable values of the ACTs, we take τ_{max} in the ACT to be 6×10^5. We then visually inspect the ACT curves as a function of lag to ensure convergence. We further define the number of statistically independent samples (SIS) as

$$SIS = \frac{N}{L}. \tag{6.14}$$

In Figures 6.8 and 6.9, we plot the autocorrelations for all nine parameters, for both BNS1 and BNS5. For BNS1, we can see that the autocorrelations fall off relatively quickly (i.e. on the order of 10^3 lags) for all parameters, except for $\{\iota, \psi, \ln \mathcal{L}\}$. For these three parameters, the mixing of the chain is slow, and

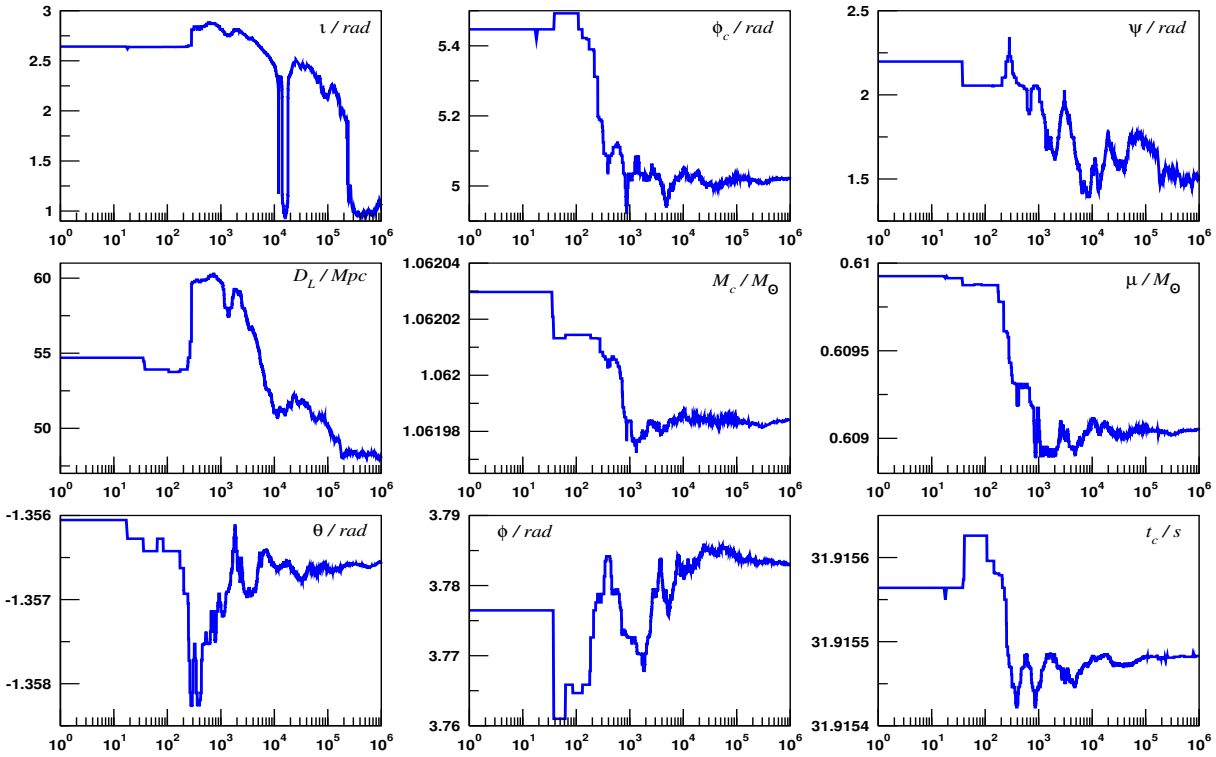

Figure 6.6: A plot of the instantaneous medians for the 10^6 DEMC chain for BNS1. The true values are not plotted as we wish to focus on the magnitude of the oscillations in the convergence.

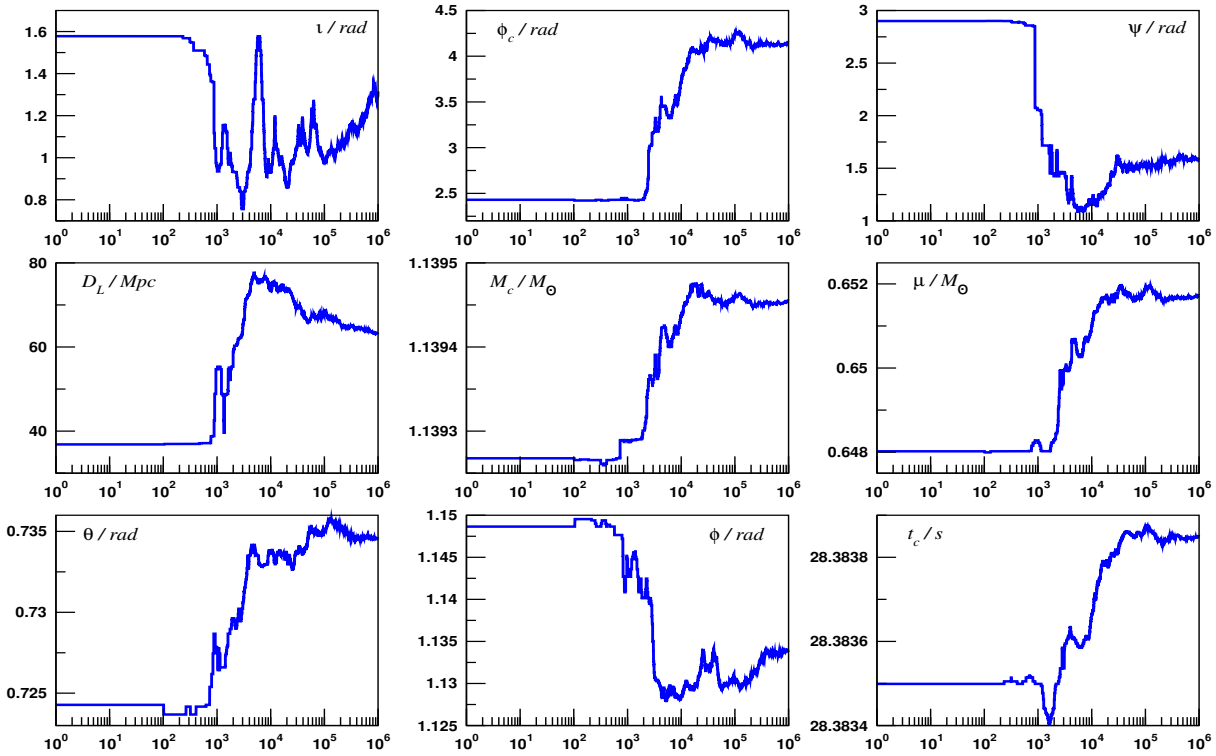

Figure 6.7: A plot of the instantaneous medians for the 10^6 DEMC chain for BNS5. The true values are not plotted as we wish to focus on the magnitude of the oscillations in the convergence.

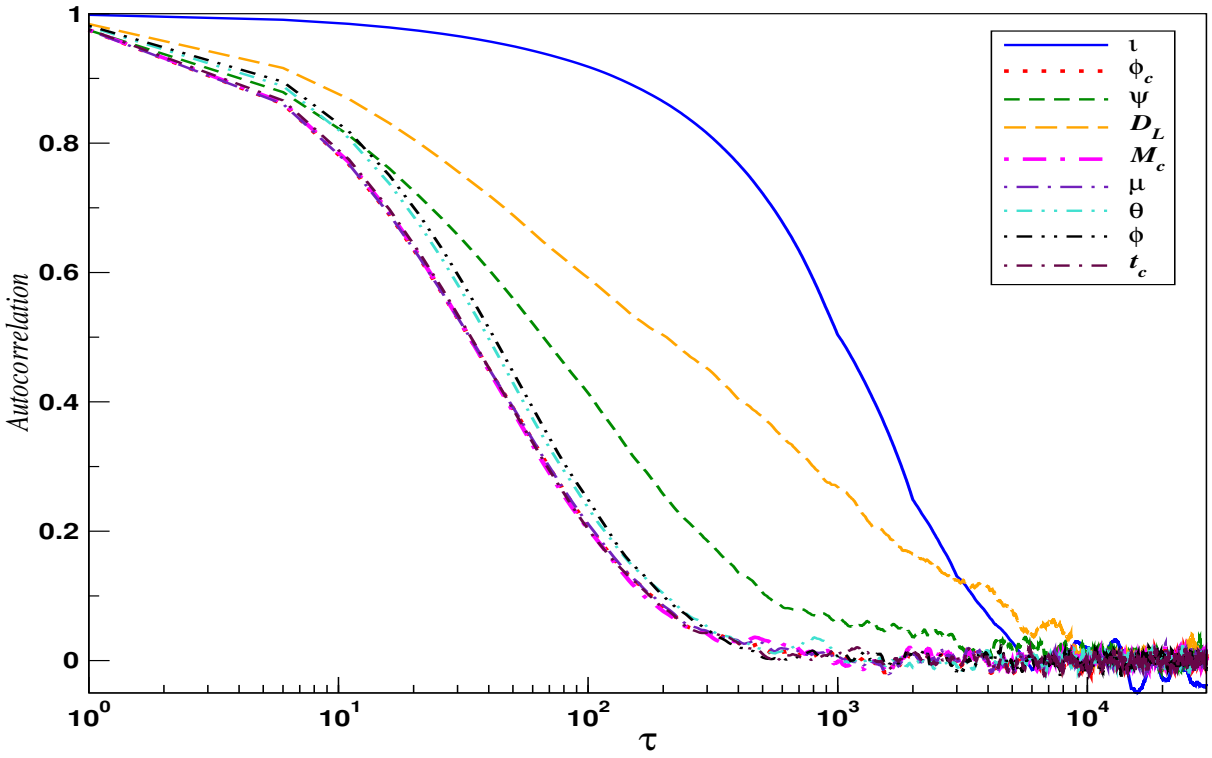

Figure 6.8: Autocorrelation as a function of lag τ for BNS1 using a 10^6 iteration DEMC. The slowest mixing chain in this case is D_L, which has zero autocorrelation at $\tau = 10450$.

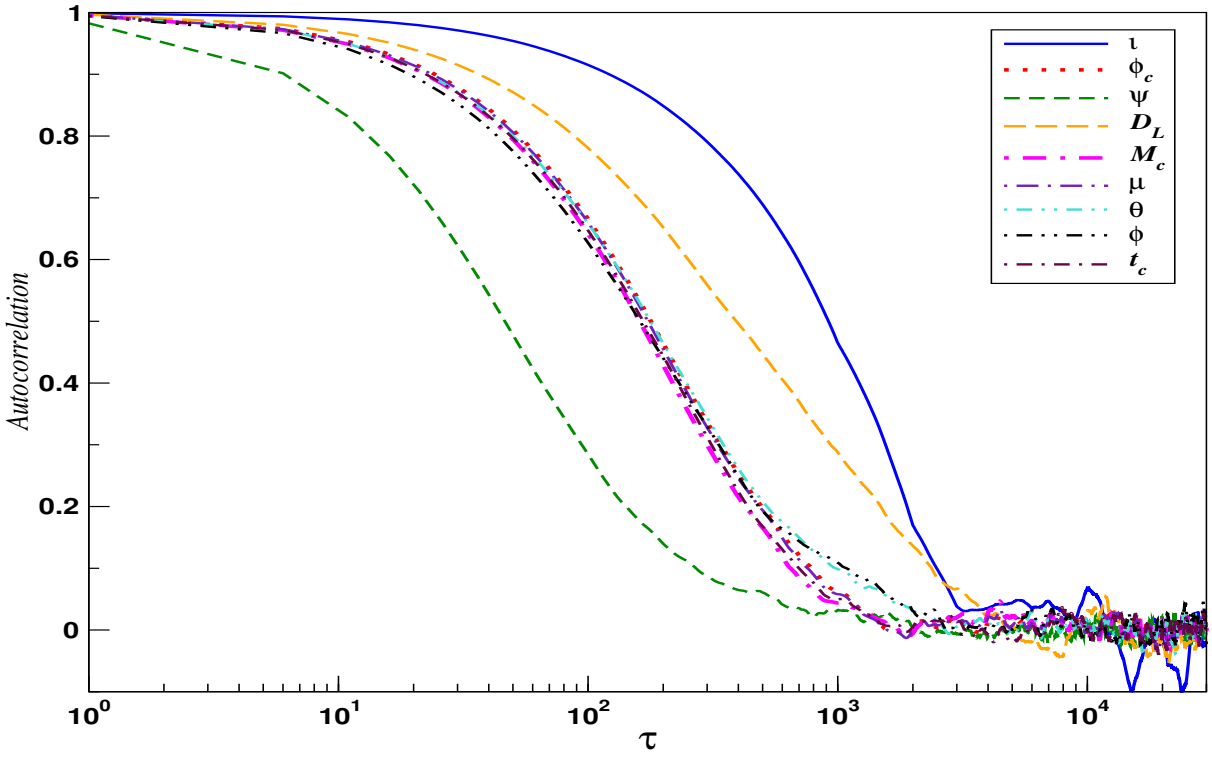

Figure 6.9: Autocorrelation as a function of lag τ for BNS5 using a 10^6 iteration DEMC. The slowest mixing chain in this case is ι which has zero autocorrelation at $\tau = 11902$

the autocorrelations eventually fall to zero at lags of $\tau_{zac} = 5703, 8939$ and 10450 respectively (see Table A.1 for full details). As we are using a univariate ACT, the effective number of SISs from the chains are based on the ACT of the slowest mixing chain. In this case, this is the chain for D_L. Therefore, using this chain, the ACT is $L = 2522$. For a chain of 10^6 iterations, this means that we have only 396 SISs coming from this chain. On a 2.9 GHz Intel i5 processor, with 16 GB of memory, the runtime for BNS1 was 4.67 hours. If we work on the premise that we require a minimum of 5000 SISs, and as the runtime scales linearly with iteration number, we would require a runtime of 59 hours to obtain the 5000 samples (see Table A.1 of Appendix A for details on all ten sources).

For BNS5, we see a slightly different behaviour. In this case, the autocorrelation for ψ falls off fastest, with most of the other parameter autocorrelations falling to zero by a lag of $\tau_{zac} \sim 2000$. Once again, both $\{\iota, D_L\}$ have autocorrelations that decline slowly, inferring once more that the chains mix slowly in these two directions. For this source, the autocorrelations go to zero at $\tau_{zac} = 11902$ and 5168 respectively. As the parameter ι has the slowest mixing chain, we find the at ACT is $L = 2586$, giving us 387 SISs. Again, the runtime for the 10^6 iteration chain is 3.49 hours, meaning that it would effectively take ~ 45 hours to produce 5000 SISs. We should finally mention that while it is usually either ι or D_L that has the slowest mixing chain, we did observe that for BNS8, the two slowest mixing chains were actually the sky position parameters $\{\theta, \phi\}$.

6.3.1.5 Parameter Estimation

In Figures 6.10 and 6.11, and in Table 6.2, we present the final parameter estimation results for BNS1 and BNS5 (final results for the other sources can be found in Appendix C).

The figures display the posterior distributions for all nine parameters, in both cases. We represent the true values with the vertical dashed line. The acceptance rates for both DEMC runs were 17.44% and 7.28% for BNS1 and BNS5 respectively. The first thing that is immediately obvious in the figures is the multi-modality of the solutions in $\{\iota, \psi\}$. While for both binaries, one of the modes for inclination includes the true value, we can see that this is not the case for polarisation angle, where the true value for BNS5 is not captured by one of the modes of the solution. However, as the value of ψ does not have a large effect on the value of the log-likelihood, it does not effect the posterior distributions for the other parameters.

We should draw attention to the fact that because we are using a three-detector network, even though our chains have clearly not converged, and even though we have low acceptance rates, the posterior distributions for $\{\theta, \phi, t_c\}$ are quite symmetric, even if they are not completely smooth. This is due to the fact that with three detectors, and with accurate measurement of the time delays between the detectors, we can localise the source in the sky very well.

For the remaining parameters, we can see that their posterior distributions are not very smooth, with many small peaks. This is to be expected given the results from the instantaneous median plots, where especially for luminosity distance, the chains are a long way from having converged. The final observation is the behaviour of the posterior distribution for the reduced mass μ for BNS1. As this is an almost equal mass source, we can see that the true value is almost exactly at a value of $\mu = 0.61$. However, the DEMC chain has trouble dealing with this somewhat artificial boundary, and we see that the majority of the distribution is shifted away from the true value. If we investigate the chains in $\{m_1, m_2\}$ space, we find that the true values of the binary are not within the 99% CI. Thus, for this source, using a DEMC chain, we would get the individual masses of the system slightly wrong.

In Table 6.2, we quote the median values for the parameter subset $\{D_L, \mathcal{M}_c, \mu, \theta, \phi, t_c\}$. While the value of the inclination is also of astrophysical interest, we omit it from the table due to its bi-modality. For each median, we also quote the 99% confidence interval (CI). In all cases, the true parameter values are contained within the CIs. We will later use these values for comparison with the HMC results. We should point out from the table, as we suspected from the plots on the posterior distributions, all of the true parameter values are contained within the 99% CIs, except for the value of reduced mass for BNS1.

Finally, using the chains for $\{\theta, \phi\}$, we can calculate an error box for the sky position according to [108]

$$\Delta\Omega = 2\pi\sqrt{\Sigma^{\theta\theta}\Sigma^{\phi\phi} - \left(\Sigma^{\theta\phi}\right)^2}, \tag{6.15}$$

where

$$\Sigma^{\theta\theta} = \langle \Delta\cos\theta \Delta\cos\theta \rangle , \tag{6.16}$$

$$\Sigma^{\phi\phi} = \langle \Delta\phi \Delta\phi \rangle , \tag{6.17}$$

$$\Sigma^{\theta\phi} = \langle \Delta\cos\theta \Delta\phi \rangle , \tag{6.18}$$

and $\Sigma^{k\nu} = \left\langle \Delta\lambda^k \Delta\lambda^\nu \right\rangle$ are elements of the variance-covariance matrix, calculated directly from the chains. We find that the sky errors for BNS1 and BNS5 are 0.026 square degrees and 0.27 square degrees respectively.

BNS	1	5
D_L/Mpc	43 $48.084^{+16.135}_{-26.202}$	72 $63.420^{+39.880}_{-39.880}$
$\mathcal{M}_c/M_\odot$	1.06203 $1.06198^{+0.00009}_{-0.00017}$	1.13951 $1.13946^{+0.00024}_{-0.00041}$
$\mu/M_\odot$	0.60996 $0.60905^{+0.00093}_{-0.00342}$	0.65247 $0.65173^{+0.00280}_{-0.00788}$
θ / rad	-1.35612 $-1.35655^{+0.00455}_{-0.00455}$	0.73653 $0.73455^{+0.02981}_{-0.04546}$
ϕ / deg	3.77689 $3.78309^{+0.06658}_{-0.06658}$	1.13272 $1.13395^{+0.05517}_{-0.05517}$
t_c / secs	31.91560 $31.91548^{+0.00027}_{-0.00045}$	28.38391 $28.38385^{+0.00068}_{-0.00102}$
$\Delta\Omega/sq.deg.$	0.025093	0.268290

Table 6.2: True and median chain values for a subset of parameters for BNS1 and BNS5 using a 10^6 iteration DEMC chain. The error estimates on the median values are the 99% credible intervals. We omit values of the inclination ι as the posterior distributions are bi-modal.

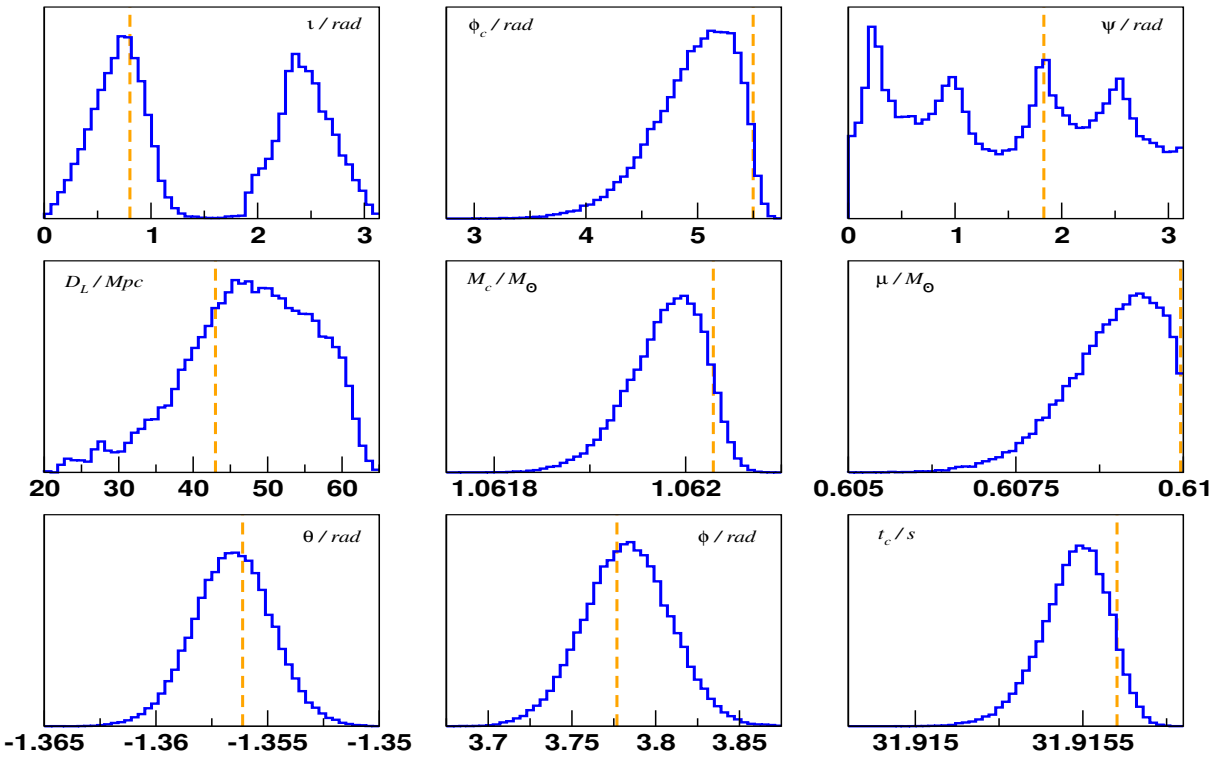

Figure 6.10: Posterior distributions for BNS1. The true values are represented by the orange dashed lines.

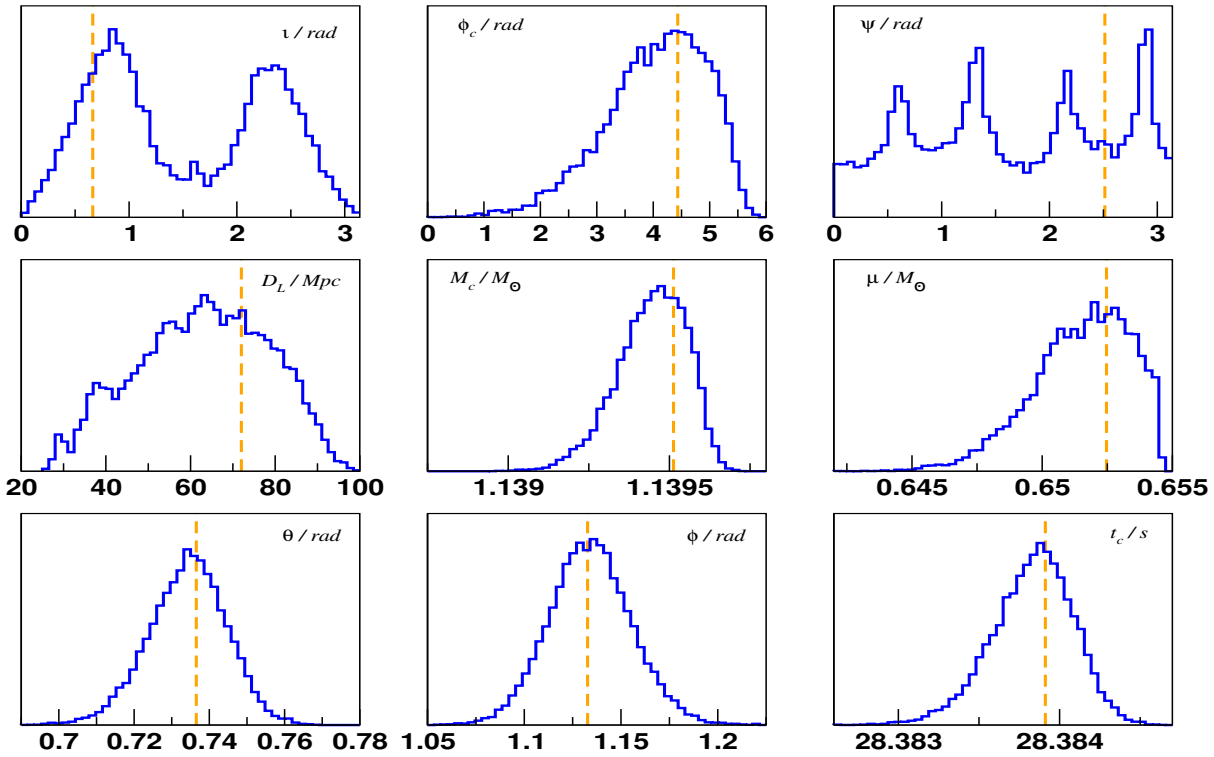

Figure 6.11: Posterior distributions for BNS5. The true values are represented by the orange dashed lines.

Chapter 7

An introduction to Hamiltonian Monte Carlo

7.1 Introduction

The Hamiltonian Monte Carlo (HMC) is a Monte Carlo method that was first implemented by Duane et al in 1987 in the application of lattice field simulations [109]. In this work, they proposed a process that combines stochastic Monte Carlo with deterministic molecular dynamics approaches. In fact, their algorithm was a combination of a Metropolis-Hastings algorithm where Hamilton's equations were used for the proposal with a Gibbs sampling for the momenta, and this was the reason why they decided to call this algorithm Hybrid Monte Carlo. Not long after this landmark paper, another application of the HMC was then realised for Bayesian analysis of neural networks [110] and the HMC started to be recognised in the field of statistics [111]. But it was only after the comprehensive and influential review of Neal in 2012 [112], that the popularity of HMC increased substantially and the method was applied to various scientific fields including applied statistics [113], cosmology [79] and gravitational waves [114].

The reason why the HMC is considered to be a superior alternative to MCMC methods like the Metropolis-Hastings algorithm, is because the HMC avoids the random walk behavior found in most MCMC samplers. As we have seen in Section 4.5.0.1, a Metropolis-Hastings Markov Chain walks randomly in the parameter space with jumps induced by a proposal distribution $q(. \mid .)$. If $q(. \mid .)$ does not reflect the geometry of the posterior distribution, the performances of the algorithm suffers greatly and the chain can either be stuck in parts of the parameter space, or, suffer small exploration where adjacent samples are highly correlated.

To tackle this problem, one solution would be to proposed moves for the chain based on the local geometry of the posterior distribution at the point we currently are. One idea would be to use the gradient of the posterior distribution as a guideline for the jump proposals. However, we do not want to be in a case where the gradient pushes the chain points towards the nearest maximum of the posterior distribution without any opportunity to get away from it. If we make a mechanical analogy, we are in the same situation where a particle is trapped inside a gravitational potential well. To get out of the potential well, the particle needs to have a momentum that is high enough to climb up the potential. The HMC algorithm exploits this analogy by introducing a set of canonical momenta associated to the position parameters on the posterior distribution in order to counteract the effect of the gradient of the posterior distribution that pulls the chain towards the mode.

The Hamiltonian Monte Carlo algorithm then takes advantage of the geometry of the posterior distribution by using a proposal distribution based on Hamilton's equations. By comparing the Hamiltonian at the end points of the trajectories with a Metropolis-Hastings ratio, the algorithm defines a Markov Chain that converges to the posterior distribution in a very efficient manner. In fact, if the algorithm is well tuned, the acceptance rate of the HMC algorithm is often very high and the autocorrelation between two adjacent points of the chain is low. Empirically, studies have shown that the HMC is capable of being $\mathcal{D}$ times more efficient than samplers like Metropolis-Hastings, where $\mathcal{D}$ is the dimension of the problem at hand [79, 114].

There are though a couple of caveats that has prevented this algorithm from being used as a primary sampler choice. The main problem is the computational cost associated with the gradient of the target density that needs to be computed many times for a single trajectory. Another issue is that the algorithm has a number of free parameters that need to be fine tuned in order for the algorithm to be efficient.

Some methods tried to solve this problem by using different criteria to arrive at the optimal choice for these parameters [115]. However, given the complexity of the posterior distribution for most data analysis problems, these methods can be inapplicable in practice. This is the reason why an initial benchmarking study is most of the time needed in order to get the most efficient algorithm. This problem is however a generic issue in MCMC methods that is not specific to HMC.

In this section, we will first describe the general formulation of the algorithm and review aspects of Hamilton's mechanics. We will then explain how this algorithm produces a Markov Chain whose invariant distribution is the posterior distribution.

7.2 Framework of the algorithm

The foundation of HMC is to consider that the inverted logarithm of the posterior distribution or target density $p(q^\mu)$ can be seen as a gravitational potential energy, $\mathcal{U}(q^\mu)$, that depends on the coordinate positions q^μ [109, 112],

$$\mathcal{U}(q^\mu) = -\ln\left[\mathcal{L}(q^\mu)\pi(q^\mu)\right], \tag{7.1}$$

where $\mathcal{L}(q^\mu)$ is the log-likelihood and $\pi(q^\mu)$ is the prior distribution. In our case, the coordinate position q^μ represents the parameters of the gravitational waves template λ^μ but we will stick with the notation q^μ as it is the common notation used in the literature. For every coordinate position q^μ, we associate a set of canonical momenta p^μ, where (q^μ, p^μ) define a phase space coordinate. The kinetic energy of the system is given by the expression,

$$\mathcal{K}(p^\mu) = \frac{1}{2} M^{-1}_{\mu\nu} p^\mu p^\nu, \tag{7.2}$$

where $M_{\mu\nu}$ is a positive definite mass matrix. The mass matrix is the first free parameter of the algorithm that needs to be specified. As we will see, the mass matrix is essential for the dynamics of the solution and should reflect the dynamical ranges of the parameters. A usual choice for the mass matrix is to consider a diagonal mass matrix where its matrix elements are described as,

$$M_{\mu\nu} = \begin{cases} m_\mu & \text{if } \mu = \nu \\ \\ 0 & \text{if } \mu \neq \nu \end{cases} \tag{7.3}$$

where $m^\mu > 0$ in order to satisfy the positiveness condition of the matrix. Now that we have defined the potential and kinetic energy, we can put everything together to construct the Hamiltonian of the system as,

$$\begin{aligned} \mathcal{H}(q^\mu, p^\mu) &= \mathcal{U}(q^\mu) + \mathcal{K}(p^\mu), & (7.4)\\ &= -\ln\left[\mathcal{L}(q^\mu)\pi(q^\mu)\right] + \frac{1}{2} m_\mu^{-1} \left(p^\mu\right)^2. & (7.5) \end{aligned}$$

The Hamiltonian $\mathcal{H}(q^\mu, p^\mu)$ defines the total energy of the system. The dynamical evolution of the system in fictitious time t can then be inferred from Hamilton's equations as,

$$\begin{aligned} \frac{\mathrm{d}q^\mu}{\mathrm{d}t} &= \frac{\partial \mathcal{H}}{\partial p^\mu} = \frac{\partial \mathcal{K}}{\partial p^\mu}, & (7.6)\\ \frac{\mathrm{d}p^\mu}{\mathrm{d}t} &= -\frac{\partial \mathcal{H}}{\partial q^\mu} = -\frac{\partial \mathcal{U}}{\partial p^\mu}. & (7.7) \end{aligned}$$

The two previous differential equations completely determine the time evolution of the set (q^μ, p^μ).

7.3 Hamiltonian dynamics

We define the mapping T_τ to be the linear map between an initial phase space state $(q^\mu(0), p^\mu(0))$ to the phase space state $(q^\mu(\tau), p^\mu(\tau))$ as dictated by Hamilton's equations where τ is a given fixed time. First of all, Hamilton's equations are time-reversible, meaning that a mapping T_τ is one-to-one and has a unique inverse $T_{-\tau}$. In our configuration, the inverse mapping is obtained by negating the sign of the

momentum,

$$\begin{aligned} T_\tau &\rightarrow T_{-\tau}, && (7.8)\\ t &\rightarrow -t, && (7.9)\\ q^\mu &\rightarrow q^\mu, && (7.10)\\ p^\mu = m^\mu \frac{\mathrm{d}q^\mu}{\mathrm{d}t} &\rightarrow m^\mu \frac{\mathrm{d}q^\mu}{-\mathrm{d}t} = -p^\mu. && (7.11) \end{aligned}$$

Another important property is the conservation of the Hamiltonian for any mapping T_τ. Once again, this can be easily proved using Eq. (7.6) and Eq. (7.7) ,

$$\frac{\mathrm{d}\mathcal{H}}{\mathrm{d}t} = \frac{\partial \mathcal{H}}{\partial q^\mu}\frac{\mathrm{d}q^\mu}{\mathrm{d}t} + \frac{\partial \mathcal{H}}{\partial p^\mu}\frac{\mathrm{d}p^\mu}{\mathrm{d}t} = \frac{\partial \mathcal{H}}{\partial q^\mu}\frac{\partial \mathcal{H}}{\partial p^\mu} - \frac{\partial \mathcal{H}}{\partial p^\mu}\frac{\partial \mathcal{H}}{\partial q^\mu} = 0. \tag{7.12}$$

Finally, Hamiltonian dynamics preserves the phase space volume. This result is also known as Liouville's theorem. To prove this assertion, one can compute the divergence induced by a mapping T_τ as,

$$\nabla_{T_\tau} = \frac{\partial}{\partial q^\mu}\frac{\mathrm{d}q^\mu}{\mathrm{d}t} + \frac{\partial}{\partial p^\mu}\frac{\mathrm{d}p^\mu}{\mathrm{d}t} = \frac{\partial}{\partial q^\mu}\frac{\partial \mathcal{H}}{\partial p^\mu} - \frac{\partial}{\partial p^\mu}\frac{\partial \mathcal{H}}{\partial q^\mu} = \frac{\partial^2 \mathcal{H}}{\partial q^\mu \partial p^\mu} - \frac{\partial^2 \mathcal{H}}{\partial p^\mu \partial q^\mu} = 0. \tag{7.13}$$

Since the divergence is equal to 0, the divergence theorem states that the phase space volume is conserved for any T_τ.

Most of the time we can not solve Hamilton's equations analytically and we need to solve them iteratively using integrators. The positions and momenta are then evaluated discretely along a trajectory with l steps separated by step size ϵ. Since we use numerical approximations, some of the previous properties defined before might not be verified. We say that the integrator is time-reversible if its numerical integration scheme is time-reversible, and we say that it is symplectic if the integrator preserves phase space volume as define in Eq. (7.13). For all integrators, each step evaluation of q^μ and p^μ introduces a local error on the conservation of the Hamiltonian that depends on the value of the step size ϵ. This results in a global error on the trajectory that depends on the number of steps l and ϵ.

Following the example in [112], we will present three different integrators applied for a simple example in order to understand the caveats related to solving numerically Hamilton's equations. Let us consider a simple one-dimensional example where the Hamiltonian is given by

$$\mathcal{H}(q,p) = \frac{1}{2}q^2 + \frac{1}{2}p^2, \tag{7.14}$$

and Hamilton's equations take the simple form

$$\begin{aligned} \frac{\mathrm{d}q}{\mathrm{d}t} &= p, && (7.15)\\ \frac{\mathrm{d}p}{\mathrm{d}t} &= -q. && (7.16) \end{aligned}$$

For each of the integrator, we compute one numerical trajectory with $l = 20$ steps, step size $\epsilon = 0.3$ and an initial phase space point $\left(q^0 = 0, p^0 = 1\right)$. In Figure 7.1, we plot the resulting numerical trajectory (solid line) along with the analytical solution (dashed line).

The first example of an integrator is Euler's method,

$$\begin{aligned} p(t+\epsilon) &= p(t) + \epsilon\frac{\mathrm{d}p}{\mathrm{d}t} = p(t) - \epsilon q(t), && (7.17)\\ q(t+\epsilon) &= q(t) + \epsilon\frac{\mathrm{d}q}{\mathrm{d}t} = q(t) + \epsilon p(t). && (7.18) \end{aligned}$$

In the left hand cell of Figure 7.1, we observe that the numerical trajectory using Euler's method quickly diverges from the true trajectory. If we reduce the value of the step size ϵ, the errors on the numerical trajectories will be smaller because the local error in the conservation of the Hamiltonian is of order $\mathcal{O}(\epsilon^2)$ for a single step and $\mathcal{O}(\epsilon)$ for the full trajectory, but the divergence to infinity is still present. The reason for this divergence comes from the fact that Euler's method is not a sympletic integrator and is as a consequence not adapted to solve Hamilton's equations.

The second integrator is a modification of Euler's method where we use the updated value of the momentum in Eq. (7.18),

$$\begin{aligned} p(t+\epsilon) &= p(t) - \epsilon q(t), && (7.19)\\ q(t+\epsilon) &= q(t) + \epsilon p(t+\epsilon). && (7.20) \end{aligned}$$

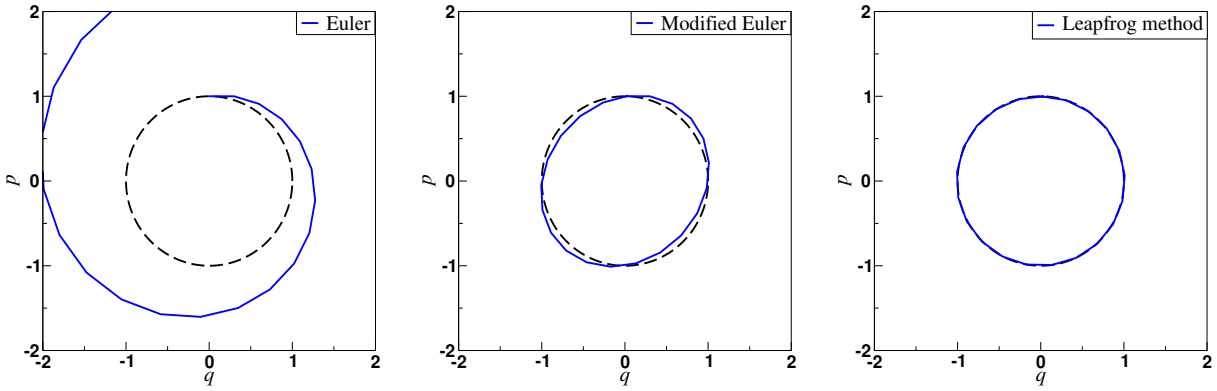

Figure 7.1: Example of different integrators to solve Hamilton's equations: Euler's method (left), modified Euler's method (center) and Leapfrog method (right). The numerical simulations have been generated for $l = 20$ steps with $\epsilon = 0.3$ and are plotted in blue. The analytical solution is given in black dashed line.

While the local and global errors of modified Euler's method are of the same order than for Euler's method, unlike Euler's method this integrator is sympletic. And as a result, we see in the center cell of Figure 7.1 that the resulting trajectory is much better than in the previous case. However, some differences between the analytical and numerical trajectories are still noticeable.

The last integrator is the Leapfrog method,

$$p\left(t+\frac{\epsilon}{2}\right) = p(t) - \frac{\epsilon}{2}q(t), \tag{7.21}$$

$$q\left(t+\epsilon\right) = q(t) + \epsilon p\left(t+\frac{\epsilon}{2}\right), \tag{7.22}$$

$$p(t+\epsilon) = p\left(t+\frac{\epsilon}{2}\right) - \frac{\epsilon}{2}q(t+\epsilon). \tag{7.23}$$

In this case, we observe that the numerical trajectory the right hand cell of Figure 7.1 matches perfectly with the true solution. Like modified Euler's method, the Leapfrog algorithm is sympletic. But the increased accuracy of the Leapfrog integrator compared to modified Euler's method, is explained by the fact that in this case the local error for each step size is of the order $\mathcal{O}(\epsilon^3)$ and of the order $\mathcal{O}(\epsilon^2)$ for the whole trajectory. Finally, since it will be of importance later on, we should highlight that this integrator is time reversible providing we negate the sign of the momentum at the end of the trajectory. .

As we have mentioned before, the main bottleneck of HMC is the computation required for the gradient of the potential energy. All the three previous integrators only required a single evaluation of the gradient every step. It would be possible to use higher order sympletic integrators to have a better precision than the Leapfrog method. However, these methods would require more evaluations of the gradient which would result in a large increase in computation time. For the Hamiltonian defined in Eq. (7.5), the Leapfrog equations method is expressed as,

$$p^\mu(t+\epsilon/2) = p^\mu(t) - \frac{\epsilon}{2}\frac{\partial \mathcal{U}(q^\mu)}{\partial q^\mu}\Big|_{q^\mu(t)}, \tag{7.24}$$

$$q^\mu(t+\epsilon) = q^\mu(t) + \epsilon m_\mu^{-1} p^\mu(t+\epsilon/2), \tag{7.25}$$

$$p^\mu(t+\epsilon) = p^\mu(t+\epsilon/2) - \frac{\epsilon}{2}\frac{\partial \mathcal{U}(q^\mu)}{\partial q^\mu}\Big|_{q^\mu(t+\epsilon)}. \tag{7.26}$$

7.4 How is the HMC used as a MCMC algorithm?

Now that we have defined the main setup of the algorithm, we still need to understand how the algorithm can be used as a MCMC algorithm. If we look at the problem from the point of view of statistical mechanics, the energy of the system defines a canonical distribution or joint probability distribution $\Pi(q^\mu, p^\mu)$ given by,

$$\Pi(q^\mu, p^\mu) = \frac{1}{C_{\mathcal{H}}}\exp(-\frac{\mathcal{H}(q^\mu, p^\mu)}{T}), \tag{7.27}$$

where T is the temperature of the system and C is a normalising constant such that the probability sums to one. If we set the temperature to $T = 1$ and inject the expression from Eq. (7.5) into the previous equation, we find that the canonical distribution is rewritten as,

$$\Pi(q^\mu, p^\mu) = \frac{1}{C_\mathcal{H}} \exp(-\mathcal{U}(q^\mu)) \exp(-\mathcal{K}(p^\mu)), \tag{7.28}$$

$$= \frac{1}{C_\mathcal{H}} p(q^\mu)\Pi(p^\mu), \tag{7.29}$$

where we used the expression of the potential energy from Eq. (7.1) to get Eq. (7.29). Using the expression of the kinetic energy in Eq. (7.2), we find that the probability distribution of the momenta $\Pi(p^\mu)$ is a multivariate Gaussian distribution with mean 0 and variance m^μ,

$$\Pi(p^\mu) = \exp\left[-\frac{(p^\mu)^2}{2m^\mu}\right],$$

$$\rightarrow \quad p^\mu \sim \mathcal{N}(0, m^\mu). \tag{7.30}$$

The equation Eq. (7.29) tells us that the joint probability distribution is separable. This is a very important property because it implies that the posterior distribution $p(q^\mu)$ and the distribution of the momenta $\Pi(p^\mu)$ are independent. This implies that if we generate the momenta directly from their true distribution from Eq. (7.30), and then use a proposal distribution driven by Hamilton's equations to sample from the joint canonical distribution $\Pi(q^\mu, p^\mu)$, we marginalise over the momenta distribution and have directly access to the posterior distribution. The HMC algorithm was designed in such a way to sample from the canonical joint distribution using a Markov Chain in phase space (q_t^μ, p_t^μ). For every realisation t, the HMC algorithm is given by the following process,

1. Draw the momenta from their exact distribution $\mathcal{N}(0, m_\mu)$ independently of positions to define the starting position (q_t^μ, p_t^μ).

2. Use Hamilton's equations as a proposal distribution to propose a new phase space state (q_F^μ, p_F^μ). To do that, we compute the numerical trajectory from (q_t^μ, p_t^μ) using the Leapfrog method for a trajectory of l steps with step size ϵ.

3. At the end of the trajectory, evaluate the Metropolis-Hasting ratio between the initial phase space state and the proposed phase space state with a negation of the momenta as

$$\alpha = \min\Big\{1, \exp\left[\mathcal{H}(q_t^\mu, p_t^\mu) - \mathcal{H}(q_F^\mu, -p_F^\mu)\right]\Big\} \tag{7.31}$$

4. With probability α, set $(q_{t+1}^\mu, p_{t+1}^\mu) = (q_F^\mu, p_F^\mu)$ otherwise set $(q_{t+1}^\mu, p_{t+1}^\mu) = (q_t^\mu, p_t^\mu)$

The Metropolis-Hastings step in the algorithm is necessary to take care of the non conservation of the Hamiltonian along a trajectory numerically generated with the Leapfrog method. The reason why the momenta are negated at the end of the trajectory is to assure time-reversibility hence assuring that the proposal distribution is symmetric, allowing us to write the Metropolis-Hastings ratio in the form given in Eq. (7.31).

To prove that the HMC algorithm indeed samples from the joint canonical distribution $\Pi(q^\mu, p^\mu)$, we need to show that $\Pi(q^\mu, p^\mu)$ is the invariant distribution of the Markov Chain generated by the HMC. Regarding the first step of the algorithm, since we draw the momenta from their true distribution, it is straightforward to see that this step indeed leaves the joint canonical distribution invariant. We are now left to prove that the Metropolis-Hastings step where Hamilton's equations are used as a proposal distribution leaves the joint canonical distribution invariant. To do that, we can show that the associated transition kernel with this Metropolis-Hastings step is in detailed balance with $\Pi(q^\mu, p^\mu)$. We present here a comprehensive proof of detailed balance presented in [112].

Let us partition the phase space state into regions A_k that have the same volume V and denote by B_k the image produced by solving Hamilton's equations with the Leapfrog method for l steps plus the negation of the momentum at the end. Since the Leapfrog method is sympletic, the regions B_k have the same volume V than the regions A_k. If we now write the detailed balance condition from Eq. (4.51) in this case, we have for all i, j,

$$P(A_i)T(B_j|A_i) = P(B_j)T(A_i|B_j), \tag{7.32}$$

where P is the probability under the joint distribution Π and $T(X|Y)$ is the probability under the kernel transition of proposing a move and accept it in region X for a current state in region Y. Since Hamilton's

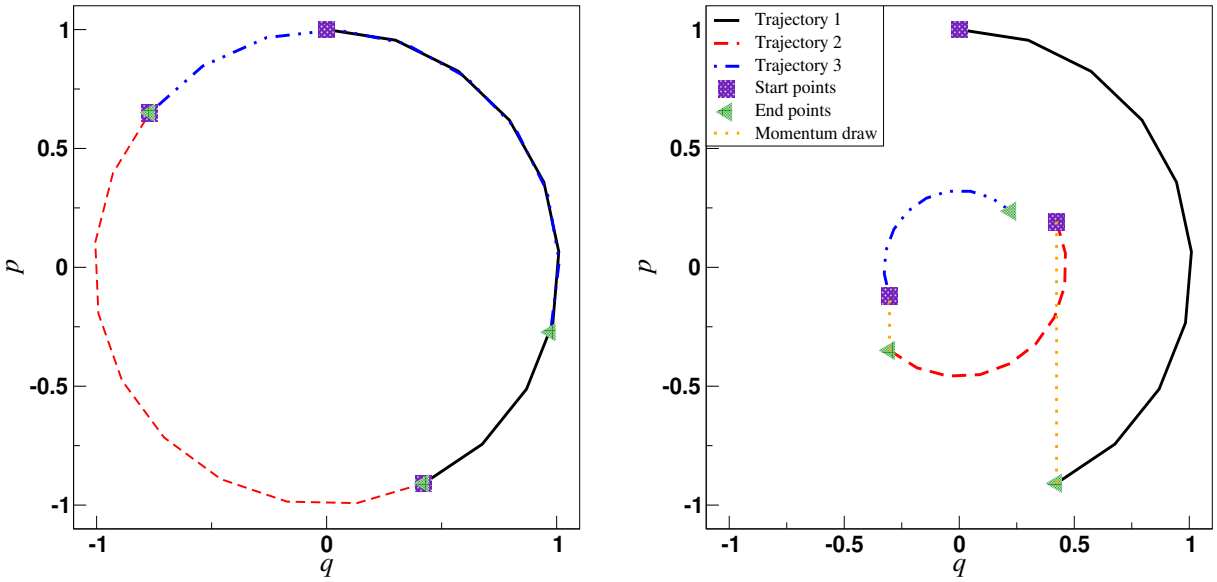

Figure 7.2: Illustration of the effect of momentum draw for a set of three numerical trajectories. The trajectories are computed with the Leapfrog method using $l = 10$ steps and a step size $\epsilon = 0.3$. On the left hand plot we do not redraw the momentum, while on the right hand plot the momentum is redrawn before the beginning of each trajectory.

equations are deterministic, if $i \neq j$ then $T(B_j|A_i) = T(A_i|B_j) = 0$ and the detailed balance condition in Eq. (7.32) is verified. In the case where $i = j = k$, let us consider that we take the limit where V tends to 0 such that the Hamiltonian $\mathcal{H}_{A_k}$ and $\mathcal{H}_{B_k}$ in regions A_k and B_k is constant. Using the expression of the canonical joint distribution from Eq. (7.27) integrated on the volume V, we can rewrite the detailed balance condition from Eq. (7.32) as,

$$\frac{V}{C_{\mathcal{H}}} \exp(-\mathcal{H}_{A_k}) \min\Big\{1, \exp\left[-\mathcal{H}_{B_k} + \mathcal{H}_{A_k}\right]\Big\} = \frac{V}{C_{\mathcal{H}}} \exp(-\mathcal{H}_{B_k}) \min\Big\{1, \exp\left[-\mathcal{H}_{A_k} + \mathcal{H}_{B_k}\right]\Big\} \tag{7.33}$$

It is then straightforward to see that the previous condition is indeed verified also in this case and we then prove that the joint canonical distribution is the invariant distribution of the HMC algorithm. Note that volume preservation is essential to make detailed balance hold. Finally since the momenta are drawn from their true distribution, the HMC algorithm will directly generate samples from the posterior distribution.

In addition to that, the Markov chain generated by the HMC is ergodic and the ergodic theorem then assures us that it generates samples of the posterior distribution. The exploration of the posterior distribution is assured by the momenta draw in the first step of the algorithm. In fact, the change in momenta changes the value of the Hamiltonian hence making the chain explores various parts of the joint canonical distribution and thus posterior distribution. Without this change of momenta, the algorithm would be confined to iso-surfaces of constant Hamiltonian and the algorithm would not be ergodic. As an illustration of the importance of momenta draw, we take the same example presented in the previous section. We use Leapfrog method to generate three trajectories with $l = 10$ and $\epsilon = 0.3$ in two cases, one where we do not redraw the momentum (left hand plot) and one where the momentum are drawn from a Gaussian distribution with mean 0 and sigma 0.5 at the beginning of each trajectory. The numerical trajectories are plotted in Figure 7.2. Without a momentum draw, we observe that the trajectories will always constrain us to move on the same circle without exploring other parts of the parameter space. When we introduce a momenta draw, we then have different values for the Hamiltonian and different exploration of the space parameter. This illustrates the importance of the first momentum draw in order to make the algorithm ergodic.

Chapter 8

Application of HMC to parameter estimation for BNS

In the previous chapter, we have introduced the HMC algorithm and described how it works in a general framework. The goal of this work was to apply the HMC algorithm in the context of parameter estimation for BNS sources. In order to make the HMC algorithm as efficient as possible, we had to go though a phase of algorithm benchmarking where we tested the algorithm on a single BNS source. For the source, we have selected BNS1 from the set of binaries introduced in Chapter 6.

A number of aspects needed to be investigated to develop a HMC algorithm that is as efficient as possible. First of all, we have seen in Chapter 7 that the HMC algorithm has a number of free parameters, namely the matrix $M_{\mu\nu}$, the step size ϵ and the length of the trajectory l. Since there is no generic method to give a values for these parameters, we had to test different options and see how they impact the algorithm both in terms of exploration of the parameter space and acceptance rate. The results of these investigations are presented in the first two sections of the chapter.

Then we had to find a solution for the bottleneck of the algorithm, that is the computation time required for the evaluation of the gradient of the posterior distributions at each step of the trajectory. As a starting point, we tried to apply the method that was already implemented for parameter estimation of supermassive black hole binaries with LISA [114]. However, we did not have any guarantee that this method would work in the context of BNS parameter estimation where we need to deal with highly multi-modal posterior distribution. As a consequence, the main research effort was dedicated to solving this problem. We present in Section 8.3 the main results of this investigation and the solution we found for the gradient bottleneck.

When the development was mature enough, we decided to apply it for a full parameter estimation of BNS1 and compare the results with those obtained using the DEMC chain. In Section 8.6, we compare the results of the HMC and DEMC chain and demonstrate that our HMC algorithm is both able to produce the correct posterior distribution and able to surpass the performances of the DEMC in all aspects.

8.1 Mass matrix

The mass matrix $M_{\mu\nu}$ plays an important role in the dynamics of the system. It is used both to compute Hamilton's equations in the Leapfrog equations and for the initial draw in momenta at the beginning of each trajectory. For the study outlined in Chapter 7, we only considered a diagonal mass matrix. In this section, we will explain how we choose the mass matrix in order to have an efficient algorithm.

The first option that we have considered for the mass matrix is an identity mass matrix given by,

$$M_{\mu\nu} = I_{\mu\nu} \tag{8.1}$$

This option is the easiest choice and the Leapfrog equations equations are then only dependent on the step size ϵ. At this initial stage of the algorithm benchmarking, we were not interested in optimising the value of the step size, and we only wanted to have a value such that the acceptance rate was good enough. In this case, we found that a step size of $\epsilon = 10^{-6}$ gave a good acceptance close to 98%, while increasing to $\epsilon = 5.0 \times 10^{-6}$ decreased the acceptance rate down to 14%. To test the efficiency of this mass matrix option, we have decided to run a simulation for 200 trajectories with trajectory length

$l = 200$. In Figure 8.1, we plot the resulting chain for all nine parameters depending on the number of trajectories. For $\mathcal{M}_c$, μ and t_c, we see that the chain is moving and oscillating but the exploration of the parameter space stays limited. For θ and ϕ, the exploration of the chain is even smaller and we do not see the oscillations anymore. For the rest of the parameters, the movement of the chain appears to be constant. A closer inspection reveals that the chain is moving but the amplitude of the exploration is not visible at the scales used for the plot. This study indicates that the identity mass matrix is a poor choice and results in an extremely limited exploration of the parameter space. This can be understood by the fact that the dynamical ranges of the parameters is extremely different. As an example, we know from the preliminary study with the DEMC chain that the dynamical range for chirp mass is $\Delta\mathcal{M}_c \sim 10^{-5}$ while the dynamical range for inclination is $\Delta\iota \sim 1$. However, an identity mass matrix assumes that the dynamics of the parameters should be the same for all the parameters. This essentially means that the exploration is limited by the parameter with the smallest dynamical range, and will then restrict the exploration of the others.

To solve for this problem, we have decided to test another option for the mass matrix by taking a diagonal mass matrix where the diagonal components m_μ are not equal and reflect the dynamical ranges of the problems. In this case, it is possible to rewrite the Leapfrog equations as [112],

$$\begin{aligned}
\tilde{p}^\mu(\tau + \epsilon^\mu/2) &= \tilde{p}^\mu(\tau) + \frac{\epsilon^\mu}{2}\frac{\partial \ln\left[\mathcal{L}(q^\mu)\right]}{\partial q^\mu}\Big|_{q^\mu(\tau)}, \\
q^\mu(\tau + \epsilon^\mu) &= q^\mu(\tau) + \epsilon^\mu \tilde{p}^\mu(\tau + \epsilon^\mu/2), \\
\tilde{p}^\mu(\tau + \epsilon^\mu) &= \tilde{p}^\mu(\tau + \epsilon^\mu/2) + \frac{\partial \ln\left[\mathcal{L}(q^\mu)\right]}{\partial q^\mu}\Big|_{q^\mu(\tau+\epsilon^\mu)},
\end{aligned} \tag{8.2}$$

where we define the scaled momenta $\tilde{p}^\mu = s_\mu p^\mu$, the scaled step sizes $\epsilon^\mu = s_\mu \epsilon$ and $s_\mu = \sqrt{m_\mu^{-1}}$. In this form, the phase space variables $(q^\mu, \tilde{p}^\mu)$ no longer follow Hamiltonian trajectories at constant times. However, the joint distribution remains unchanged due to the Metropolis-Hastings step at the end of the trajectory. Furthermore, we now draw all momenta from a Gaussian distribution with mean 0 and variance 1, i.e. $p^\mu \sim \mathcal{N}(0, 1)$. We also highlight here that since we chose uniform prior distribution for the parameters (see Chapter 6), the gradients of the target distribution in the previous Leapfrog equations are replaced by the gradients of the log-likelihood.

To set the values of the diagonal components of the mass matrix, we have decided to use the variance predicted by the Fisher information matrix (FIM) $C_{\mu\nu} = \Gamma^{-1}_{\mu\nu}$,

$$m_\mu^{-1} = C_{\mu\mu} = \left(\sigma_\mu^{FIM}\right)^2, \tag{8.3}$$

which yields the values for s_μ,

$$s_\mu = \sigma_\mu^{FIM}. \tag{8.4}$$

The approximation given by the Fisher information matrix is then sufficient to set an order of magnitude for the dynamical ranges of the parameters. For the source BNS1, we did not have a problem with the condition number of the Fisher matrix and its inverse was numerically stable.. However, we will see in the next chapter that we need to add extra conditions on the mass matrix when the inverse of the FIM is numerically unstable.

As for the identity mass matrix case, we have ran a simulation with 200 trajectories and $l = 200$. For the step size, we observed that the value $\epsilon = 10^{-3}$ gave a similar acceptance than the one we had when using the identity mass matrix. In Figure 8.1, we plot the chains obtained with this simulation. We clearly see here that the algorithm is now capable of wider exploration in the parameter space for all the parameters. We highlight here that the exploration is for the moment not optimised because we still need to find the best values for the step size ϵ.

8.2 Step size of the Leapfrog integrator

Now that we have fixed the mass matrix $M_{\mu\nu}$, we are left with choosing the step size of the algorithm ϵ. To do that we must consider two criteria: the acceptance rate and exploration of the algorithm. We will consider these two aspects in these section and derive the optimal value we found for ϵ

First of all, we investigated the impact of ϵ on the acceptance rate. We have seen that the error introduced in the Hamiltonian by solving Hamilton's equations iteratively with the leapfrog equations

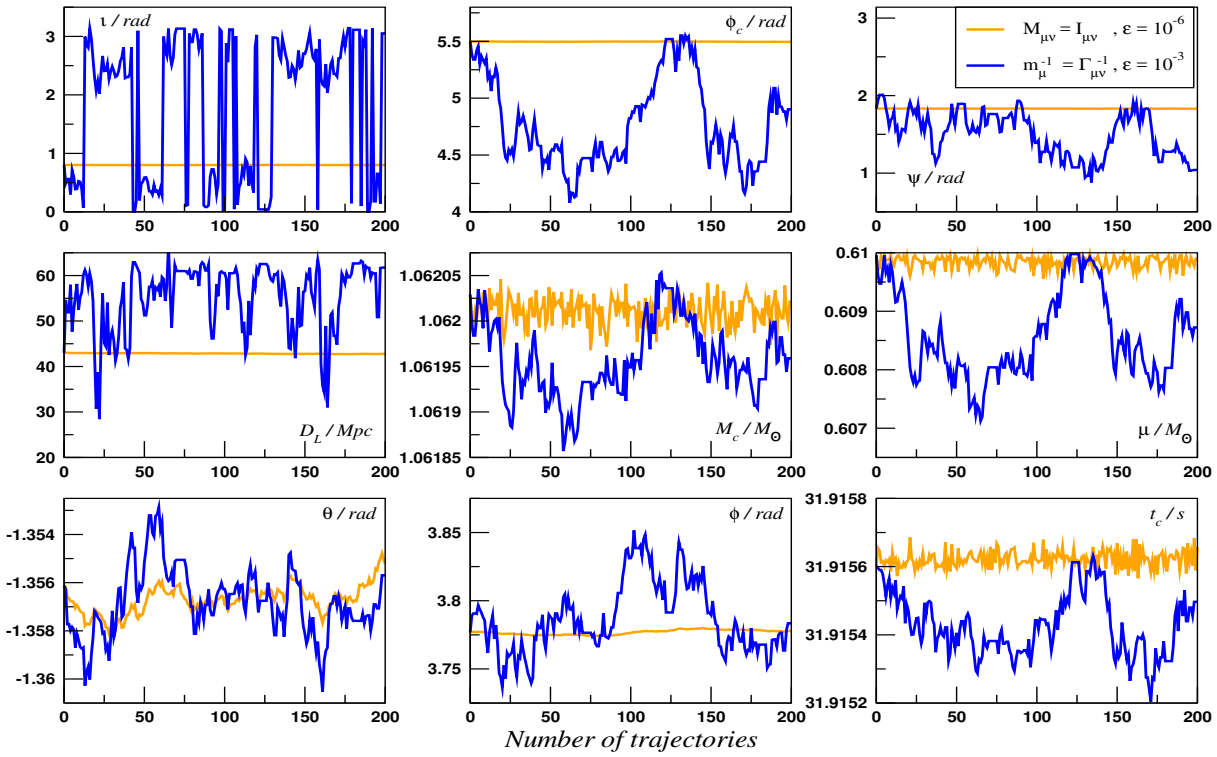

Figure 8.1: A 200 trajectory HMC chain with $l = 200$ where the mass matrix was set to the identity matrix (orange) and scaled with the FIM (blue). For the identity mass matrix the step size was set to $\epsilon = 10^{-6}$ while for the mass matrix scaled with the FIM, the step size was $\epsilon = 10^{-3}$.

for a single trajectory is of the order $\mathcal{O}(\epsilon^2)$. As a consequence, the acceptance rate will decrease if we increase the value of ϵ. To assess this effect, we ran a set of simulations for 500 trajectories with $l = 200$ and different values for the step size ϵ. In Figure 8.2, we plot the resulting acceptance rate as a function of the number of trajectories for each of the simulations. We observe that for $\epsilon < 2.5 \times 10^{-3}$, the acceptance rate is always higher than 95%. For larger values of ϵ, the acceptance rate drops down to 90% and 72% for $\epsilon = 5.0 \times 10^{-3}$ and $\epsilon = 1.0 \times 10^{-2}$ respectively. These results are thus in agreement with what we expected initially.

We then investigated the effect of the value of ϵ in terms of exploration. In Figure 8.3, we plot the end points of the trajectories for the set of simulations in the two dimensional surface $\{\iota, D_L\}$. If we look at the first case with $\epsilon = 1.0 \times 10^{-4}$, we see that the points of the chain are clustered in a small area of the parameter space. If we increase the step size up to $\epsilon = 5.0 \times 10^{-4}$, the algorithm manages to widely explore one mode of the posterior distribution for values of inclination $\iota < \pi/2$. However, not a single trajectory manages to reach the other mode of the posterior distribution. To explore the second mode, we need to increase the step size up to $\epsilon = 1.0 \times 10^{-3}$, but even in this case the exploration is extremely limited. If we now increase the step size to $\epsilon = 2.5 \times 10^{-3}$ we see that the exploration of the algorithm is good and both modes are explored during the first 500 trajectories of the HMC. Finally for higher values of ϵ, we do not see much improvement in terms of exploration.

As a conclusion, we found that the value $\epsilon = 2.5 \times 10^{-3}$ was the optimal value in terms of acceptance and exploration of the algorithm.

8.3 Calculating the gradients of the posterior distribution

As we described in Chapter 7, the main bottleneck of the HMC algorithm is the computational cost required for the evaluation of the gradients of the posterior distribution at each step of the Leapfrog method. Even though the efficiency of the algorithm is higher than most MCMC methods, the computation time related to calculating the gradient of the posterior distribution can be prohibitive and make the HMC algorithm not usable in practice. A key aspect of this work has been to design a method to reduced the computation time while keeping the acceptance and efficiency of the algorithm high. In this section,

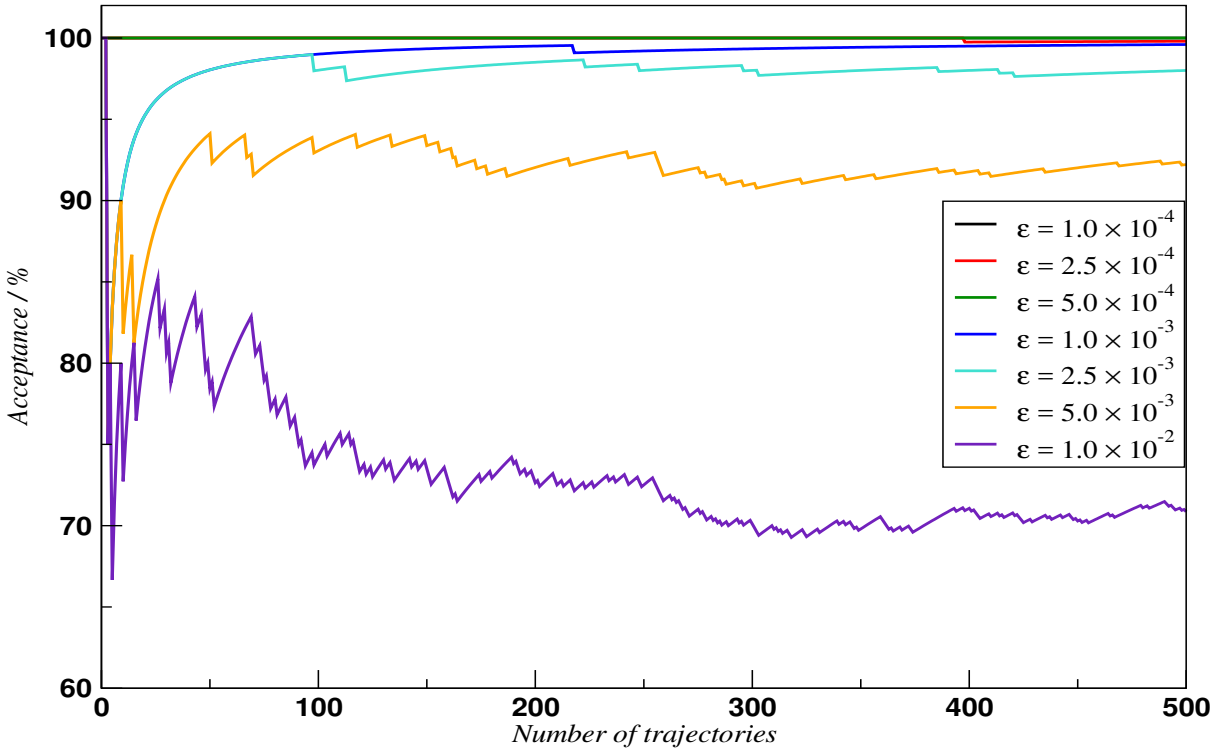

Figure 8.2: Acceptance rate as a function of trajectory number for a set of simulations with different values for the step size ϵ. The simulations were run for 500 trajectories with trajectory length $l = 200$.

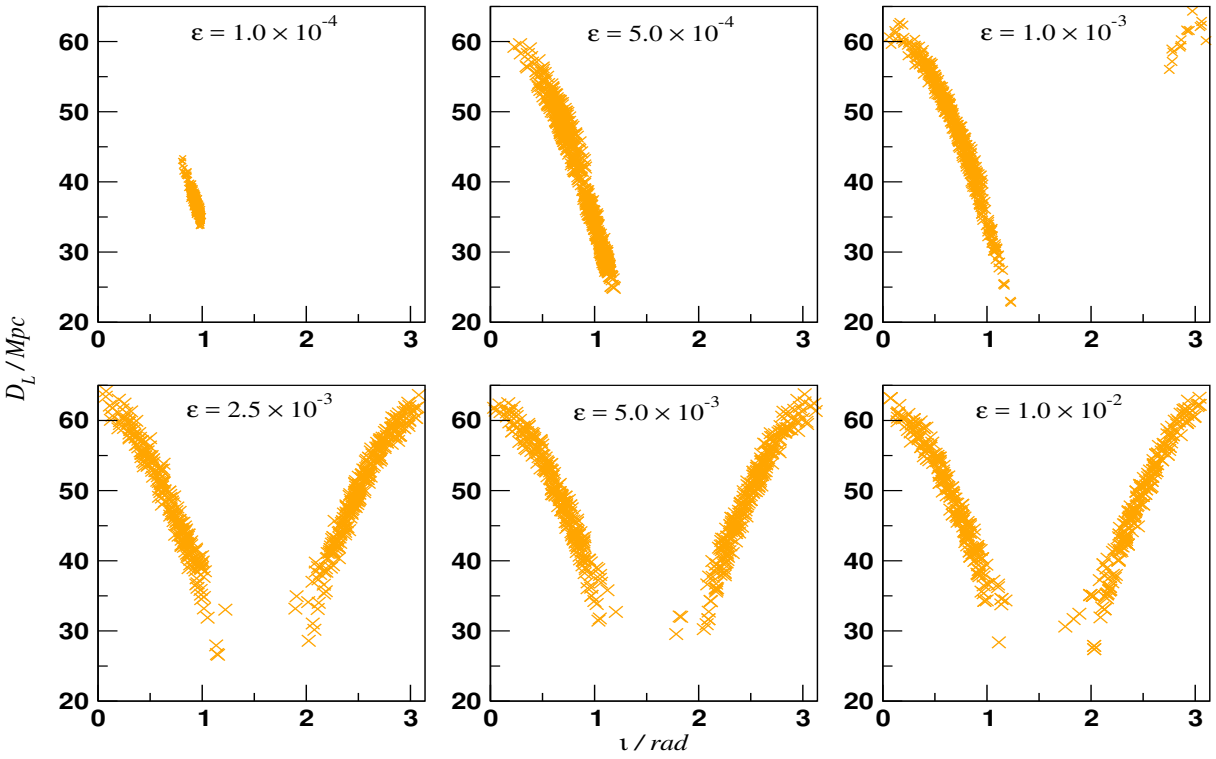

Figure 8.3: End points of the trajectory in the $\{\iota, D_L\}$ two dimensional surface for a set of simulations with different values for the step size ϵ. The simulations were ran for 500 trajectories with trajectory length $l = 200$.

we will present the main results and implementations that were performed to reduce the computation time to an acceptable level.

8.3.1 Calculating the gradient of the log-likelihood

As we have mentioned in Chapter 6, we use uninformative prior distribution for all the parameters. Since these prior distributions are uniform, the derivatives of posterior distribution become dominated by the gradients of the log-likelihood. As our algorithm uses the reduced log-likelihood from Eq. (4.35), the gradients of the log-likelihood can be written as

$$\frac{\partial \ln \mathcal{L}}{\partial q^\mu} = \Big\langle s \Big| \frac{\partial h}{\partial q^\mu} \Big\rangle - \Big\langle h \Big| \frac{\partial h}{\partial q^\mu} \Big\rangle. \tag{8.5}$$

To calculate the gradients of the log-likelihood, we then need to compute the derivatives of the waveform with respect to the parameters. We could use numerical differencing to evaluate the derivatives of the waveform, but this would require us to compute 18 waveforms to evaluate the gradients at a single step of the trajectory. As an example, for a single trajectory of the algorithm with $l = 200$ steps, this would translate in 3600 waveforms evaluations. In order to speed up the process, we have derived analytical expressions for the derivatives of the waveform. For $\{\psi, \sin\theta, \phi\}$, due to the complexity of the analytic derivatives of the beam pattern functions, we decided to calculate the derivatives numerically. In this case, we found it was sufficient to use a forward-differencing scheme,

$$\frac{\partial F^{+,\times}}{\partial q^\mu} \approx \frac{F^{+,\times}(q^\mu + \delta q^\mu) - F^{+,\times}(q^\mu)}{\delta q^\mu} \tag{8.6}$$

where we found that an offset of $\delta q^\mu = 10^{-6}$ gave the best value. As a result, the number of waveform generations is reduced to nine for a single step and 1800 for a full trajectory.

8.3.2 Analytic approximation of the gradients

A solution to speed up the computation time for the gradients has been proposed in [114]. In this study, they proposed to approximate the gradients of the log-likelihood with an analytical cubic fit derived from a set of points, coming from accepted trajectories where the gradients have been computed numerically. After the fit, the trajectories in phase space are then driven by the approximated gradient instead of the numerical gradient which greatly reduces the computation time. From an algorithmic point of view, the movement of a particle induced by the approximated gradients can be seen as evolving in a "shadow potential ̈that approximates the proper potential $\mathcal{U}(q^\mu)$. As we have seen in Chapter 7, the Hamiltonian trajectories can be thought of as a proposal distribution. This means that, as long as we move back to the proper potential $\mathcal{U}(q^\mu)$ at the end of the trajectory to evaluate the Hamiltonian, we can use these approximated trajectories as proposal distribution to reduce the computation time of the algorithm.

In terms of the algorithm, this means that we now have three distinct phases. During Phase I, the code is ran for a number of N initial trajectories using numerical gradients of the log-likelihood. For each accepted trajectory, both the values of the parameters and the associated values of the gradients of the log-likelihood are stored numerically for the fit. We highlight here that only the points from the accepted trajectories are contributing to the fit. If the trajectory is rejected, it means that the trajectories has visited a part of the parameter space where the values for the gradients of the log-likelihood was numerically unstable leading to a non-conservation of the Hamiltonian. At the end of Phase I, we are left with a set of m points, where for each point i we represent the parameters values by q_i^μ and the gradients of the log-likelihood by y_i^μ. During the Phase II of the algorithm, the HMC code is stopped to perform the fit using the M points. Finally in Phase III, we use the analytical fit derived in Phase II to compute trajectories with the approximated values of the gradients.

To do the fit in Phase II, we use a cubic fit with the m points, where each gradient of the log-likelihood is approximated by an analytical function $f(q^\mu)$ that is written as,

$$f(q^\mu) = \sum_{i=1}^{D} a_i q^i + \sum_{j=1}^{D}\sum_{k=j}^{D} a_{jk} q^j q^k + \sum_{l=1}^{D}\sum_{v=l}^{D}\sum_{w=v}^{D} a_{lvw} q^l q^v q^w, \tag{8.7}$$

where D is the dimension of the problem and a_i, a_{jk} and a_{lvw} represent the coefficients of the fit. We highlight that a_{jk} represent the independent coefficients of a $D \times D$ symmetric matrix and a_{lvw} the coefficients of a $D \times D \times D$ symmetric tensor. In our case, the dimension of the parameter space is $D = 9$

which yields a total number of 220 independent coefficients for a single gradient of the log-likelihood approximation. To solve for the fit coefficients for a single gradient of the log-likelihood with respect to the parameter q^μ, we introduce a least square function defined by,

$$S(a) = \sum_{i=1}^{M} \left(\sum_{j=1}^{220} a_j \phi_j(q_i^\mu) - y_i^\mu \right)^2 , \tag{8.8}$$

where we define,

$$\phi_j(q^\mu) = \frac{\partial f(q^\mu)}{\partial a_j}. \tag{8.9}$$

If we rewrite the least square function in matrix form, we have the following expression

$$S(a) = (\mathbf{Ja} - \mathbf{y})^{\mathrm{T}} (\mathbf{Ja} - \mathbf{y}) , \tag{8.10}$$

where $\mathbf{y}$ is the vector with components y_i^ν, $\mathbf{a}$ is the vector containing the coefficients for the fit and $\mathbf{J}$ is the Jacobian matrix expressed as,

$$\mathbf{J} = \begin{pmatrix} \phi_1(q_1^\mu) & \cdots & \phi_m(q_1^\mu) \\ \vdots & \ddots & \vdots \\ \phi_1(q_M^\mu) & \cdots & \phi_m(q_M^\mu) \end{pmatrix}. \tag{8.11}$$

To solve for the coefficients, we need to find the minimum of the least square function $\mathbf{a}_{fit}$ as

$$\frac{\partial S(\mathbf{a}_{fit})}{\partial a_k} = 0. \tag{8.12}$$

If we substitute the expression from Eq. (8.10) in the previous expression, we find that

$$2\mathbf{J}^T\mathbf{J}\mathbf{a}_{fit} - 2\mathbf{J}^T y = 0. \tag{8.13}$$

The expression of the coefficients from the fit is then given by,

$$\mathbf{a}_{fit} = \left(\mathbf{J}^T\mathbf{J}\right)^{-1} \mathbf{J}^T y. \tag{8.14}$$

Practically, we have used the method of singular value decomposition to obtain the solution for the least square function $\mathbf{a}_{fit}$. The computation time to solve the fit for a single gradient fit with this method is of the order of $\mathcal{O}(M)$. We highlight here that in the general case where we have a number of fit coefficients $k < M$, the computation time to solve the system with singular value decomposition is of the order of $\mathcal{O}\left(Mk^2\right)$

8.3.3 Application of the cubic fit approximation

The quality of the cubic fit defined previously is highly dependent on Phase I of the algorithm where we run N numerical trajectories. First of all, the more trajectories we run, and assuming a constant acceptance rate, the more points we have for the fit. In addition to that we also need to ensure that the exploration in this first Phase is sufficient to have a good coverage of the parameter space. As we were in the testing phase of the algorithm, we decided to run the algorithm on the APC cluster, whose processors are 2.66 GHz E5640 Intel(R) Xeon(R) processors [116]. The average computation time for a single trajectory during Phase I on the APC cluster was around 30 s, but the computation time reduces down to about 6 s on an Intel i7 processor.

To test the performances of the cubic fit, we carried out a first series of tests where we used different values for the number of initial trajectories with, $N = \{100, 250, 750, 1000, 1500, 2000, 3000\}$. We have used the configuration that was described before with $\epsilon = 2.5 \times 10^{-3}$ and $l = 200$. In Table 8.1, we summarise for each of these cases the acceptance rates, the associated number of points for the fit, the total computation time and the averaged computation time for a single trajectory during the initial phase where gradients are computed numerically. In terms of acceptance rate, we see that it is almost constant for all the simulations with a value close to 96%. Now, regarding the computation time, we observe that as expected it is linearly dependent on the number of initial trajectories. The average computation time for a single trajectory over all the simulations is 28.78 s. In terms of sampling, this means that the code takes around 30 s to produce a sample using HMC with numerical gradients. This is an illustration of

N	$\text{AR}_{PI}/\%$	M	t_{PI}/hrs	t_{PI}^{traj}/s
100	97.030	19400	0.77	27.72
250	96.813	48400	1.93	27.79
500	97.804	97800	4.31	31.03
750	97.603	146400	6.17	29.60
1000	97.203	194400	8.07	29.05
1500	97.135	291400	11.87	28.49
2000	97.050	388200	15.67	28.21
3000	96.934	581600	23.60	28.32

Table 8.1: Table presenting the value of the acceptance rate at the end of Phase I AR_{PI}, the number of points generated for the fit M, the total computation time for Phase I t_{PI} along with the average computation time for a single trajectory during Phase I t_{PI}^{traj}. These values are given for the different scenarios where we use N initial trajectories for the fit

the gradient bottleneck problem we discussed before. This number will be taken as a reference to see what is the improvement when using fitted gradients instead.

During Phase II of the algorithm, we then have performed the cubic fit described in the previous section using a singular value decomposition. In Table 8.2, we give the computation time required to fit the gradient coefficients for each scenario. We observe that the duration of Phase II linearly depends on the number N of fit points as expected from a singular value decomposition method. In all the cases, we observe that the time to do the fit is around an order of magnitude lower than the time to produce the numerical initial trajectories.

Finally, to test the goodness of the fit, we ran the code for 10^5 trajectories using the same values of ϵ and l used during the initial phase. In Figure 8.4, we plot the acceptance rate as a function of the number of trajectories N. In addition, we give in Table 8.2 the final acceptance rate, the total computation time and the computation time for a single trajectory when using the fitted gradient.

First of all, let us investigate the dependence of the final acceptance rate on the number of points used for the fit. For $N = 100$, the acceptance rate is around 8% which can be understood given the small number of points used for the fit, $M = 19400$. However, as we increase the number of initial numerical trajectories, we see that the final acceptance rate increases up to a point where it becomes equal to 43% for $N = 1500$ and a corresponding number of fit points, $M = 291400$. If we further increase N to 3000, we do not see any improvement on the final acceptance rate that stays around 43%. In any case, this is still a significant drop in final acceptance rate compared to Phase I of the algorithm where the final acceptance was around 96% with the same values of ϵ and l. In addition, we observe that sometimes the acceptance rate decreases monotonically for a number of trajectories. For instance for $N = 1000$, the acceptance drops from 37 to 34% between the trajectories 28000 and 30000. This indicates that the HMC was stuck in a part of the parameter space without being able to to move for over 2000 trajectories. Since we do not observe this behavior in the numerical phase, it shows that the fit seems to be even worse in some parts of the parameter space, which prevents the chain for moving.

Overall, these results indicate that something is wrong with our cubic fit approximation and that the gradient is not numerically stable. However if we look at the computation time, as expected, we see a huge improvement when using the approximate gradient with a total time around 1.2 hours for 10^5 trajectories. In terms of computation time per trajectory, we find that we have a factor of 700 improvement compared to the case with the numerical gradients, with an averaged computation time of 42.19 ms.

To understand why the fit was not working, we first looked at the distribution of the gradient fit points generated during Phase I for the different values of N. In Figure 8.5 we plot the distribution of the fitting points in the two-dimensional surface for $\{\iota, D_L\}$ as a function of the number of initial numerical trajectories. We see that even for $N = 100$, the two modes are clearly visible in the distribution of the fitting points. However, the middle branch in inclination, for ι between 1 and 2, is still unexplored. If we increase the number of initial trajectories to $N = 500$, the two modes start to be more populated and are connected in the middle branch. However, there is still a lack of sufficient coverage in the middle branch. At $N = 750$, the fitting points start to have a relatively good coverage of all the relevant parts of the parameter space for the posterior distribution, with more points in the middle branch. If we increase the number of initial number of numerical trajectories from $N = 750$ to $N = 3000$, we see that the density of points increases but there is no additional part of the parameter space that is explored.

N	t_{PII}/hrs	AR_{PIII}/%	t_{PIII}/hrs	t_{PIII}^{traj}/ms
100	0.03	7.93	0.91	32.68
250	0.08	22.48	1.18	42.38
500	0.32	29.31	1.20	43.27
750	0.76	34.11	1.21	43.72
1000	1.13	37.06	1.21	43.45
1500	1.58	42.91	1.21	43.64
2000	2.35	44.46	1.22	44.10
3000	3.39	42.58	1.28	46.20

Table 8.2: Table presenting the values for the total computation time for the fit of the gradient coefficients during Phase II t_{PII}, the acceptance rate at the end of Phase III AR_{PIII}, the total computation time for Phase III t_{PIII} along with the average computation time for a single trajectory t_{PIII}^{traj}. These values are given for the different scenarios where we use N initial trajectories for the cubic fit approximation.

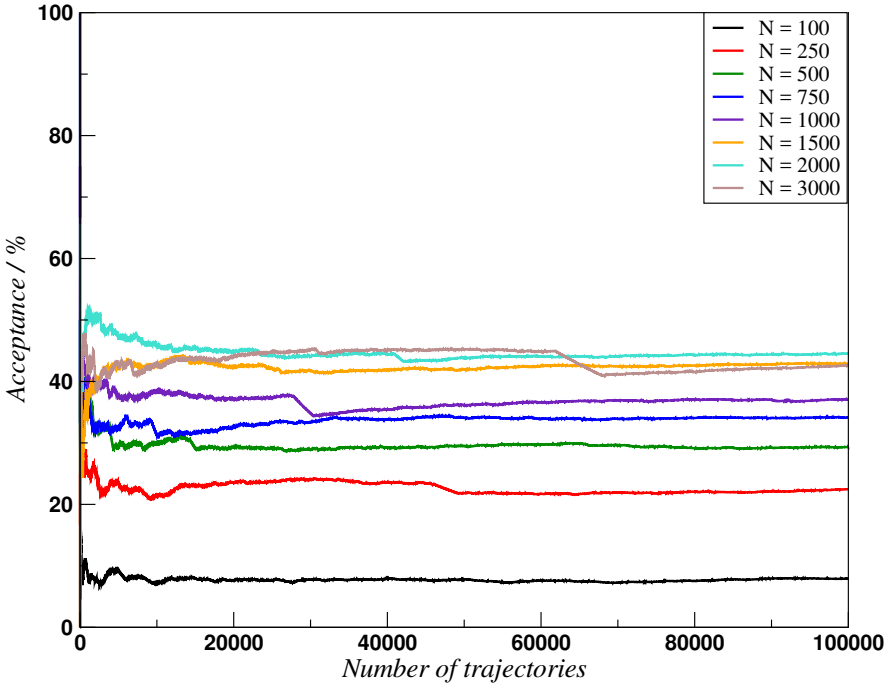

Figure 8.4: Acceptance rate as a function of the number of trajectories when using a cubic fit approximation for the gradients of the log-likelihood. These results are presented for different number of initial trajectories N.

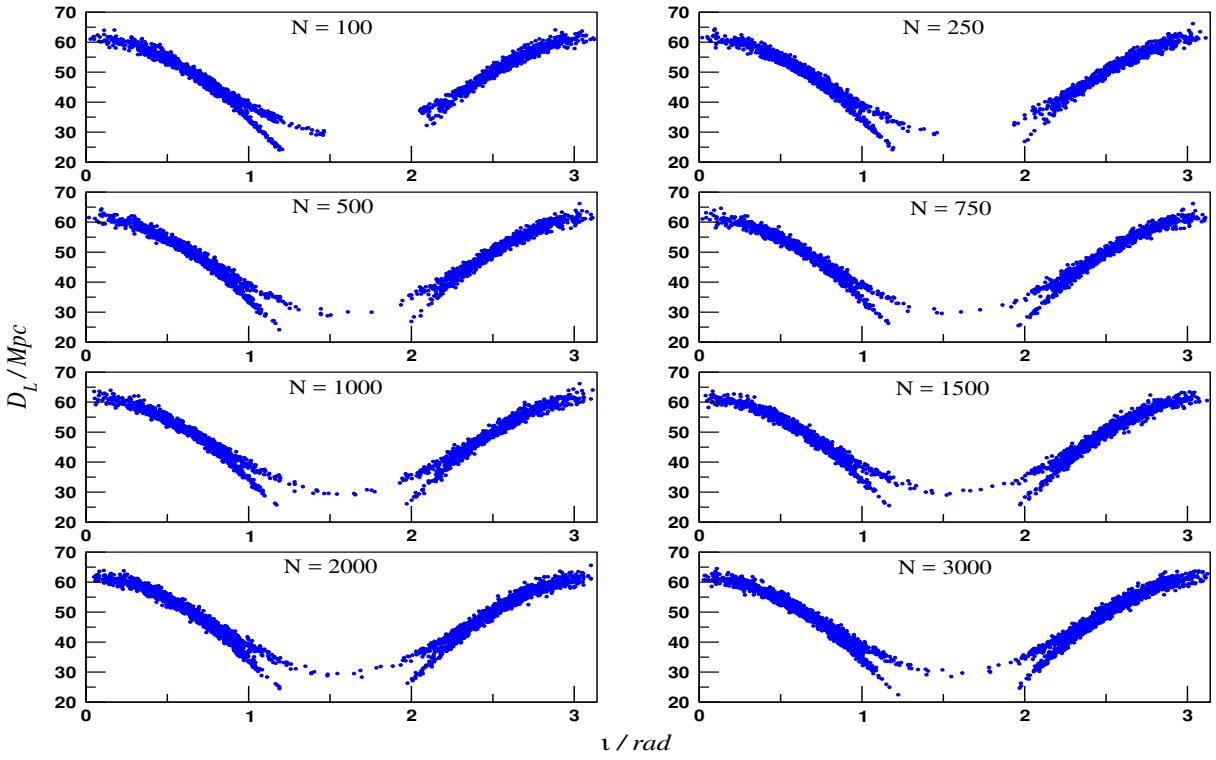

Figure 8.5: Two-dimensional surface plot in $\{\iota, D_L\}$ showing the distribution of the M points for the fitting depending on the number N of initial numerical trajectories. The inclination ι is expressed in radian and the luminosity distance D_L is expressed in Mpc.

In Figure 8.6, we plot the same distributions of fitting points but this time in the two dimensional surface defined by the masses parameters $\{\mathcal{M}_c, \mu\}$ in units of solar masses. Once again, we observe that at $N = 750$, most of the parts of the parameter space relevant for the posterior distribution have been covered during the initial numerical trajectories. Increasing the number of initial numerical trajectories from $N = 750$ to $N = 3000$ then increases the density of the fitting points coverage. However, we note that some parts of the parameter space for $\mathcal{M}_c \in [1.0618, 1.06185]\, M_\odot$ and $\mu \in [0.6055, 0.6065]\, M_\odot$ are covered only when $N = 2000$ or greater. In terms of fitting, this means that some part of the parameter space might not be properly represented during the fit in Phase II which then impacts the quality of the cubic fit approximation in Phase III.

What we observe from this investigation is that after $N = 750$, the set of points generated by the numerical trajectories for the fit has a good coverage of the parameter space and should be able to produce good fits for the gradients of the log-likelihood. However, we have seen that in this case the final acceptance rate at the end of Phase III was only 34%. Then to understand why the algorithm still was not performing as expected, we decided to run a test where instead of using approximate gradients for all nine parameters, we used the approximate gradient for one parameter only, while using simultaneously numerical gradients for all other parameters. By doing so, we can assess the goodness of the fit for each of the gradient and discriminate what are the parameters that are causing the decrease in acceptance rate. For this test, we used $N = 750$ initial numerical trajectories and then run the code for another 2250 trajectories making a total of 3000 trajectories. In Figure 8.7, we plot the values of the acceptance rates for each of the simulation where we specify in the legend the parameter for which the gradient was computed using the cubic fit. From this study, we see two different cases depending on the parameters. For the set of parameters $\{\ln \mathcal{M}_c, \ln \mu, \phi_c, \ln t_c, \sin(\theta), \phi\}$, we see that the acceptance rate stays almost constant after $N = 750$, with a value close to 96%. What this tells us is that the quality of the approximation for the gradient is good and ensures that the conservation of the Hamiltonian on the trajectory is almost as good as when using numerical gradients.

For the three other parameter, namely $\{\cos\iota, \psi, \ln D_L\}$, we see that the acceptance rate decreases after 750 trajectories from 96 % to 80% for ψ, and to 25% for $\cos\iota$ and $\ln D_L$. For the latter parameters, we can not differentiate the two curves in the plot as none of the trajectories were accepted when using

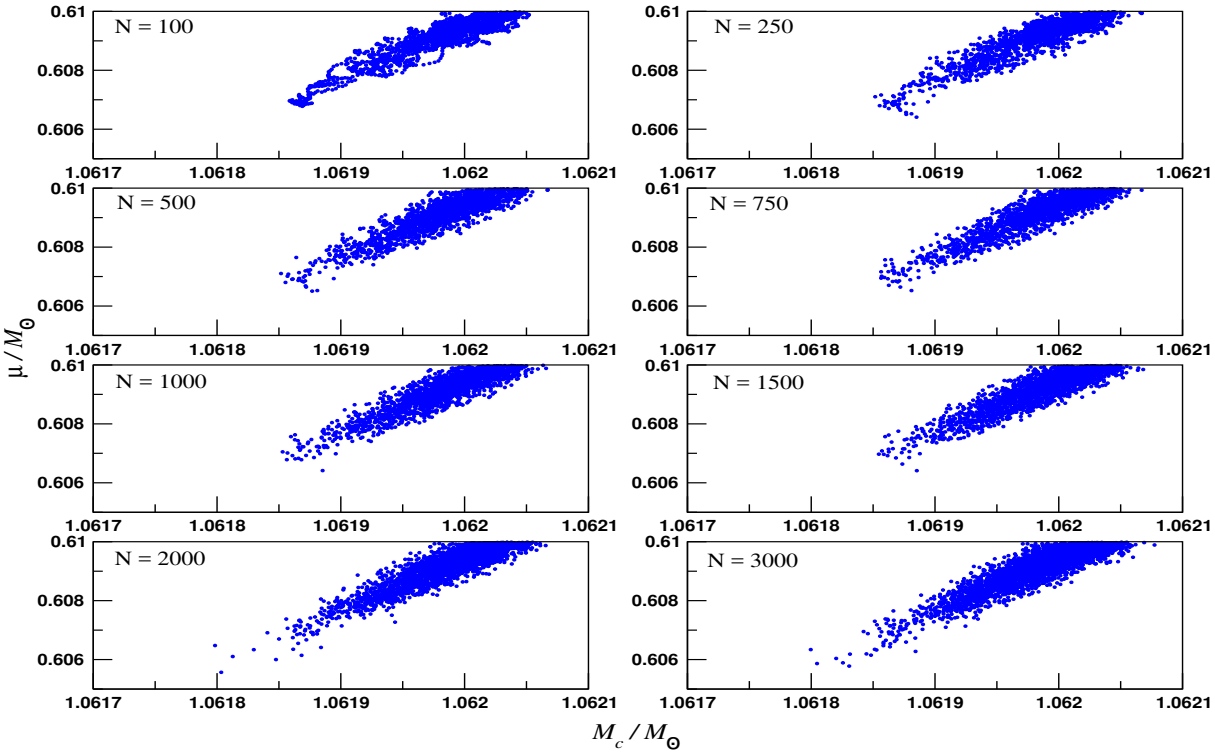

Figure 8.6: Two-dimensional surface plot in $\{\mathcal{M}_c, \mu$ showing the distribution of the M points for the fitting depending on the number N of initial numerical trajectories. The masses parameters are expressed in solar masses $M_\odot$.

the approximate gradient. As we have seen in Chapter 6, these three parameters are connected to the multi-modal nature of the log-likelihood. Thus it seems that a cubic fit approximation is not able to properly represent the gradients for these parameters.

The first test we conducted was to investigate the values for the gradients on a single trajectory and compare the values obtained when using either numerical or approximated gradients. To do that, we ran the code for a single trajectory of $l = 800$ steps using numerical gradients to compute the dynamics in Hamilton's equations. By comparing, the Hamiltonian at the end points of the trajectory, we checked that this trajectory is accepted by the code and thus should be a good representation of a trajectory of the algorithm. At each point of the trajectory, we also computed what would have been the value given by the approximate gradient, using the fit from 750 initial numerical trajectories. In Figure 8.8, we plot the values of the numerical and analytical gradients at each point of the trajectory.

The first thing we observe is that the approximate gradients for the set of parameters $\{\ln \mathcal{M}_c, \ln \mu, \phi_c, \ln t_c, \sin(\theta), \phi\}$ are a good visible match to the numerical gradients. This is in agreement with what was observed before, and explains why the acceptance rate stays constant when using the cubic fitted gradients in Figure 8.7.

Now in the case of the set of parameters $\{\cos \iota, \psi, \ln D_L\}$, we see discrepancies between the approximate and numerical gradients explaining the decrease in acceptance rate. For $\cos \iota$ and $\ln D_L$, we see that the period of the oscillations for the fitted gradients is the same than the ones with the numerical gradients but the amplitude do not match. In addition, we clearly see the correlation between these two parameters on the values of their gradients. For ψ, even the oscillatory structure of the fitted gradient is not in concordance with the numerical gradients. However, we have seen in Figure 8.7 that the decrease in acceptance rate for ψ was limited compared to $\cos \iota$ and $\ln D_L$. Thus it seems that the impact of the gradient for ψ is limited even though the fitted gradient is worse.

8.3.4 Tested solutions beyond the cubic fit approximation

We have seen that the cubic fit approximation was not sufficient to properly model the gradients on of the log-likelihood for $\{\iota, \psi, D_L\}$. To find a solution for these parameters, we have tested a variety of methods,

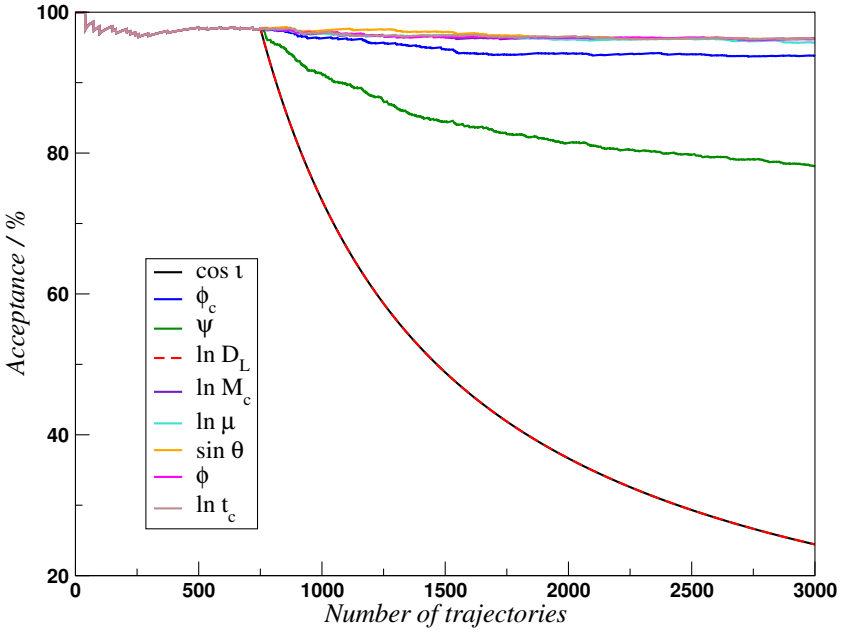

Figure 8.7: Acceptance rate depending on the number of trajectories for a set of simulations where we ran $N = 750$ initial numerical trajectories and then used fitted gradient with respect to the parameter denoted in the legend while keeping the other eight gradients computed numerically. For $\cos\iota$ and $\ln D_L$, the curves lie on top of each other and none of the trajectories are accepted.

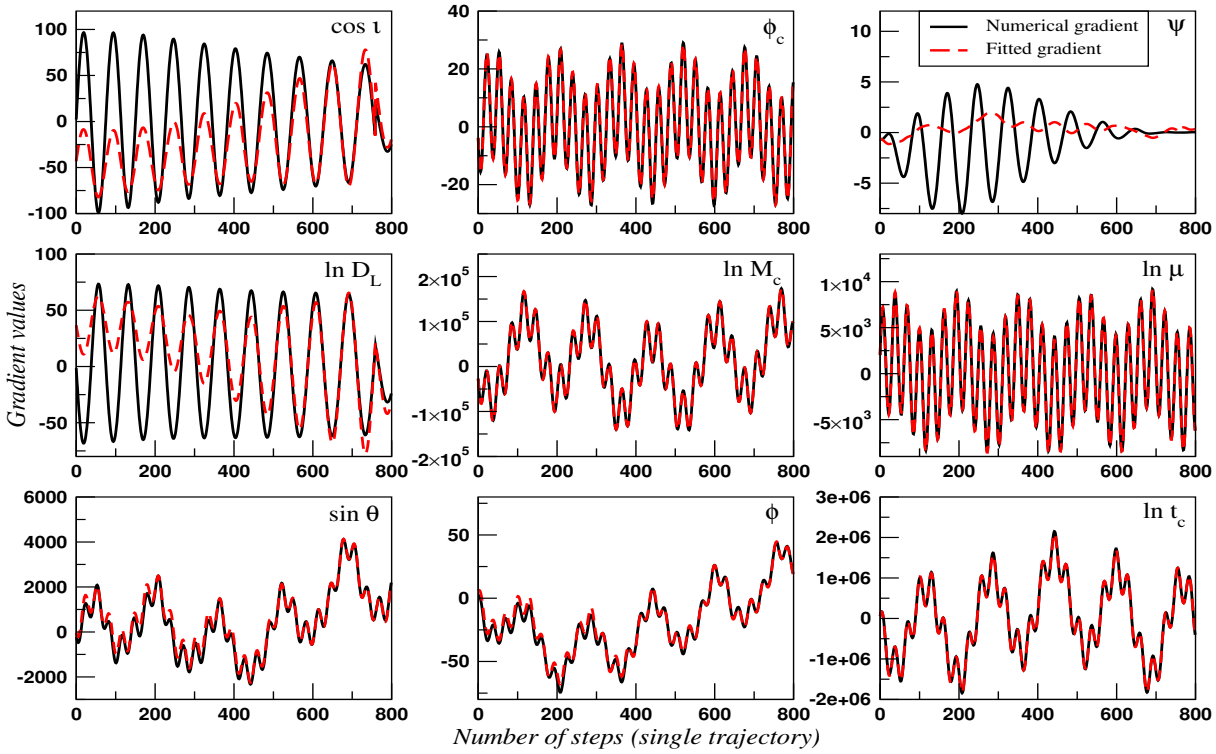

Figure 8.8: Comparison of the numerical and approximate gradients for a single trajectory with $\epsilon = 2.5 \times 10^{-3}$ and $l = 800$. For the cubic analytical gradient, we have used the coefficients derived from the set of points generated with $N = 750$.

1. Higher-order polynomial fit

2. Splitting the pool of points in two sets depending on the value of $\cos\iota$

3. Radial basis functions fit.

In the following section, we present the main results obtained using these various methods.

8.3.4.1 Higher-order polynomial fit

In the case of the cubic fit, the gradients were approximated by a function given in Eq. (8.7). A possible improvement for the fit would be to increase the order of the polynomial fit up to quartic and quintic order for the fit. In this case, the number of coefficients increases to 715 for a quartic fit and 2002 for a quintic fit. However, by increasing the number of coefficients for the fit, we increase the size of the Jacobian matrix and the computation time time required to solve the system of equations with the singular value decomposition.

Since we wanted to have a broad study of the goodness of higher order polynomial fits, used a range of initial numerical trajectories during Phase I. As for the cubic fit, we present in Figure 8.9 the acceptance rate for a simulation of 10^5 trajectories for the quartic (left panel) and quintic fit (right panel), and present the computation time associated in Table 8.3. First of all, we highlight here that some of the simulations failed due to memory limitations. The size of the Jacobian matrix is much larger for higher polynomial fits and the cluster used to run the simulations had an upper memory limit of 3 Gigabytes. This is the reason why only seven simulations are represented for the quartic fit and only two simulations for the quintic fit.

Looking at the acceptance rate for the quartic fit, we observe that the acceptance rate is smaller than in the cubic case. Even for $N = 2000$, the final acceptance is 38.4% which is smaller than the 44.46% we had in the cubic case, and much smaller than the numerical acceptance rate. Similarly, in the quintic fit we do not see improvement and the acceptance rate stays small. In terms of computation time, as expected, the time required to do the singular value decomposition inversion is much larger than in the cubic case. In the quartic case, the time for the inversion becomes even larger than the one required to do the initial numerical trajectories. For instance, for $N = 2000$, the time for the numerical trajectories was 15.67 hours and the time for the quartic fit is 22.45 hours. It is even worse for the quintic fit since for $N = 250$, it takes 1.93 hours for the numerical trajectories and 12.25 hours for the fit. Regarding the average computation time for a single trajectory, we do not see much change compared to the cubic case with an averaged computation time around 45 ms. This means that even for a larger polynomial expression, most of the computation time for a single trajectory is dedicated to the waveform computation necessary for the Metropolis-Hastings ratio.

The conclusion we draw from this study is that increasing the order of the polynomial fit did not improve the quality of the fit.

Fit order	N	t_{PII}/hrs	AR_{PIII}/%	t_{PIII}/hrs	t^{traj}_{PIII}/ms
quartic	100	0.33	5.32	0.86	30.91
quartic	250	1.26	15.53	1.16	41.64
quartic	500	5.98	25.51	1.34	48.16
quartic	750	10.57	33.77	1.35	48.56
quartic	1000	15.82	34.48	1.39	50.18
quartic	1500	20.45	34.96	1.33	47.95
quartic	2000	22.45	38.40	1.38	49.83
quintic	100	2.78	4.81	1.10	39.71
quintic	250	6.69	12.25	1.30	46.91

Table 8.3: Table presenting the values for the total computation time for the fit of the gradient coefficients during Phase II t_{PII}, the acceptance rate at the end of Phase III AR_{PIII}, the total computation time for Phase III t_{PIII} along with the average computation time for a single trajectory t^{traj}_{PIII}. These values are given for the different scenarios where we use N initial trajectories for the quartic and quintic approximation.

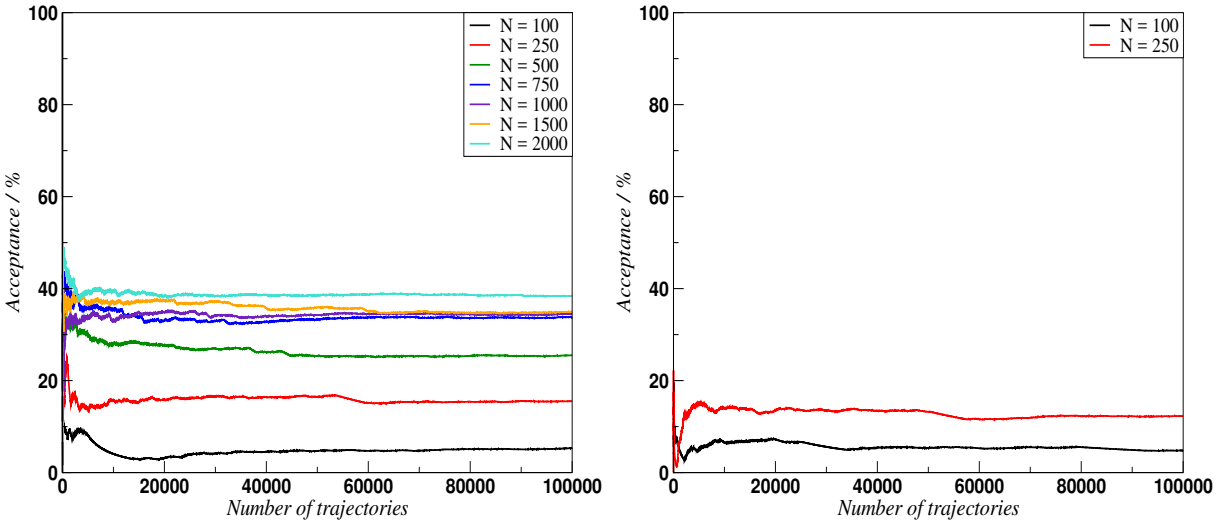

Figure 8.9: Acceptance rate as a function of the number of trajectories when using quartic (left) and quintic (right) fits for the gradients of the log-likelihood in Phase III of the algorithm. These results are presented for different number of initial trajectories N up to the limit for the available memory on the cluster where the simulations were run.

8.3.4.2 Split-fit in inclination

From the previous study, we have seen that the problems with the fit were related to the parameters whose distribution are multi-modal. The option that we considered then, was to split the M points into two sets depending on the value of inclination, where for the first set $\iota < \pi/2$ and $\iota > \pi/2$ for the second set. By doing so, each set of the points represent a mode in the $\{\iota, D_L\}$ posterior distribution. We then used the same cubic fit derived before for each of the set. Since we have seen that the cubic fit was good for the set of the six parameters, we only applied the split for the coefficients of the fit for $\cos(\iota)$, ψ and $\ln D_L$. In terms of the algorithm, this means that we have now three set of coefficients for the fit

- The coefficients for the fit for $\{\ln \mathcal{M}_c, \ln \mu, \phi_c, \ln t_c, \sin(\theta), \phi\}$
- The coefficients for the fit for $\{\cos \iota, \psi, \ln D_L\}$, when $\cos \iota < 0$
- The coefficients for the fit for $\{\cos \iota, \psi, \ln D_L\}$, when $\cos \iota > 0$

Since higher order polynomial fits did not improve the fit, we decided to implement only a cubic fit for the split in inclination. In terms of the algorithm, this implies that each gradient has a fit with three sets of 220 coefficients. Once again, we have used the same set of number of initial numerical trajectories used in the previous studies. In Figure 8.10, we plot the acceptance rate for 10^5 trajectories using the fitting routine described previously. We do not give here the computation time since they were similar to the ones we had in the case where we used a global cubic fit for all the parameters.

What we observe in this study is that overall the acceptance rate is higher than in the cubic case presented in Figure 8.4. For $N = 100$, the acceptance rate is around 18% while it was close to 8% in the previous case. For large number of initial numerical trajectories, $N = \{1500, 2000, 3000\}$, we find that the acceptance rate is close to 58%. This suggests that the quality of the fit was improved by taking into account the multi-modality of the posterior distribution. However, we note that we do not manage to obtain an acceptance rate as high as what we had in the numerical case. Increasing the number of fit points starting from $N = 1500$ did not improve the quality of the fit and the acceptance rate stayed constant at 58%.

In conclusion, we have learned from this study that the fit was improved by considering the multi-modality of the posterior distribution. In order to take this idea further, we have decided to implement fit methods where only the points in the locality of the point where we want to evaluate the gradient.

8.3.4.3 Radial basis functions

At this point of the study, we understood that the fit needed to be treated locally for the problem parameters. In fact, the results from the split fit indicated an overall improvement even if the acceptance

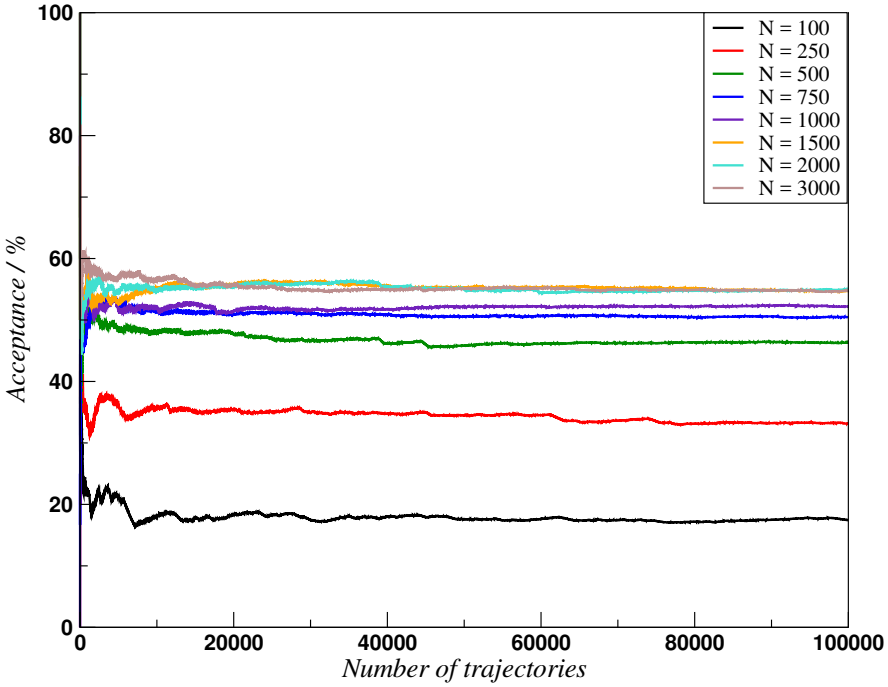

Figure 8.10: Acceptance rate as a function of the number of trajectories when using cubic fit approximation for the gradient of the log-likelihood for $\{\ln \mathcal{M}_c, \ln \mu, \phi_c, \ln t_c, \sin(\theta), \phi\}$ and a split cubic split-fit approximation depending on the value of $\cos \iota$ for $\{\cos \iota, \psi, \ln D_L\}$. These results are presented for different number of initial trajectories N.

rate was still lower than in the numerical phase of the algorithm. This is the reason why we decided to test another fitting method called radial basis functions that we present in this section [117, 118, 119].

Given the set of fit points q_i^μ, the radial basis functions method approximates the gradient at q^μ with the following function,

$$f(q^\mu) = \sum_{i=1}^{M} \lambda_i \phi(\| q^\mu - q_i^\mu \|) = \sum_{i=1}^{M} \lambda_i \phi(r_i), \tag{8.15}$$

where $\| q^\mu - q_i^\mu \| = r_i$ represents the distance between the points q^μ and q_i^μ, ϕ is a functional of the distance r_i and λ_i are weights that need to be determined. The distance $\| \, . \, \|$ and the function ϕ needs to be chosen, and influence the quality of the fit.

For the distance, we have considered two types of distances that were adapted to our 9-dimensional problem. The first distance that we have introduced is the scaled Euclidian distance between two points $\mathbf{x}$ and $\mathbf{y}$ expressed as,

$$\|\mathbf{x} - \mathbf{y}\|_E^2 = \sum_{i=1}^{9} \left(\frac{x^k - y^k}{\sigma_k^2} \right), \tag{8.16}$$

where x^k and y^k are the coordinates of the points $\mathbf{x}$ and $\mathbf{y}$, and σ_k is the component derived from the inverse of the Fisher matrix. The scales σ_k are necessary owing to the differences in dynamical ranges for the parameters already discussed for the mass matrix. With the distance defined in Eq. (8.16), we ensure that we give the same weight in all coordinate directions. The second option we considered is the Mahalanobis distance that is written in matrix form as,

$$\|\mathbf{x} - \mathbf{y}\|_M^2 = \mathbf{x}^T \mathbf{C} \mathbf{y}, \tag{8.17}$$

where $\mathbf{C}$ is the covariance matrix computed using the set of fit points q_i^μ. As for the scaled Euclidian distance, the Mahalanobis distance is well suited to tackle problems with different dynamical ranges.

Regarding the function of the distance $\phi(r)$, we have used functions that are commonly introduced in the literature on radial basis functions. These functions are presented in Table 8.4. They all depend on a number r_0 that is a typical distance for the problem and should be specified by the user. Usually r_0 is taken should be larger than the average distance separation between the fit points. In this study we have decided to set $r_0 = 2\overline{r}$ where $\overline{r}$ is the average distance between the fit points.

Name	$\phi(r)$
multiquadratic	$\left(r^2 + r_0^2\right)^{1/2}$
inverse multiquadratic	$\left(r^2 + r_0^2\right)^{-1/2}$
thin-plate spine	$r^2 \ln\left(\frac{r}{r_0}\right)$
gaussian	$\exp\left(-\frac{1}{2}\frac{r^2}{r_0^2}\right)$

Table 8.4: Typical functions used for ϕ in radial basis functions fitting.

Now, to solve for the coefficients λ_i in Eq. (8.15), we use the values for the gradients y_i^μ associated with every fit point q_i^μ. This results in a system of M equations for each gradient, that can be written in matrix form as,

$$\phi\lambda = \mathbf{y}, \tag{8.18}$$

where we define

$$\phi = \begin{pmatrix} \phi(r_{11}) & \cdots & \phi(r_{1M}) \\ \vdots & \ddots & \vdots \\ \phi(r_{M1}) & \cdots & \phi(r_{MM}) \end{pmatrix}, \lambda = \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_M \end{pmatrix}, \mathbf{y} = \begin{pmatrix} y_1^\nu \\ \vdots \\ y_M^\nu \end{pmatrix}. \tag{8.19}$$

where we have $r_{ij} = \|q_i^\mu - q_j^\mu\|$. To find the solution for the weight λ_{fit}, one then needs to invert the symmetric matrix ϕ to find,

$$\lambda_{fit} = \phi^{-1}\mathbf{y}. \tag{8.20}$$

Using this set of coefficients, one can then compute the gradient at each point of the trajectory with Eq. (8.15).

The inversion of the matrix ϕ is a critical aspect of radial basis functions. The first issue is related to the ability of inverting the matrix ϕ to solve the system of equations. To ensure that the system is invertible, one can choose a function $\phi(r)$ such that the matrix ϕ is positive definite. For the functions presented in Table 8.4, the functions should theoretically produce a positive definite matrix but could still be numerically unstable. Another issue is related to the computation time for the inversion of the matrix ϕ. In fact, the size of the matrix is $M \times M$ where M is of order $\mathcal{O}(10^5)$. When we performed the global quintic fit described in Section 8.3.4.1, we saw that the computation time to invert the system with a matrix of size $2002 \times M$ was already very high. Furthermore, the size of the matrix becomes too large for the memory of the cluster we used to run the codes on. In order to make the method work, this would require to drastically reduce the number of fit points.

As a consequence, we first decided to test the method using only a fraction of the M points generated with $N = 750$ initial numerical trajectories. Given the constraints on computation time, we found that the limit for the fraction of selected points was of 1 out of 25 points from the set of 146400 fit points. As a consequence, we decided to run simulations where we use 1 out of 200, 100, 50 and 25 points from the original set of 146400 fit points. For the distance, we tested both the scaled Euclidian and Mahalanobis distance, and for the function $\phi(r)$, we tested the four options given in Table 8.4. In all cases, we found that none of the trajectories were accepted during Phase III of the algorithm when using cubic fit approximation for the gradients of $\{\ln \mathcal{M}_c, \ln \mu, \phi_c, \ln t_c, \sin(\theta), \phi\}$ and RBF approximation for the gradients of $\{\cos\iota, \psi, \ln D_L\}$. To further test the RBF approximation, we also did a study where we used approximate gradient for one parameter only from the set $\{\cos\iota, \psi, \ln D_L\}$, while using simultaneously numerical gradients for all other parameters. In this case, we found that the final acceptance rate at the end of Phase III was always below 35% for the three parameters $\{\cos\iota, \psi, \ln D_L\}$.

To investigate why the RBF method failed to approximate the gradients, we first tried to find methods to decrease the computation time required for the inversion of the matrix ϕ so that we could increase the number of fit used for the fit. To do that, we tried to apply conjugate gradient method to invert the matrix ϕ. Since this method requires that the matrix ϕ is positive definite matrix, we tested the positive definiteness of ϕ using Cholesky decomposition tests from GSL [120]. In our case, we found that the matrix failed the test when we take more than 1 out of 100 points from the original set of 146400 fit points. This told us that we both could not use the conjugate gradient method and not trust the matrix inversion for the matrix ϕ using singular value decomposition.

To make the matrix ϕ positive definite, we considered two options. The first option was to multiply the matrix ϕ by its transpose and redefine the system of equations in Eq. (8.18) in terms of the matrix $A = \phi^T \phi$. Theoretically, the matrix A should yield a positive definite matrix but numerically we found that the matrix A also failed the Cholesky decomposition test. In addition, we also considered that the time required to compute the matrix A is higher than the time required for its inversion with singular value decomposition of conjugate gradient method. The second option we considered was to slightly modify the approximation given by the radial basis functions in Eq. (8.15) by adding a linear polynomial function in q^μ as [118]

$$f(q^\mu) = \sum_{i=1}^{M} \lambda_i \phi(r_i) + \mathbf{b}^T q^\mu + b_0, \tag{8.21}$$

where $\mathbf{b}$ and b_0 are 10 extra coefficients needed for the fit. To solve for these coefficients, we also add the following conditions,

$$\sum_{i=1}^{M} \lambda_i = 0, \tag{8.22}$$

$$\sum_{i=1}^{M} \lambda_i q_i^\mu = 0 \text{ for } \mu = 1..9. \tag{8.23}$$

It has been shown that this choice of RBF approximation could improve the stability of the matrix inversion. However, in our case we found that regardless of the choice of distance or $\phi(r)$, this new RBF approximation still yields a matrix that is not positive definite. Given that this other test failed, we concluded that the RBF method could not be usable in our case and tried to find another approximation method.

8.3.5 Local fit with look-up tables

Since the RBF fitting method did not work either, we have decided to use another local fitting method based on look-up tables. We will describe in this section how we built the method to adapt it for BNS parameter estimation.

To illustrate this approximation method, let us take an example where we want to fit the gradient of the log-likelihood with respect to $\ln D_L$ at a position $\overline{q}^\mu = \ln \overline{D}_L$ in the parameter space. To evaluate the gradient at $\ln \overline{D}_L$, we can search through the set of fit points to find the points with value of $\ln D_L$ that are the closest to $\ln \overline{D}_L$. If we only use these points to do the fit, we then only use information in the locality of $\overline{q}^\mu$ which should improve the quality of the fit, as we showed in Section 8.3.4.2 that the split-fit in inclination provided better results for the approximate gradients.

Now to find the set of closest points, we build a look-up table where the fit points are sorted according to their value in $\ln D_L$. We can then use standard bisection method to quickly find the point in the table with the closest value of $\ln D_L$ to $\ln \overline{D}_L$. Once we have the closest point, we can then build the set of local points for the fit. To do that, we decided to select an interval of $n_1 + 1$ points in the sorted table that is symmetrically distributed around the closest point. The value of the parameter n_1 is then a parameter that we need to fine-tune to have the best performance. The main issue with this method is related to the multi-modality of the posterior distribution as represented in Figure 8.11. We represent here the positions of the n_1 points (red) in the $\{\cos(\iota), \ln D_L\}$ two-dimensional surface along with the M points from the initial pool of fitting points (blue) and the position where we want to approximate the gradient $\overline{q}^\mu$ (orange). We see that the value of $\ln D_L$ for the n_1 points is close to the true value. However, since we do not take into account the other parameters in the selection process, the n_1 points are spread on a band in cosine of inclination that covers the two modes of the posterior distribution. In terms of fitting, this means that the local fit will actually use points from the other mode and will therefore not be local in terms of cosine of inclination. However, we have seen in Section 8.3.4.2 that the quality of the fit improved when we use points from a single mode. This means that we need to find improve our selection method to have a pool of fitting points that is really a representation of the locality of $\overline{q}^\mu$.

To do that, we have decided to use a distance criterion to only select the closest point in terms of a distance that we define. From the n_1 set of points, we then build a subset of n_2 points, that is formed of the closest points to $\overline{q}^\mu$ from the n_1 set of points. This leaves us with the question of the definition of the distance. To guide this choice, we have considered the fact that the distance should be a reflection of how close a point is in the subspace $\{\cos(\iota), \psi, \ln D_L\}$. Once again this argument comes from the results of the previous simulations that indicate that the points of the fit should reflect the multi-modality of

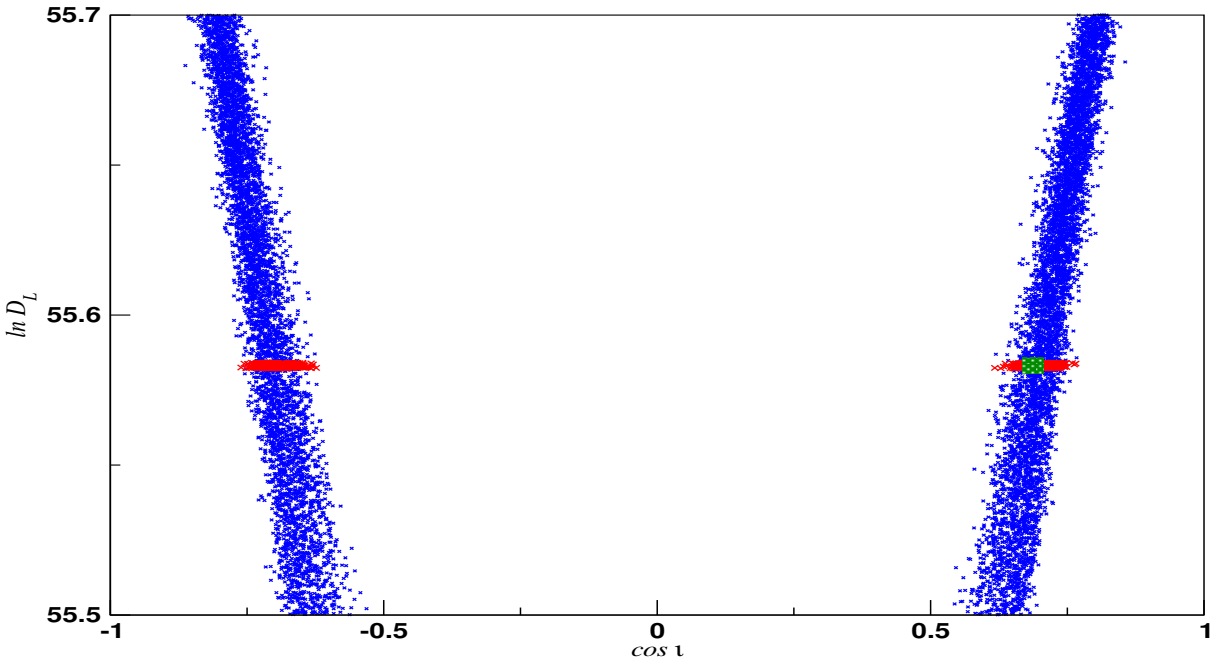

Figure 8.11: Illustration of the problem arising with the selection method of the n_1 points for the fit of the gradient of $\ln D_L$ using look-up table. We plot in the $\{\cos(\iota), \ln D_L\}$ two-dimensional surface the full set of fit points (blue), the n_1 points for the fit (red) and the true solution (green).

the posterior distribution. If we use a distance defined in terms of all the parameter, we will have the same problem than the one we had with the n_1 points where some points would be located on a different mode than the one we are at $\overline{q}^{\mu}$. The other point that needed to be taken into account is the differences in dynamical range that was already discussed when selecting a distance for radial basis functions.

This reflexion lead us to consider three choices for the distances that are closely related to the distances already introduced for RBF, but this time only for the components of the subspace $\{\cos(\iota), \psi, \ln D_L\}$. For the sake of notation, we will fix the coordinate indices of these parameters as,

$$q^0 = \cos(\iota). \tag{8.24}$$
$$q^1 = \psi. \tag{8.25}$$
$$q^2 = \ln(D_L). \tag{8.26}$$

Using this notation, the scaled Euclidian distance between two points q_i^{μ} and q_j^{μ} is given by,

$$\| q_i^{\mu} - q_j^{\mu} \|_E^2 = \left(\frac{q_i^0 - q_j^0}{\sigma^0} \right)^2 + \left(\frac{q_i^1 - q_j^1}{\sigma^1} \right)^2 + \left(\frac{q_i^2 - q_j^2}{\sigma^2} \right)^2, \tag{8.27}$$

where q_i and q_j are two points on the parameter space and σ^2 are the variance derived from the inverse of the Fisher information matrix. Since we already select the n_1 closest points in the parameter we want to fit the gradient, we have decided also to test a scaled Euclidian distance where we only consider only the other two parameters of the subspace $\{\cos\iota, \psi, \ln D_L\}$. Depending on the parameter for which we are fitting the gradient of the log-likelihood, we have three expressions for the distance given for $\cos\iota$, $\ln D_L$ and ψ respectively by,

$$\| q_i^{\mu} - q_j^{\mu} \|_{E'}^2 = \left(\frac{q_i^1 - q_j^1}{\sigma^1} \right)^2 + \left(\frac{q_i^2 - q_j^2}{\sigma^2} \right)^2, \tag{8.28}$$
$$\| q_i^{\mu} - q_j^{\mu} \|_{E'}^2 = \left(\frac{q_i^0 - q_j^0}{\sigma^0} \right)^2 + \left(\frac{q_i^2 - q_j^2}{\sigma^2} \right)^2, \tag{8.29}$$
$$\| q_i^{\mu} - q_j^{\mu} \|_{E'}^2 = \left(\frac{q_i^0 - q_j^0}{\sigma^0} \right)^2 + \left(\frac{q_i^1 - q_j^1}{\sigma^1} \right)^2. \tag{8.30}$$

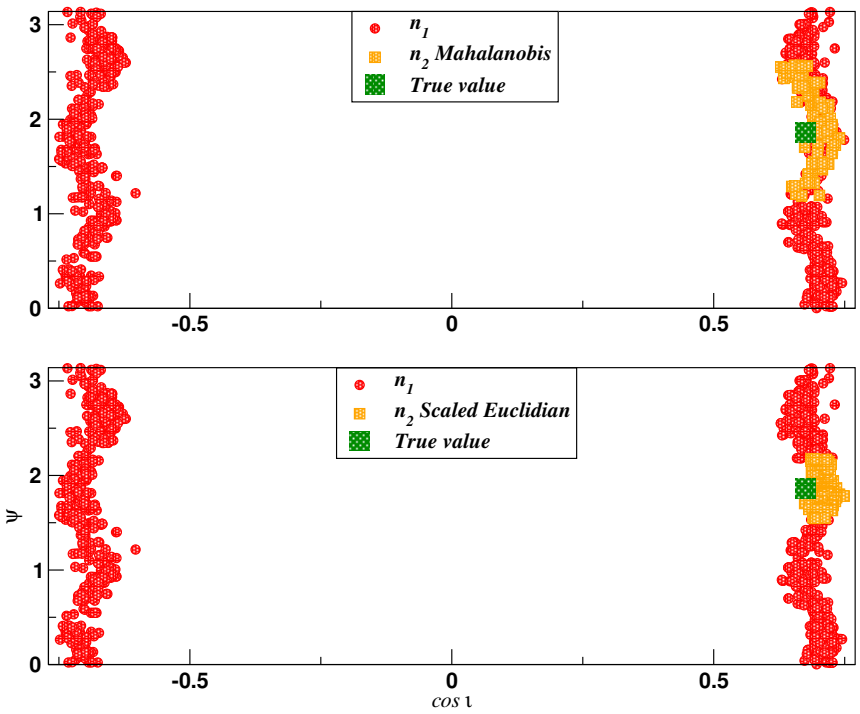

Figure 8.12: Illustration of the selection of the n_2 points to fit the gradient of $\ln D_L$ at the position q^{μ}_{fit} represented in orange. The points are plotted in the $\{\cos\iota, \psi\}$ two-dimensional surface using scaled Euclidian distance $\| \, . \, \|_E$ and $\| \, . \, \|_{E'}$ (top panel) and Mahalanobis distance $\| \, . \, \|_M$ (bottom panel).

Finally, we have also considered the Mahalanobis distance that is expressed in matrix form as,

$$\| \, q^{\mu}_i - q^{\mu}_j \, \|^2_M = \left(q^{sub}_i\right)^T \Sigma q^{sub}_j, \tag{8.31}$$

where $q^{sub}_i = \left(q^0_i, q^1_i, q^2_i\right)$ and Σ is the covariance matrix of the n_1 points in the subspace $\{\cos(\iota), \psi, \ln D_L\}$.

As an illustration of how the selection of the n_2 points works, we present in Figure 8.12 a case where we want to fit the gradient of the log-distance on the true position represented in orange. We plot here the positions of the n_1 points in the two-dimensional surface $\{\cos\iota, \psi\}$ along with the n_2 points using the Mahalanobis distance $\| \, . \, \|_M$ (top panel) and the scaled Euclidian distances $\| \, . \, \|_E$ and $\| \, . \, \|_{E'}$ (bottom panel). We highlight here that the n_2 points given by the two Euclidian-like distances were exactly the same and are thus represented on a single panel. In this example, we have taken the values $n_1 = 1000$ and $n_2 = 100$. The first thing we observe is that for both choices of distances, the n_2 points that have been selected are located on the same mode, which proves that the selection method we introduced works. If we now look at the dispersion of the points, we see that the points are closer to the true position when using the scaled Euclidian distance. However, it is not possible to conclude from this figure how the choice of distance impact the quality of the fit. Finally, since we found that the points were the same with both scaled Euclidean distances, we decided to focus only on distance $\| \, . \, \|_{E'}$ since the computation time in this case is faster than for the other distance.

The final step of the fit is to use the n_2 points we selected before to perform a polynomial fit to compute the value of the gradient. Since we need to redo this fit at each step of the trajectory, we have decided to use only a linear fit with 10 coefficients, since higher order fits would not be computationally affordable.

If we put everything together, the local fit for the set of parameters $\{\cos\iota, \psi, \ln D_L\}$ is described by the following process at each step of the trajectory

1. Find the closest parameter position to the position we currently are using sorted tables,
2. Select $n_1 + 1$ points distributed symmetrically around the closest position in the sorted table
3. Compute the distance between each of these n_1 points and the parameter position we currently are
4. Select the n_2 closest points and do a linear fit to compute the fitted value of the gradient.

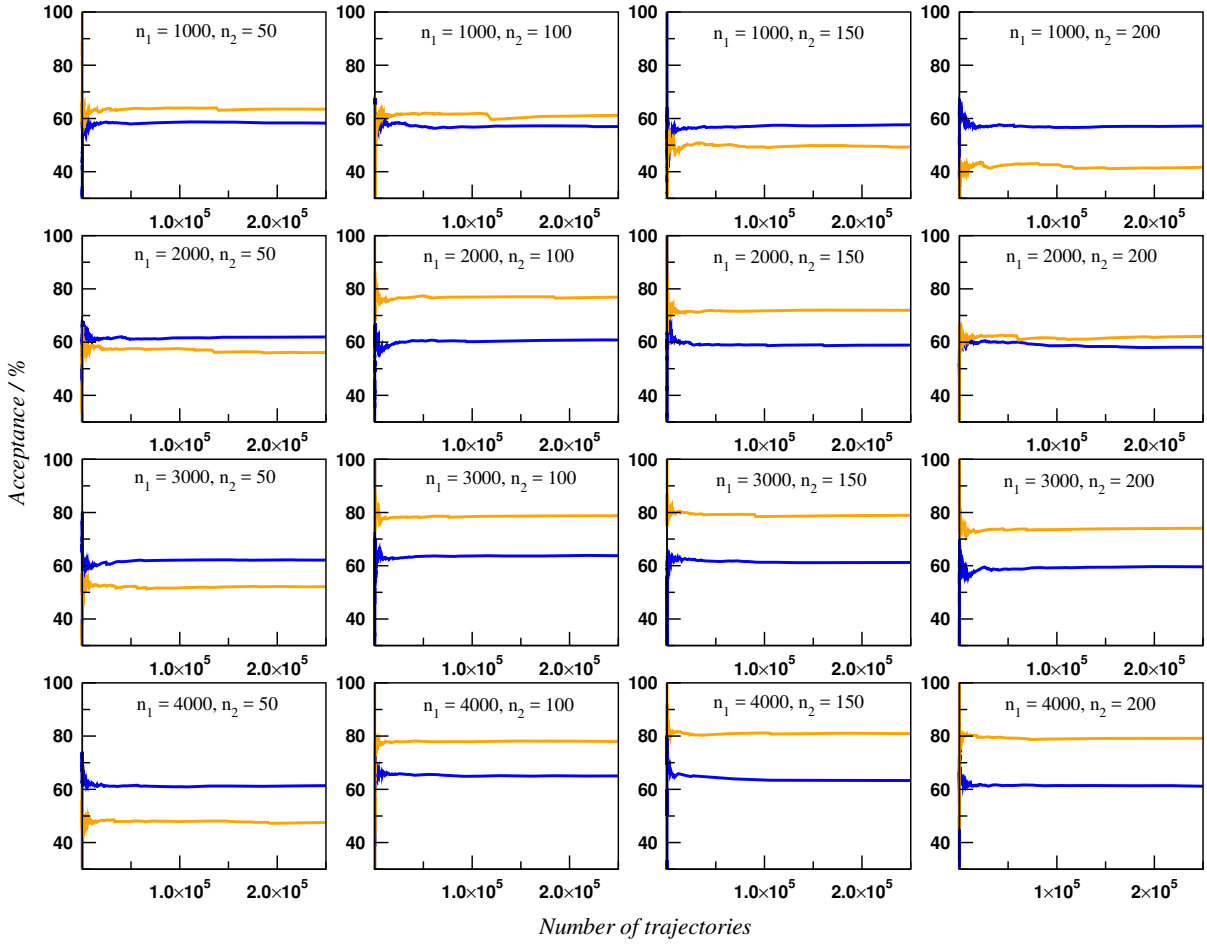

Figure 8.13: Acceptance rate depending on the number of trajectories when using look-up tables fit for the gradients of the log-likelihood of the set of parameters $\{\cos\iota, \psi, \ln D_L\}$ and cubic fit for the other parameters. The initial fit points were generated with 1500 numerical trajectories and different values for n_1 and n_2 have been used for the look-up table fit. These simulations have been done both for the scaled Euclidian distance (orange) and Mahalanobis distance (blue).

There are three parameters that we need to fine-tune in this method: n_1, n_2 and the option for the distance. To do that, we decided to run a series of simulations with different values for the parameters (n_1, n_2) for both type of distances. Each of these simulations were run for 250000 trajectories and we decided to use $N = 1500$ numerical trajectories in order to have a high enough density of points to properly perform the local fit. Once again, we highlight that we used the cubic fit approximation for all other parameters. In Figure 8.13 we present the values of the acceptance rate as a function of the number of trajectories for all the simulations. In Table 8.5 we give the associated computation time for the simulations.

Let us first discuss the results of acceptance rate given in Figure 8.13. In the case where we use the Mahalanobis distance (blue curves), we observe that the acceptance rate is more or less constant regardless of the value of n_1 and n_2 with an acceptance rate close to 60%. This acceptance rate is close to the acceptance rate we had when using the split-fit approximation for the inclination described in Section 8.3.4.2. In the case where we use scaled Euclidian distance (green curves), we see here that we have large variations of the acceptance rate depending on the value of n_1 and n_2. For $n_1 = 1000$, the acceptance rate is less or equal to 60% and drops down to 40% when $n_2 = 200$. For $n_1 = 2000$, we manage to have good mixing of the chain when $n_2 = 100$ with an acceptance rate close to 80%. If we increase further n_1, we also obtain acceptance rates close to 80% if the number of points n_2 is superior or equal to 100. This suggests that the results obtained with the scaled Euclidian distance have the capability of providing better results with acceptance rates as high as 80%.

We will now look at the computation time given in Table 8.5. The computation time related to the

use of the Mahalanobis distance is larger than the computation time with the scaled Euclidian distance. As an example for $n_1 = 4000$ and $n_2 = 200$, the total computation time with the Mahalanobis distance is 33.13 hours while it is equal to 29.18 hours for the scaled Euclidian distance. This result is expected since the Mahalanobis distance requires us to compute the covariance matrix of the n_1 points at each step of the trajectory. Now in terms of n_1 and n_2, we see huge variations of computation time since the code takes around 8.68 hours only for scaled Euclidian distance with $n_1 = 1000$ and $n_2 = 50$ compared to the 29.18 hours with $n_1 = 4000$ and $n_2 = 200$. This means that we need to do a trade-off between acceptance rate and computation time to have the best performances for our algorithm.

The decision we made in regards of these results was to use the combination of scaled Euclidian distance with $n_1 = 2000$ and $n_2 = 100$. In this scenario, we managed to have 80% acceptance rate with a total computation time of 14.77 hours. The associated average computation time for a single trajectory is in this case equal to 213 ms. If we compare this time with the averaged computation time for a single numerical trajectory roughly equal to 30s, this means that the look-up table fit managed to speed-up the algorithm by a factor of 100 while almost keeping the same value of acceptance rate.

n_1	n_2	t_{Euc} (hrs)	t_{Mah} (hrs)	t_{traj}^{Euc} (ms)	t_{traj}^{Mah} (ms)
1000	50	8.68	11.28	125	163
1000	100	10.10	13.91	145	200
1000	150	11.56	16.49	167	237
1000	200	12.89	17.98	186	259
2000	50	13.36	16.21	192	233
2000	100	14.77	18.39	213	265
2000	150	16.29	19.33	235	278
2000	200	20.33	23.55	293	339
3000	50	18.27	22.75	263	328
3000	100	21.75	24.68	313	355
3000	150	23.25	26.08	335	376
3000	200	24.86	28.11	358	405
4000	50	21.98	27.74	316	399
4000	100	26.47	29.25	381	421
4000	150	28.06	30.84	404	444
4000	200	29.18	33.13	420	477

Table 8.5: Table presenting the total computation time for the simulations presented in Figure 8.13 for the different values of n_1 and n_2. t_{Euc} and t_{traj}^{Euc} are the total time and time per trajectory when using scaled Euclidian distance while t_{Mah} and t_{traj}^{Mah} are the total time and time per trajectory when using Mahalanobis distance.

As an additional test for the look-up table fit, we decided to compare the values of the numerical and approximate gradients on the same single trajectory of 800 steps already presented in Figure 8.8. In the left panels of Figure 8.14, we give the values of the gradients of the log-likelihood for $\cos\iota$, ψ and $\ln D_L$ computed numerically and with the cubic fit using $N = 1500$ initial numerical trajectories. As it was already stated in Section 8.3.3 we see that the fit does not match with the numerical values. In the right panels of Figure 8.14, we compare here the values of the numerical gradients with the fitted values using look-up tables with $n_1 = 2000$, $n_1 = 100$ and scaled Euclidian distance. In this case, we see that the values of the fitted gradients are now close to the values of the numerical gradients and this indicates that the fit method we have developed for the set of parameters $\{\cos\iota, \psi, \ln D_L\}$ works properly.

8.4 Handling physical boundaries in parameter space with the HMC

The Hamiltonian trajectory computed with the HMC does not take into account any type of constraints on the physical system. However, we both have physical constraints on the parameters and the range of priors that was described in Chapter 6. We explain here how we have treated these boundaries with the HMC algorithm.

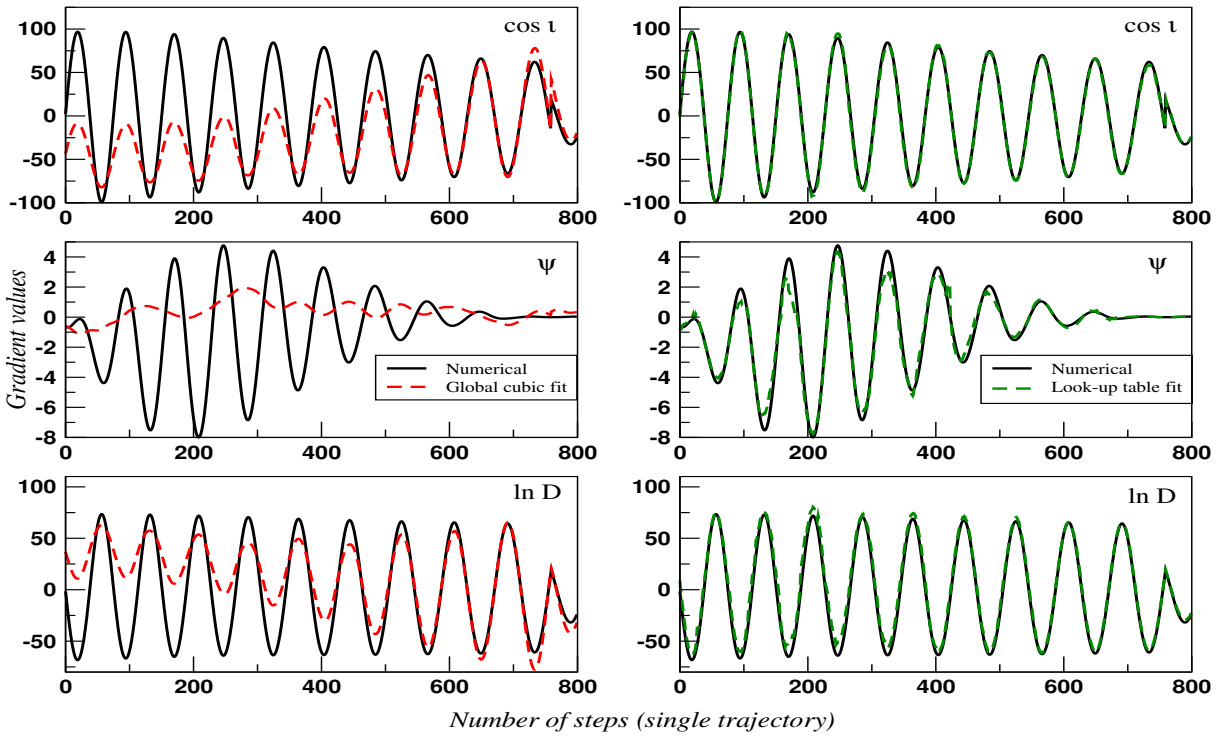

Figure 8.14: Comparison of the numerical and analytical gradients of the log-likelihood with respect to $\{\cos\iota, \psi, \ln D_L\}$ for a single trajectory with $\epsilon = 2.5 \times 10^{-3}$ and $l = 800$ and using the fit points generated after 1500 initial numerical trajectories. The trajectory is identical to the trajectory already presented in Figure 8.8. The right panels present the gradients computed with the cubic fit, while the left panels show the values obtained with the look-up table fit with $n_1 = 2000$ and $n_2 = 100$.

First of all, we re-map the angles ψ and ϕ_c to their natural range at each step of the trajectory. For $\cos\iota$, at each step of the trajectory we checked if the value computed along the trajectory is withing the interval $[-1,+1]$. If not, we re-map the value of $\cos\iota$ such that,

$$-1 \leq \cos\iota \leq 1, \tag{8.32}$$

For the sky angles, we first map the sky angles onto Cartesian coordinates (x,y,z), i.e.

$$\begin{aligned} x &= \sin(\theta)\cos(\phi), & (8.33)\\ y &= \sin(\theta)\sin(\phi), & (8.34)\\ z &= \cos(\theta), & (8.35) \end{aligned}$$

and we map back the set of coordinates (x,y,z) to colatitude and longitude according to

$$\begin{aligned} \theta &= \text{acos}(\phi), & (8.36)\\ \phi &= \text{atan}(y/x). & (8.37) \end{aligned}$$

For the luminosity distance, we have chosen a prior $10^{-6} \leq D_L \leq 200$ Mpc. If a step of a HMC trajectory proposes a position for D_L that is across this boundary, we decided to use a reflective boundary condition where we flip the sign of the the momentum associated with D_L, which reverses the direction of the trajectory. By doing so, we keep D_L in the selected range and at the same time, preserve the time-reversibility of the HMC.

Finally for the masses, we had to deal with two types of boundaries. The first boundary is the prior range for the masses m_1 and m_2. The second boundary is related with the mapping of the masses. Since we use the chirp mass and reduced mass to evolve the trajectory, the HMC algorithm can propose a point in the trajectory where the masses parameters are unphysical and the symmetric mass ratio is superior to $\eta > 0.25$. For both these cases, we have used the same reflective boundary condition as the one for D_L where at each step of the trajectory we check the values of m_1, m_2 and η, and negate the sign of the canonical momenta if the position is outside of the boundaries.

8.5 Final structure of the algorithm

In this section, we summarise the structure of the algorithm that was derived after the benchmarking of the algorithm. The algorithm is divided into three distinct phases:

- Phase I: We run the HMC algorithm using numerical gradients of the log-likelihood as presented in Section 8.3.1. This phase is run for $N = 1500$ trajectories, where for each accepted trajectory we record the positions and values of the gradients of the points inside the trajectory. In this phase, we use a step size $\epsilon = 2.5 \times 10^{-3}$ with a trajectory length $l = 200$. At the end of this phase, we then have a total of M points that are used to fit the coefficients of the approximate gradients.

- Phase II: We first perform the cubic fit using the M points generated during the first phase for the gradients of the log-likelihood with respect to the set of parameter $\{\ln\mathcal{M}_c, \ln\mu, \phi_c, \ln t_c, \sin(\theta), \phi\}$. The solution for the coefficients of the fit are found using a singular value decomposition method. After the cubic fit, we create three $10 \times M$ tables containing the positions of the fit points along with the value of the gradient of the log-likelihood with respect to each of the parameter of the set $\{\cos\iota, \psi, \ln D_L\}$. We then sort these three tables according to each of the three parameters.

- Phase III: We run the HMC algorithm for the remaining 10^6 trajectories, but this time we use approximate values for the gradients of the log-likelihood to compute the trajectories. For $\{\ln\mathcal{M}_c, \ln\mu, \phi_c, \ln t_c, \sin(\theta), \phi\}$, the gradients are computed using the cubic fit approximation with the coefficients computed in Phase II. For $\{\cos\iota, \psi, \ln D_L\}$, we use the sorted tables built in Phase II to approximated the gradients with the local fit approximation detailed in the previous section with the distance , $n_1 = 2000$ and $n_2 = 100$. For the HMC parameters, we keep the same value for the step size $\epsilon = 2.5 \times 10^{-3}$ used in Phase I, but we decrease the value of the length of the trajectory down to $l = 100$. In Phase I, it was important to have a value $l = 200$ in order to generate enough fit points. However, we observed that the exploration of the parameter space with $l = 200$ and $l = 100$ was comparable and yielded a slightly higher acceptance rate in Phase III. Thus, by reducing the value of $l = 100$, we both gain computation time and increase the number of accepted trajectories.

8.6 Results

In this section, we present the results that we have obtained using the algorithm configuration described in the previous section. At this point of the study, the code was tested on the single test system, BNS1. To make a comparison with the DEMC runs presented in Chapter 6, the code was run on a 2.9 GHz Intel i5 processor instead of the cluster used for the benchmarking of the algorithm. The summary of the computation time for all three phases of the algorithm is given in Table 8.6. We highlight here that the computation time in this case is much faster than in the case where the simulations were run on the APC cluster.

t_{PI}/hrs	t_{PI}^{traj}/s	t_{PII}/hrs	t_{PIII}/hrs	t_{PIII}^{traj}/ms	t_{tot}/hrs
2.6	6.24	1.4	18.3	66	22.3

Table 8.6: Computation time associated with the HMC run for BNS1. t_{PI} is the total time for Phase I of the algorithm run for 1500 trajectories and t_{PI}^{traj} is the average computation time per trajectory during Phase I. t_{PII} is the time required for the fitting part of the algorithm during Phase II. t_{PIII} is the total time for the remaining 998500 trajectories in Phase III and t_{PIII}^{traj} is the average time per trajectory during Phase III. Finally t_{tot} is the total computation time of the algorithm.

8.6.1 Exploration of the posterior distribution

The first thing we want to see is how well the HMC is able to explore the difficult parts of the parameter space, especially the bimodality in $\{\iota, D_L\}$. In Figure 8.15, we give snapshots of the HMC chain in the $\{\iota, D_L\}$ two-dimensional surface for 10^2, 10^3, 10^4, 10^5 and 10^6 trajectories. The major difference between the HMC and the DEMC chain presented in Figure 6.4 is that the HMC is able to explore the secondary mode in less than 10^2 trajectories. In the case of the DEMC chain, the secondary mode was only visited after 10^4 trajectories. However, we still note that the HMC is not able to produce sample points in the middle branch even though we have seen that some trajectories are passing in this part of the parameter space.

8.6.2 Convergence of the marginalised posterior distribution

Next, we investigate the convergence of the marginalised posterior distribution of the HMC chain. In Figure 8.16, we plot the posterior distribution of the set of parameters $\{\iota, D_L, \mathcal{M}_c, \mu, \theta, \phi\}$ for 10^3 (orange), 10^4 (red), 10^5 (blue) and 10^6 (black) trajectories. As expected, the posterior distribution for 10^3 trajectories are still peaked and the chain has not converged yet. However, as mentioned before, the general features of the posterior distribution are already present. For instance, we see a major improvement in the posterior of inclination compared to the DEMC chain where the bi-modality is already well defined at 10^3 HMC trajectories only. For 10^4 trajectories, the marginalised posterior distribution start to smoothen and the peakidness is fading away. At 10^5 trajectories, the marginalised posterior distributions are now smooth and we do not observe visible differences with the marginalised posterior distributions obtained with 10^6 trajectories. Furthermore, in comparison with the DEMC distribution, we observe that the marginalised distribution of μ with the HMC chain is shifted towards the true value. We also see that most of the posterior distributions are also tighter compared to the DEMC results. Putting everything together, we find that the HMC chain is already displaying strong signs of convergence at 10^5 trajectories which is much better than in the DEMC case where the marginalised posterior distributions started to exhibit convergence at 10^6 iterations.

8.6.3 Convergence of the instantaneous median

In Figure 8.17, we plot the evolution of the instantaneous median for the HMC chain (blue) along with the instantaneous median for the DEMC chain already presented in Chapter 6 (red) as a function of the number of iterations/trajectories for the nine parameters. We see that the variation scales in the HMC case are much smaller than in the DEMC case. In addition, we observe here that at $\sim 10^5$ trajectories, the median of the mass parameters, the sky angles, the time and phase at coalescence show sign of convergence for the HMC chain. For D_L and ψ,, the median starts to flatten at 10^5 trajectories for the HMC while it still oscillates for the DEMC chain.

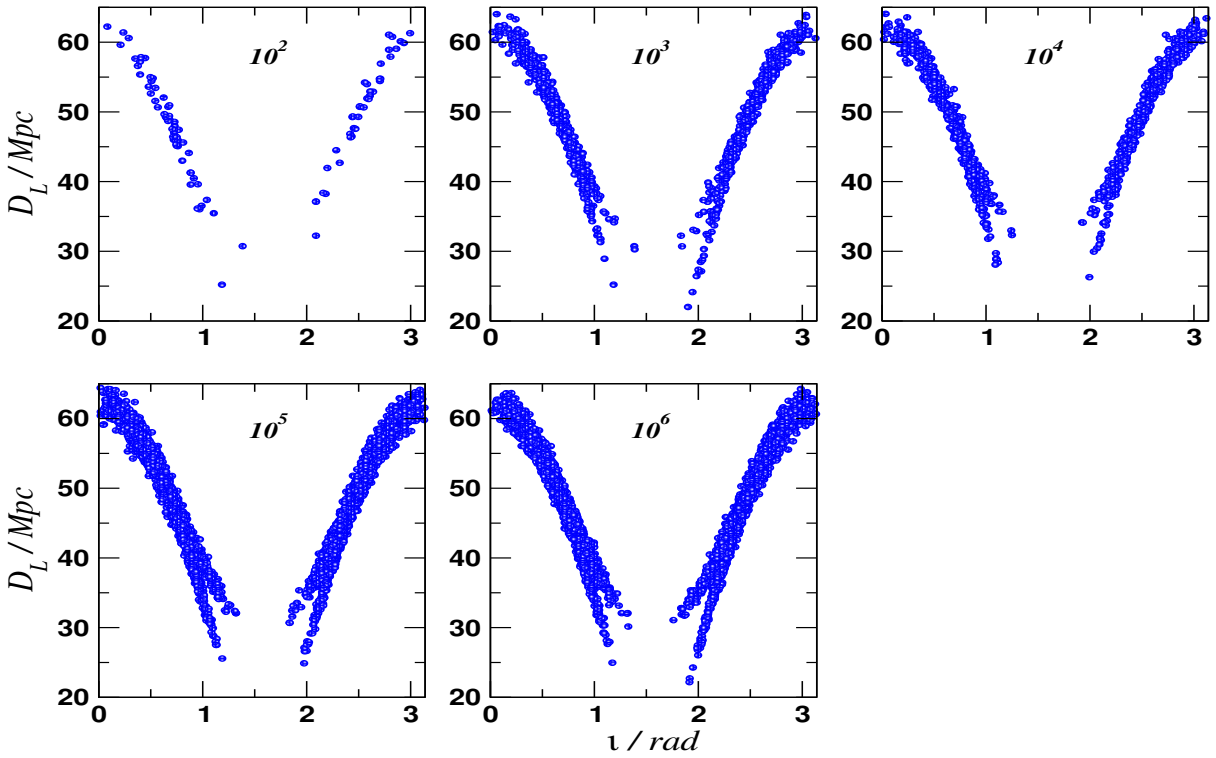

Figure 8.15: Snapshots of the exploration of the 2D $\{\iota, D_L\}$ posterior distribution for BNS1 using a HMC chain in order of magnitude steps for $10^2 - 10^6$ trajectories

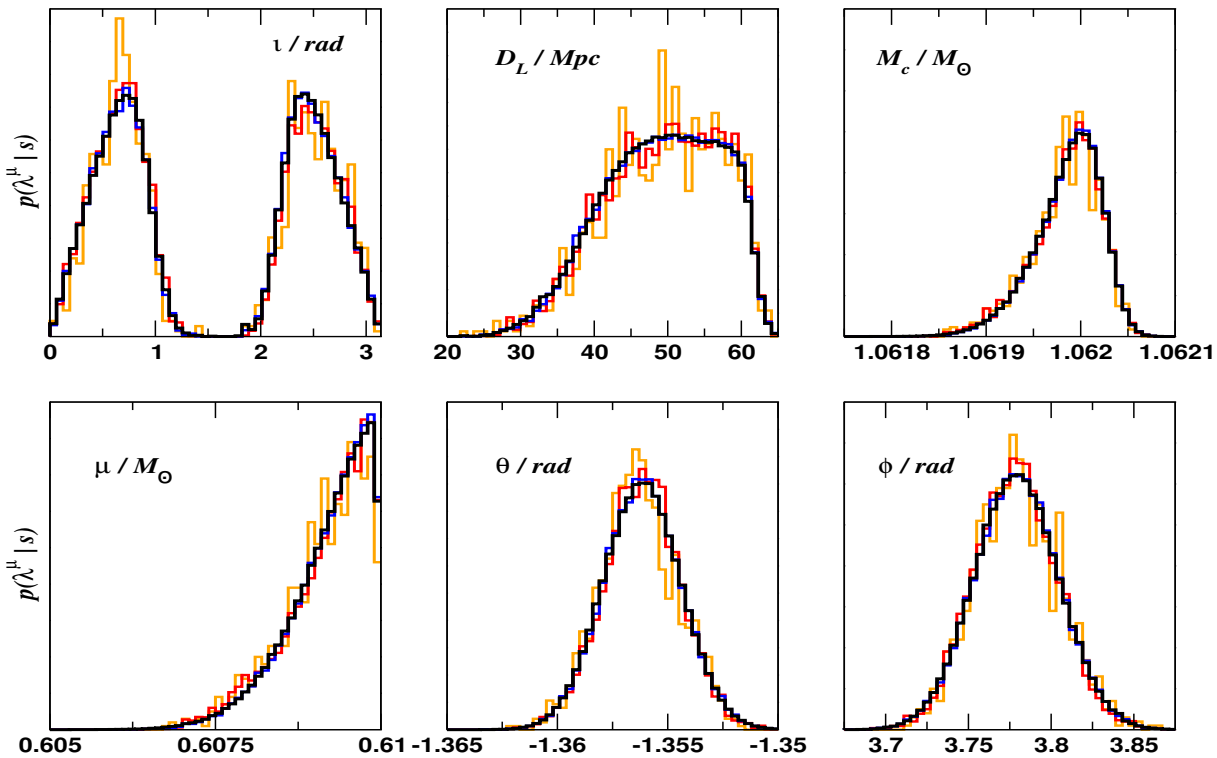

Figure 8.16: Evolution of the marginalised posterior distribution for BNS1, using a HMC chain for 10^3 (orange), 10^4 (red), 10^5 (blue) and 10^6 (black) trajectories, for the parameters $\{\iota, D_L, M_c, \mu, \theta, \phi\}$.

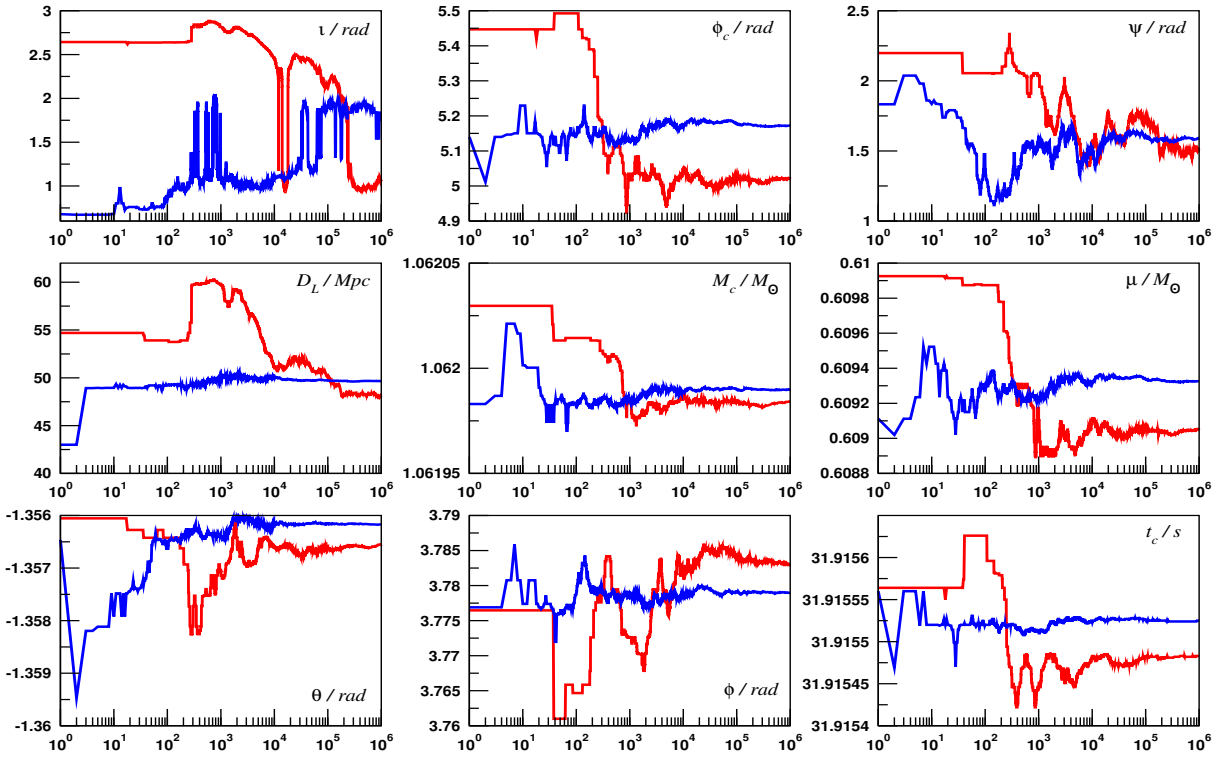

Figure 8.17: A plot of the instantaneous medians for the 10^6 HMC (blue) and DEMC (red) chain for BNS1. The true values are not plotted as we wish to focus on the magnitude of the oscillations in the convergence.

8.6.4 Autocorrelation and integrated autocorrelation time

In Figure 8.18, we plot the values of the chain autocorrelation for the nine parameters as a function of the lag τ. For all the parameters, the autocorrelation quickly falls off to zero for lags superior to $\sim 10^2$, which is an order of magnitude better than in the DEMC case. For the parameters ι and D_L, we even see that the autocorrelation values are the lowest compared to the other parameters and fall to zero at respective lat $\tau_{zac} = 35$ and $\tau_{zac} = 51$. In this case the slowest mixing chain is for ψ where the autocorrelation falls to zero at lag $\tau_{zac} = 155$. In terms of number of independent samples, using the slowest mixing chain of the polarisation angle as a reference, we find that the integrated autocorrelation length has a value of $L = 44$. This means that for the full 10^6 trajectory chain, we have a total of 22727 SISs for a total computation time of 22.3 hours. If we require only 5000 SISs, the required runtime for the algorithm is then approximatively equal to 4.91 hours. Now if we compare with the DEMC results, we find that this corresponds to a factor 10 improvement to the 59 hours required to obtain 5000 SISs.

8.6.5 Parameter estimation

In Figure 8.19, we plot the marginalised posterior distributions inferred from the 10^6 trajectories HMC chain for the nine parameters in blue. In addition, we also replot the posterior distribution obtained with the DEMC chain in red and already presented in Figure 6.10. We see here that marginalised posterior distribution are smoother and less peaked for the HMC chain compared to the DEMC chain. For ι, we observed a slight difference in the heights of the two modes of the posterior distribution suggesting that for this source, there is equal weight to a face-on or face-off solution. Similarly, we find that the multi-modal marginalised distribution for ψ is better represented with the HMC chain. We see that there is a shift of the luminosity distance posterior distribution between the DEMC and HMC chain, and the tails of the distribution are smoother with the HMC chain. There is also differences in the posterior distributions for the mass parameters, especially for the reduced mass distribution that is shifted towards the true value for the HMC chain. We observe the same behavior for the phase and time at coalescence. Overall, these results indicate that the precision of the HMC chain is far better than the posterior distribution obtained with the DEMC, which is in agreement with all the improvements discussed in the previous subsections.

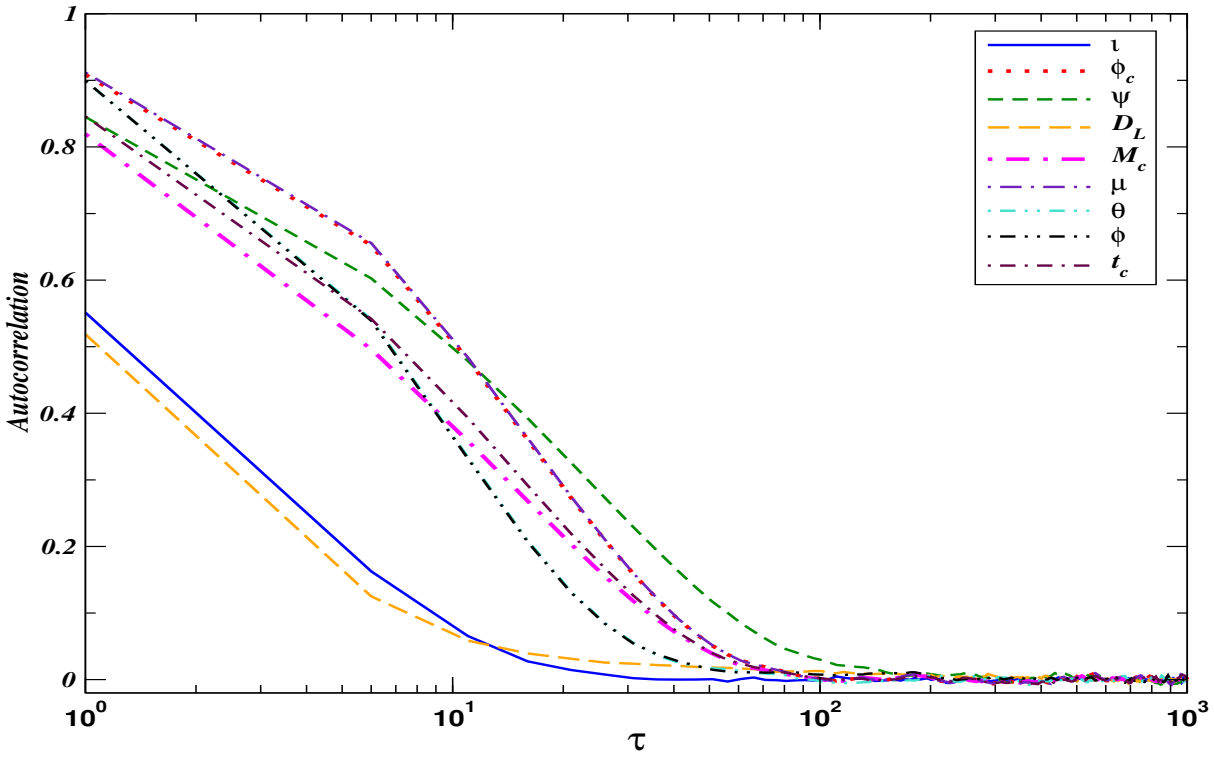

Figure 8.18: Autocorrelation as a function of lag τ for BNS1 using a 10^6 trajectory HMC. The slowest mixing chain in this case is ψ, which has zero autocorrelation at $\tau = 155$.

Finally in Table 8.7, we give the values of the median and 99% credible intervals for the subset $\{D_L, \mathcal{M}_c, \mu, \theta, \phi, t_c\}$ using a 10^6 trajectories HMC. As a reference, we recall the values we obtained with a 10^6 iteration DEMC chain. We highlight that for the HMC all the true values are contained within the credible intervals. In addition, we observe that the widths of the credible intervals are in all cases smaller when inferred from the HMC chain. As mentioned before, we see that the median of μ is higher and shifted towards the true value in the HMC case. Finally, we note that the error box for the sky is also smaller using the HMC chain.

BNS	1 (HMC)	1 (DEMC)
D_L/Mpc	43 $49.661^{+15.467}_{-24.064}$	$48.084^{+16.135}_{-26.202}$
$\mathcal{M}_c/M_\odot$	1.06203 $1.06199^{+0.00008}_{-0.00018}$	$1.06198^{+0.00009}_{-0.00017}$
$\mu/M_\odot$	0.609969 $0.609326^{+0.0007}_{-0.0034}$	$0.60905^{+0.00093}_{-0.00342}$
θ / rad	-1.35612 $-1.35617^{+0.0045}_{-0.0045}$	$-1.35655^{+0.00455}_{-0.00455}$
ϕ / deg	3.77689 $3.77896^{+0.0660}_{-0.0660}$	$3.78309^{+0.06658}_{-0.06658}$
t_c / secs	31.9156 $31.91552^{+0.00027}_{-0.00046}$	$31.91548^{+0.00027}_{-0.00045}$
$\Delta\Omega/sq.deg.$	0.022095	0.025093

Table 8.7: True and median chain values for a subset of parameters for BNS1 using a 10^6 trajectory HMC chain. The error estimates on the median values are the 99% credible intervals. We omit values of the inclination ι as the posterior distributions are bi-modal.

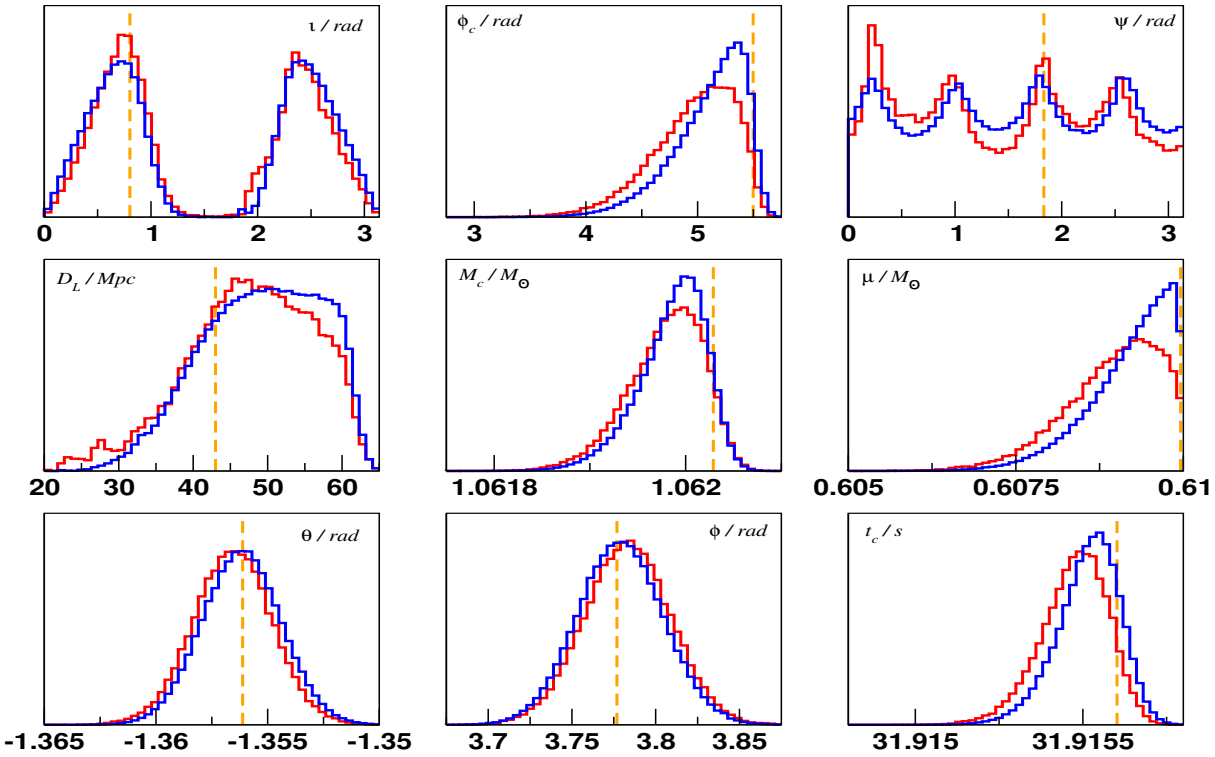

Figure 8.19: Posterior distributions for BNS1 using a 10^6 trajectory HMC chain (blue) and 10^6 iteration DEMC chain (red). The true values are represented by the orange dashed lines.

Chapter 9

Optimisation of the HMC algorithm.

In the previous section, we used a single BNS source to benchmark the algorithm so that the parameters of the HMC, namely the mass matrix $M_{\mu\nu}$, the timestep ϵ and the length of the trajectory, are adapted to the problem to have an efficient algorithm. In addition, we derived a fit method that manages to speed up the computation time required for the calculation of the gradients of the log-likelihood needed for the Leapfrog equations. At this stage of development, we had a working algorithm that produced better sampling than usual MCMC methods for a single BNS. However, we still needed to test how the algorithm performs on other sources and see if it was still as performant as it was for BNS1. To do that, we used the set of BNS sources previously used for the DEMC algorithm.

In Table 9.1, we give the performance statistics of the algorithm in its current state, as applied to the ten binary test sources. As we can see, our algorithm worked quite well for some systems, and very badly for others. In column two, we give the acceptance rates at the end of Phase I. We can see that except for BN2 and BNS3, the acceptance rates were always greater than 97%. For BNS2/3, they were 86% and 52% respectively. These results suggested that there was a possible problem with the scalings s^μ appearing in the leapfrog equations. The effect of this was that for these two sources, we only accumulated 254000 and 78000 data points for the fit, whereas we had closer to 3×10^5 fit points for the other parameters. A combination of a potential scaling issue and the lack of data points meant that the acceptance rates for these two binaries had fallen to 35% and 31% respectively by the end on Phase III. For BNS8 and BNS9, we could definitely see the effect of a scaling problem as the scales for $\ln D_L$ were on the order of $s^{\ln D_L} \geq 10^2$. This caused these chains to get stuck in parameter space, where the runs crashed due to numerical stability. We also saw a problem for BNS10 where our final acceptance rate was on the order of 50%. For the other systems, where the current version of the code worked well, we saw acceptance rates of more that 76%. In the final two columns of the table, we give the integrated autocorrelation times (ACT) and the number of statistically independent samples (SIS). We can see that in the cases where the algorithm works, we are generating between 20000 and 27000 SISs per 10^6 trajectory run.

These problems pushed us to make further changes to the algorithm, that we will discuss below.

9.1 Optimisation of the HMC algorithm

As we saw in Table 6.1, BNS2 is the brightest source of our test systems with a network SNR$\sim$ 86. In theory, this should have been one of the easiest systems to deal with. However, we saw that our algorithm handled this systems quite badly, which led us to concentrate on this binary in order to optimise the algorithm.

9.1.1 Investigating the dynamical scaling factors s^μ

As presented in Section 8.2, we defined the interdimensional stepsizes for the leapfrog equations according to $\epsilon^\mu = s^\mu \epsilon$, where ϵ is taken as a constant. To calculate the values of s^μ, we used two different options

- In the case where we can invert the FIM, $\Gamma_{\mu\nu}$, we use the square root of the diagonal elements of the variance-covariance matrix $C_{\mu\nu}$ to obtain the scalings, i.e.

$$s^\mu = \sqrt{C_{\mu\mu}}, \tag{9.1}$$

  where $C_{\mu\nu} = (\Gamma_{\mu\nu})^{-1}$.

BNS	$AR_{PI}/\%$	t_{PI}/hrs	$N \times 10^5$	t_{PII}/hrs	$AR_{PIII}/\%$	t_{PIII}/hrs	ACT	SIS
1	97	2.66	2.981	1.53	83	17.91	50	20000
2	85	3.70	2.546	0.87	35	17.94^s	*	*
3	26	1.56	0.786	0.22	31	16.62	*	*
4	98	3.50	2.948	1.06	85	18.94	49	20400
5	98	3.59	2.940	1.05	83	18.75	41	24400
6	96	2.57	2.886	0.99	76	18.18	50	20000
7	97	3.56	2.908	0.99	91	18.69	37	27000
8	99	1.54	2.968	1.45	80	C	*	*
9	97	3.56	2.908	0.99	91	C	*	*
10	98	2.39	2.952	1.46	50	17.55^s	*	*

Table 9.1: HMC performance of a set of ten BNS sources. The table lists the Phase I acceptance rates and runtimes, the number of fit points and the time taken to perform the fit, the acceptance rates and runtimes for Phase III, plus the integrated autocorrelation times and numbers of statistically independent samples. C denotes that the run crashed, an asterix denotes that no information is available for these quantities, while a superscript s denotes that the chain got stuck in parameter space and remained there for an excessive number of trajectories.

- When we cannot invert the FIM, for lack of a better option, we simply use the inverse of the diagonal elements of the FIM, i.e.
$$s^\mu = (\Gamma_{\mu\mu})^{-1/2} \tag{9.2}$$

In all of our test cases, the three-detector network FIM was invertible. While this was advantageous, there were situations where the FIM returned the incorrect scalings. This is due to the numerical stability of the matrix inversion and the fact that the FIM is close to being singular. When the FIM failed, it usually produced scalings for $\cos\iota$ and $\ln D_L$ that were larger than their natural scales, i.e. $s^{\cos\iota} > 2, s^{\ln D_L} > 1$. While the scalings for (ψ, ϕ_c) were also larger than their natural scales, these had a much smaller impact on the mixing of the chain.

To properly fix the problem scalings, we tried a number of options, on relatively short HMC runs (10^5 trajectories). As a baseline, we first used the choice

$$s^{\cos\iota} = 1 \quad \text{if} \quad s^{\cos\iota} \geq 1, \tag{9.3}$$
$$s^{\psi} = 1 \quad \text{if} \quad s^{\psi} \geq 1. \tag{9.4}$$
$$s^{\ln D_L} = 0.5 \quad \text{if} \quad s^{\ln D_L} \geq 0.5 \tag{9.5}$$

This motivation for this choice was as follows : For $\cos\iota$, we simply assumed a worst-case scenario that the distribution encompassed the entire region between $\cos\iota \in [-1, 1]$. We then assumed that if this corresponded to six standard deviations, we would take a standard deviation of 1/3 as our scaling. For ψ, we observed that this parameter had little effect in general on the likelihood calculations, so we set the scaling equal of 1/3 of its total range. While for D_L, we assumed a scaling that corresponded to a 50% error in the parameter. This choice worked quite well with the acceptance rate in Phase I rising from 85% to 98%, and the acceptance rate in Phase III rising from 35% to 49%. On investigation, this first study highlighted two potential issues

1. The n_2 fit points that we use from the look-up tables in order to approximate gradients for $\{\cos\iota, \psi, \ln D_L\}$ are supposed to be the 100 closest points to the point that we want to fit. It turns out that while this is true, the fit points are not as close to the evaluation point as we would like.

2. There were still a lack of data points in certain parts of the parameter space which caused the chain to stick (e.g. along the bridge between the modes in $\{\iota, D_L\}$ space, or along the extensions of the posterior distribution to low values of D_L).

9.1.2 Improving the local fit to the gradient

Our first point of investigation was to find is the remaining problems were due to the bad approximations of the gradients. And if so, from where, the look-up tables or the cubic fit approximation. The first option we considered to solve this problem was to keep track of the local acceptance rate during Phase III of the algorithm. If the algorithm failed to produce an accepted trajectory for five consecutive approximate trajectories, we decided to stop using the approximate trajectories and use numerical trajectories instead. As the acceptance rate in Phase I of the algorithm was close to 98%, we knew that the numerical

trajectories should be able to move the chain our of difficult parts of the parameter space. As soon as the algorithm accepted a numerical trajectory, we switched back to approximate trajectories. To accelerate the computation time, we restricted the numerical trajectories to 100 leapfrog steps. While we now obtained an acceptance rate close to 90% in Phase III with this method, we found that the runtime was quite large. This was due to the number of times the chain triggered the call to the numerical trajectories.

To reduce the computation time when the chain got stuck in Phase III, we considered a second option where instead of using full numerical trajectories, the algorithm used a hybrid trajectory, where the gradients of the log-likelihood with respect to $\{\ln \mathcal{M}_c, \ln \mu, \phi_c, \ln t_c, \sin(\theta), \phi\}$ are computed using the cubic fit approximation, while the gradients of the log-likelihood with respect to $\{\cos \iota, \psi, \ln D_L\}$ are computed using numerical trajectories. The motivation for the hybrid trajectories came from the study done on BNS1 where we saw that most of the problems related with approximate gradients were related to these latter parameters. By using this hybrid scheme, our hope was to have good gradient approximations for all parameters, as well as a decreased runtimes as we only need to calculate the numerical gradients for three of the nine parameters.

For our first run with the hybrid trajectories, we set $l = 100$ for these trajectories. While trying to juggle efficiency with speed, we decided that if the approximate trajectories were stuck for ten consecutive trajectories, we would then use ten consecutive hybrid trajectories, up to a maximum of 500 trajectories. The reasoning here was that we would not expect to the the chain stick and call the hybrid trajectories more than 50 times in a run. We also decided that we would use the accepted hybrid trajectories to add data points to the look-up tables to try and improve the gradient approximations for $\{\cos \iota, \psi, \ln D_L\}$. This corresponded to the maximum possible addition of an extra 5×10^4 fitting points. At the end of this run, we found that our acceptance rate at the end of Phase III increased from 49% to 53%. Upon looking at the chains, we saw that there were some instances where the chains still stuck for a number of trajectories. As a test, we increased the number of possible hybrid trajectories from 500 to 10^3, but only used information from the first 500 accepted trajectories to update the look-up tables. In this case, we found that the acceptance rate increased an additional 3% to 56%. On further investigation of this run, we observed that in most cases, the first hybrid trajectory was accepted, meaning that calling ten consecutive hybrid trajectories was both costly and unnecessary in unsticking the chain. We then decided to invoke the call to the hybrid trajectories, if we had more than 5 consecutive approximate trajectories rejected, and then to use only 2 hybrid trajectories to get us unstuck. With this version of the code, our acceptance rate at the end of 10^5 trajectories had increased to 60%. This confirmed our suspicion that the main problem was in the population of the look-up tables.

To try and get around this problem, we decided to increase the number of points in the look-up tables up to a maximum of 3000 accepted hybrid trajectories. This new version of the algorithm increased the final acceptance rate to 67%. At this point, we decided to run full 10^6 trajectory simulations. At 1.7×10^4 trajectories, the acceptance rate was 68%, but very soon after the chain walked into a part of parameter space where it got stuck for $\sim 8 \times 10^4$ trajectories and the acceptance rate started to drop off quickly. By 2×10^5 trajectories the acceptance rate had dropped to 38%, so the run was stopped. An investigation demonstrated that while the gradients for $\{\cos \iota, \psi, \ln D_L\}$ were quite good, the cubic fit approximated gradients for the other parameters had failed, showing that we had not fully solved the problem of data density for the other parameters either. To test a hypothesis, we ran the following setup for the algorithm in Phase III :

- Run approximate trajectories with $l = 100$
- If 5 consecutive approximate trajectories are rejected, run 2 hybrid trajectories with $l = 100$
- If the hybrid trajectories fail to move the chain, use numerical trajectories with $l = 100$ until the chain is free
- Return to using approximate trajectories

This worked very well, with the run having a final acceptance rate of 52%, and no visible parts of the chain getting stuck for excessive amounts of time. The downside to the algorithm was that we required an additional $\sim$ 43000 hybrid/numerical trajectories, which extended the runtimes to $\sim$ 38 hours. We should also highlight that from these 43000 extra trajectories, approximately 14000 of them were full numerical trajectories, which are very costly to generate. The relatively high number of additional hybrid/numerical gradients demonstrated that we still had not done a good enough job in Phase I of populating the parameter space with data for the gradient fit.

The only way to solve this problem was to also use the hybrid/numerical trajectory information to update the set of fitting points for the well behaved parameters. The problem with this is that we

saw previously with the quartic/quintic approximations, the more points we have the larger our fitting matrix is, and the longer it takes for the SVD algorithm to invert the matrix. To try and get around this problem, we took our "tall and skinny" $N \times M$ matrix, where $N \geq M$, A, and tried writing our set of linear equations

$$A\vec{x} = \vec{b}, \tag{9.6}$$

in the form

$$A^T A\vec{x} = A^T\vec{b}, \tag{9.7}$$

or

$$B\vec{x} = \vec{y}, \tag{9.8}$$

where B is the Gram matrix of A, and should theoretically be an 9×9 symmetric, positive definite matrix, which we can invert via a Cholesky decomposition. However, due to numerical instabilities, B was at best positive indefinite, giving us a bad fit for the gradient coefficients.

Our final solution for this problem was to use a QR decomposition to solve the set of linear equations. This works by taking a tall and skinny $N \times M$ matrix A, and transforming it into

$$A = QR, \tag{9.9}$$

where Q is an $N \times N$ unitary matrix, and R is an $N \times M$ upper triangular matrix. The solution of our set of linear equations is then given by

$$\vec{x} = R^{-1}Q^T\vec{b}. \tag{9.10}$$

The main advantage of using a QR decomposition is that we only have to decompose the matrix A once, whereas in the SVD case, it had to be done for each parameter. This reduced the time it took for fit the gradient coefficients from $\sim$1 hour, to around 6 minutes for $\sim 3 \times 10^5$ data points. As the QR decomposition was so fast to run, we made the choice to update the data points used for the gradient fit every time a numerical trajectory is used. Then every 10^5 trajectories, we refit the coefficients of the gradients. Running this version of the algorithm improved the acceptance rate for BNS2 from 52% to 58%.

9.1.3 The Hamiltonian timestep ϵ

As we have seen, making a good choice of the Hamiltonian timestep ϵ is important as the error in a single leapfrog step is of the order $\mathcal{O}\left(\epsilon^3\right)$. In Section 8.2, we saw that if we choose ϵ to be too large, the Hamiltonian is no longer conserved, which results in a low acceptance rate. However, if ϵ or l is small, the Hamiltonian remains almost conserved, but we require many calculations of the gradient of the log likelihood, which slows down the algorithm. For the initial work with BNS1, we decided on a value of $\epsilon = 2.5 \times 10^{-3}$. Now that our algorithm was performing well, we decided to test the effect of variable sizes of epsilon to further optimise the code.

While our fixed value of $\epsilon = 2.5 \times 10^{-3}$ allowed us to develop our algorithm to a very mature level, it is not necessarily the optimal choice. Other research has tackled ways of optimally choosing ϵ but these are more for the development of generic HMC algorithms (see for example NUTs), rather than the specific case of GWs that we are focusing on. Empirically, it was shown that the optimal acceptance rate for a multidimensional HMC is $\sim$ 65%, as opposed to $\sim$ 26% for a standard Markov Chain based sampler [112]. One way of implementing a variable ϵ would be to tie it to the acceptance rate. With this in mind, we tried two different ways of doing this. The first was to tie it to the global acceptance rate. In this case, we changed ϵ according to $\epsilon \to 0.95\epsilon$ every 10^4 trajectories, if the acceptance rate drops below 65%. In this case, once the acceptance rate dropped below 65%, the recovery rate was slow enough for ϵ to become undesirably small, and the chain to embark on a random walk. In the second case, we tied the size of ϵ to the "local" acceptance rate over the last 10^4 trajectories. However, this lead to the same random walk problem. In the end, we rejected linking the size of epsilon to the acceptance rate.

Another option we considered was to draw ϵ from a random distribution [112]. To investigate this option, we first ran chains with fixed step-sizes of $\epsilon = 2.5 \times 10^{-3}$ and 5×10^{-3} on BNS1 for reference (and speed), and a chain where at each step of the trajectory, we chose ϵ from a normal distribution with a mean of 5×10^{-3}, and a standard deviation of 1.5×10^{-3}. We highlight here that when ϵ is generated from a Normal distribution, we fixed a limit such that the value of ϵ is always comprised between 1×10^{-3} and 5×10^{-2}. In all cases, we then compared different quantities such as runtimes and acceptance rates, but more importantly, the autocorrelations, integrated autocorrelation length and the number of statistically independent samples. In Figure 9.1, we plot the values of the autocorrelation as

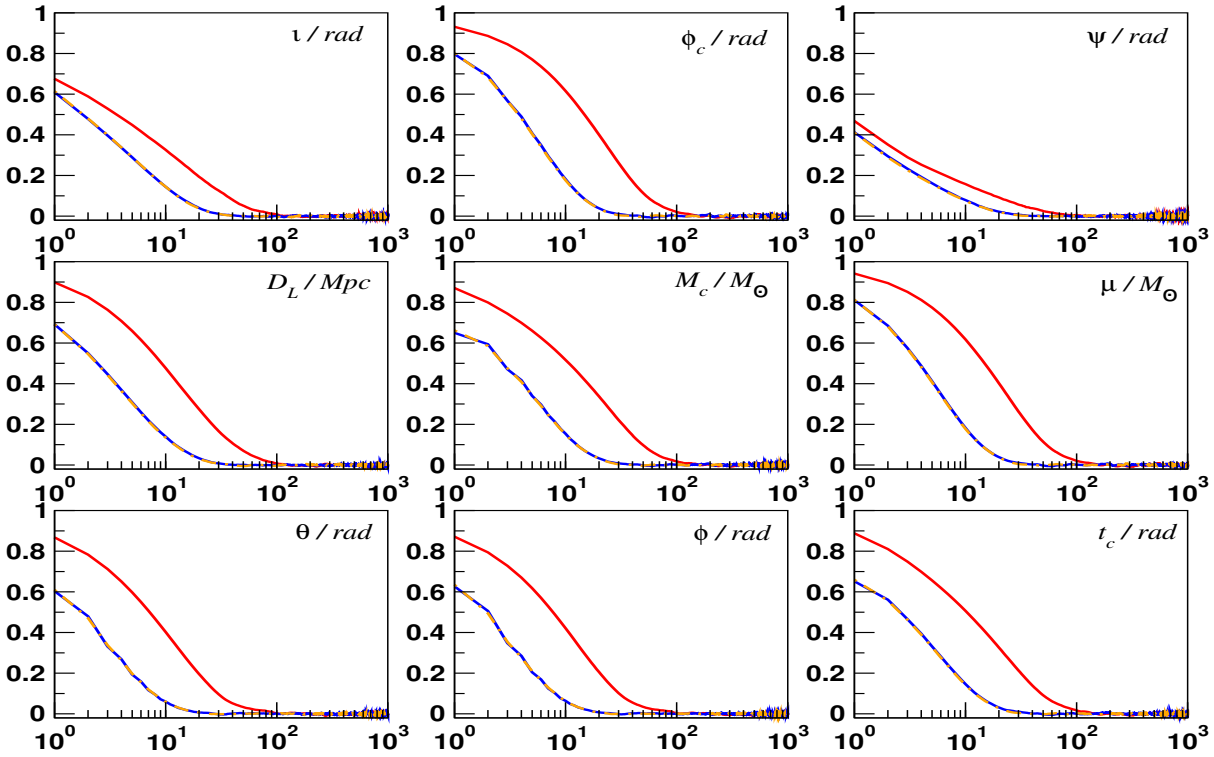

Figure 9.1: Autocorrelations for a 10^6 HMC chain as a function of the lag τ using $\epsilon = 2.5 \times 10^{-3}$(red), $\epsilon = 5.0 \times 10^{-3}$(blue) and random draw from a Normal distribution with mean 5.0×10^{-3} and standard deviation 1.5×10^{-3}(orange). We note that is is difficult to differentiate between the blue and orange curves.

a function of the lag τ for a 10^6 trajectory chain where we use $\epsilon = 2.5 \times 10^{-3}$(red), $\epsilon = 5 \times 10^{-3}$(blue) and $\epsilon \sim \mathcal{N}(5 \times 10^{-3}, 1.5 \times 10^{-3}$(orange) For all tests, the runtimes were very similar which is expected since the number of step trajectory is constant for all the runs even though ϵ varies. Now by comparing the two runs where we used fixed values of ϵ, we observed that while the acceptance rates fell slightly (from 86% with $\epsilon = 2.5 \times 10^{-3}$ to 83% with $\epsilon = 5 \times 10^{-3}$), we see in Figure 9.1 that the autocorrelations fell off almost an order of magnitude faster for all the parameters when we doubled the stepsize. This meant, for BNS1, that for our 10^6 trajectory chain, the integrated autocorrelation time fell from $\tau \sim 50$ to $\tau \sim 13$, meaning that the number of statistically independent samples increased from $\sim 20,000$ to $\sim 77,000$. Drawing from the normal distribution produced a chain with a similar acceptance rate to the fixed $\epsilon = 5 \times 10^{-3}$ run, but with a larger exploration of the parameter space. We also observed a improvement of all metrics, with most importantly, the number of statistically independent samples increasing to $\sim 83,000$ for BNS1. Given the results of these runs, we fixed on drawing the stepsize from the distribution $\epsilon \in \mathcal{N}(5 \times 10^{-3}, 1.5 \times 10^{-3})$. While for BNS2, this increased the acceptance rate from 58% to 62%, the runtime became much longer (∼10 days) due to the number of numerical trajectories being used. As we said earlier, the optimal acceptance rate for a HMC algorithm is 65%. For the final version of the algorithm, we made some further tweaks to achieve this last 3% for BNS2.

The first was to revisit the scalings. As the last choice had worked well, we made a decision to use

$$s^{\cos\iota} = 1 \quad \text{if} \quad s^{\cos\iota} \geq 1 \tag{9.11}$$

$$s^{\phi_c} = \pi, \quad \text{if} \quad s^{\cos\iota} \geq \pi \tag{9.12}$$

$$s^{\psi} = \frac{\pi}{2}, \quad \text{if} \quad s^{\psi} \geq \frac{\pi}{2} \tag{9.13}$$

$$s^{\ln D_L} = 0.5 \quad \text{if} \quad s^{\ln D_L} \geq 0.5 \tag{9.14}$$

which corresponds to a prediction of a 50% error for these four parameters. This lead to the final version of the algorithm defined below

9.1.4 Final structure of the algorithm

- Phase I : We run 1500 trajectories using numerical derivatives. The stepsize for the trajectories are drawn from a normal distribution, $\epsilon = \mathcal{N}(5 \times 10^{-3}, 1.5 \times 10^{-3}) \in [10^{-3}, 10^{-2}]$. In order to accumulate as many fit points as possible, the number of leapfrogs is set to $l = 200$. In general, this produces approximately 3×10^5 fit points. In this phase, the acceptance rate is always greater than 97%.

- Phase II : Once the numerical trajectories are completed, we use the visited points from the accepted trajectories to construct look-up tables for the gradients of the log-likelihood with respect to $\{\cos\iota, \psi, \ln D_L\}$, and use a QR decomposition to create the cubic approximation to the gradients for $\{\ln \mathcal{M}_c, \ln\mu, \sin\theta, \phi, \ln t_c\}$.

- Phase III : In this phase, we swap out the numerical trajectories for the approximate gradient trajectories. In general, this causes a $\sim 10\%$ drop in the acceptance rate of the chain due to non-conservation of the Hamiltonian. As we know that we will have to update the gradient approximations during the run, we allow for a potential 9×10^5 extra fit points. The algorithm is then defined as

  - if $AR \geq 65\%$: use approximate trajectories with $l \in \mathcal{U}[50, 150]$
  - if $50\% \leq AR < 65\%$: use hybrid trajectories with $l \in \mathcal{U}[50, 100]$. Use data from accepted trajectories to update look-up tables.
  - if $AR < 50\%$: use numerical trajectories with $l \in \mathcal{U}[50, 150]$. Use data from accepted trajectories to update cubic fit approximation

  To ensure that the chain keeps moving, we track the number of rejected trajectories. If three trajectories are consecutively rejected, we then use

  - a hybrid trajectory with $l \in \mathcal{U}[20, 100]$ and $\epsilon = 2.5 \times 10^{-3}$. If accepted, use data to update look-up tables, and return to using approximate trajectories. If not, use
  - numerical trajectories with $l \in \mathcal{U}[20, 100]$ and $\epsilon = 2.5 \times 10^{-3}$ until we are no longer stuck. If accepted, use data to update look-up tables and the cubic fit approximation, and return to using approximate trajectories. cubic fit approximation.

  Every 10^5 trajectories, update the coefficients of the gradient fit with the newest accumulated data.

9.2 Results and discussion

In this section we present the results obtained with this version of the algorithm, and compare them with the results obtained with the DEMC chain and the previous version of the HMC algorithm. We will again be focusing on two sources, namely BNS1 and BNS5, but the results for the other binaries can be found in Appendices B and C.

9.2.1 Exploration of the posterior distribution

First of all, we investigate how well the algorithm explores the posterior distribution. In Figures 9.2 and 9.3, we plot snapshots of the posterior distribution exploration of the HMC chain in (ι, D_L) for BNS1 and BNS5 respectively.

For BNS1, as in the previous version of the HMC code presented in the last chapter, we see that the HMC is able to explore both modes only after 10^2 trajectories where the DEMC chain only explores the second mode at 10^4 iterations. However, we observe differences at 10^5 and 10^6 chain trajectories where we see that the latest version of the algorithm is able to have a much better exploration of the middle branch for ι close to $\pi/2$. At 10^6 trajectories, the algorithm has accepted a number of points in the middle branch such that we see a connection between the two modes of the posterior distribution. In the previous version of the HMC algorithm, none of the trajectories were accepted in this region of the parameter space. The HMC sees such parts of parameter space as high energy barriers, which are normally detrimental to algorithm. However, it is clear that our modifications have helped the algorithm overcome such barriers.

For BNS5, we find that only after 10^2 trajectories, the HMC has already visited the two modes of the posterior distribution and is able to capture the global features of the posterior distribution. In addition,

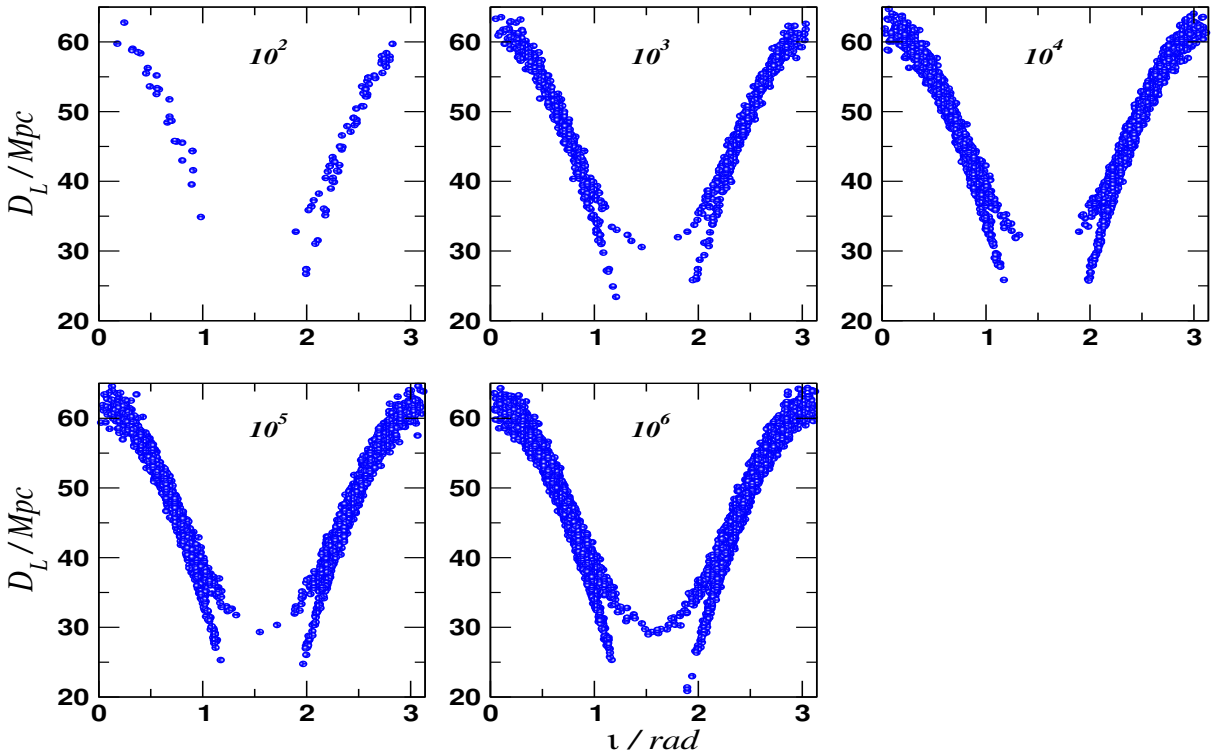

Figure 9.2: Snapshots of the exploration of the 2D (ι, D_L) posterior distribution for BNS1 using a HMC chain in order of magnitude steps for $10^2 - 10^6$ trajectories.

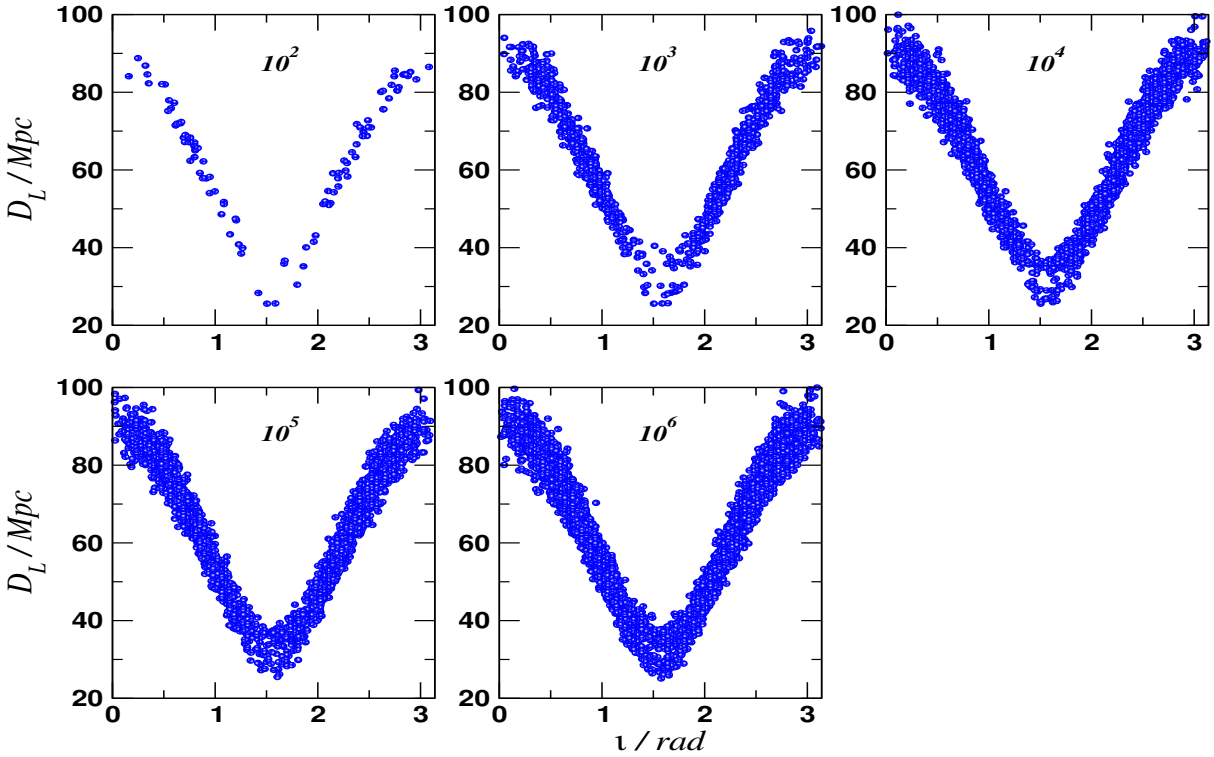

Figure 9.3: Snapshots of the exploration of the 2D (ι, D_L) posterior distribution for BNS5 using a HMC chain in order of magnitude steps for $10^2 - 10^6$ trajectories.

we even see that the two modes are already connected with their middle branch. As a comparison, the DEMC chain needed 10^4 iterations to explore both modes and 10^6 iterations to fill the bridge in the middle. Thus, even if the density of sample points is still not very high, the HMC is able to have a similar exploration at 10^2 trajectories where the DEMC required 10^6 iterations, which represent a huge improvement. As the number of trajectories increases, the HMC is able to have an even better exploration of the posterior distribution, but we do not see major changes indicating that most of the posterior distribution was already visited during the first 10^2 trajectories.

9.2.2 Convergence of the marginalised posterior distribution

In Figures 9.4 and 9.5, we plot the marginalised distributions of the HMC chain of BNS1 and BNS5 respectively for 10^3, 10^4,10^5 and 10^6 trajectories for the subset of parameters $\{\iota, D_L, M_c, \mu, \theta, \phi\}$. For BNS1, if we look at the marginalised posterior distributions inferred at 10^3 trajectories, we do not see a visible difference with the results inferred from the chain with the initial version of the algorithm. However, at 10^4 trajectories we see that the marginalised posterior distributions are already very similar to the ones inferred from 10^5 and 10^6 trajectories, and the difference is even smaller than in the result obtained with the previous version of the algorithm. As with the initial version of the algorithm, this Figure clearly demonstrates just how much faster the HMC posterior distributions converge as compared to those inferred using the DEMC chain.

For BNS5, we see that the features of the posterior distributions are already presented at 10^2 trajectories of the HMC chain. Furthermore, we already see strong signs of convergence at 10^4 trajectories and we do not see visible differences between the marginalised posterior distributions inferred from the 10^5 and 10^6 trajectories long HMC. In comparison, the posterior distributions inferred from the DEMC chain were extremely peaked even at 10^5 and 10^6 iterations. We also observe that for μ, the marginalised posterior distribution inferred from the HMC chain is shifted towards the true value compared to the DEMC marginalised posterior distributions. For D_L, we also see that the HMC is able to capture some of the details of the posterior distribution around $D_L = 40$ Mpc that were not represented with the DEMC chain.

9.2.3 Convergence of the instantaneous median

In Figures 9.6 and 9.7, we plot the instantaneous median as a function of the iteration number using HMC (blue curve) and DEMC (red curve) chains for BNS1 and BNS5 respectively. If we look first at BNS1, we find that the median of the HMC chain again converges faster than the DEMC chain. What is also remarkable is that if we compare this version of the code against the previous version of the algorithm presented in Figure 8.17, we find that our median curves display increased flatness. In fact, for $\{\phi_c, D_L, \mathcal{M}_c, \mu, \theta, \phi, t_c\}$, the instantaneous median are already converging at 10^4 trajectories while in the previous version of the code we could still see some oscillations up to 10^5 trajectories. For ι and ψ, we still observe oscillations at 10^5 trajectories, but this is to be expected due to the multi-modal nature of the posterior distribution. However, we should point out that the scale of these oscillations is reduced compared to the previous version of the algorithm.

For BNS5, we see that the instantaneous median of the HMC chain has essentially converged at 10^4 trajectories for all parameters except ι and ψ. Compared to the DEMC chain, this is a huge improvement since we clearly see that the instantaneous median has not converged at 10^6 iteration for all the parameters. For the other parameters, in general, our final version of the algorithm displays convergence over an order of magnitude faster than the DEMC chain, with the median curves becoming flat after 10^4 trajectories

9.2.4 Autocorrelation and Integrated Autocorrelation TIme

In Figures 9.8 and 9.9 we plot the autocorrelation of the HMC chain as a function of the lag τ for BNS1 and BNS5 respectively. For BNS1, we observe that the autocorrelation of the chain decreases around three times faster than the previous version of the algorithm, and almost two orders of magnitude faster than the DEMC chain. As an example, the autocorrelation of the slowest mixing chain for the DEMC chain fell to zero at $\tau_{zac} = 10450$, while in the previous version of the algorithm it fell at $\tau_{zac} = 155$, whereas for our final version, the autocorrelation chain falls to zero at $\tau_{zac} = 63$. In terms of integrated autocorrelation time, we found that the DEMC chain had $L = 17$ which gave 396 SISs, the first version of the HMC had $L = 50$ giving 20000 SISs, while this version has $L = 17$, giving 58823 SISs. As a result,

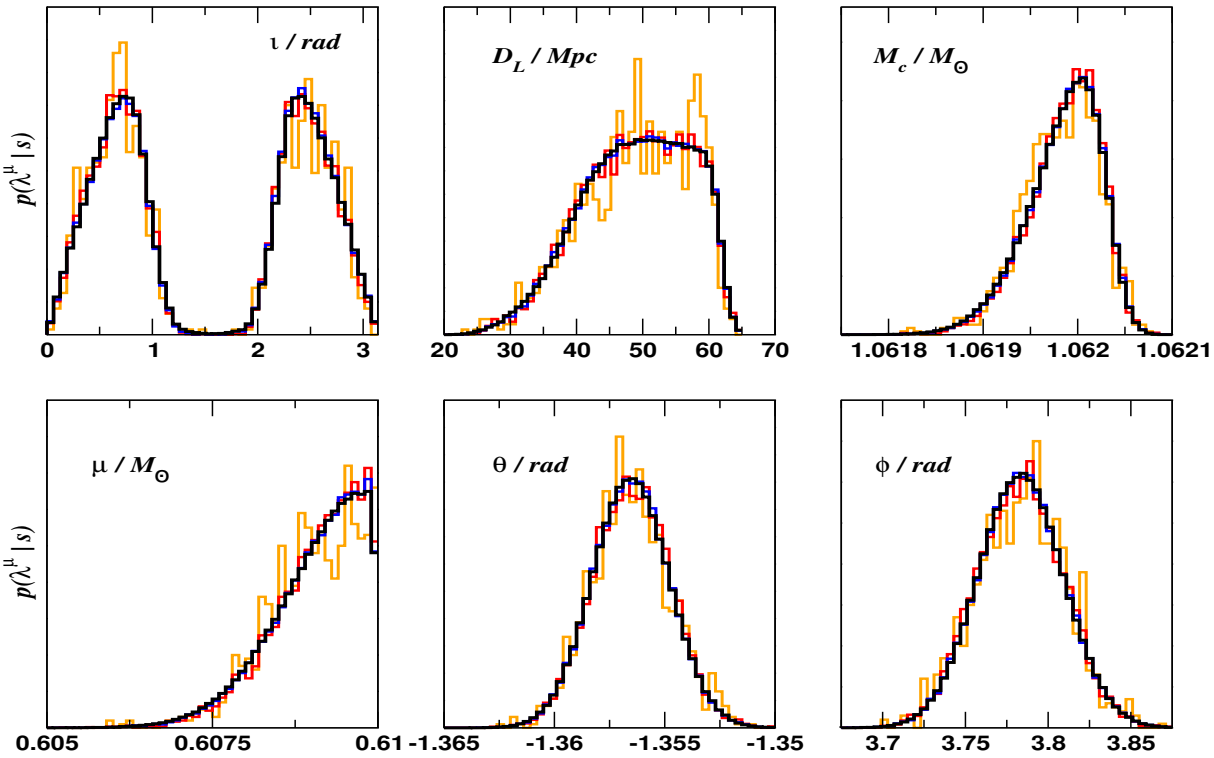

Figure 9.4: Evolution of the marginalised posterior distribution for BNS1, using a HMC chain for 10^3 (orange), 10^4 (red), 10^5 (blue) and 10^6 (black) trajectories, for the parameters $\{\iota, D_L, M_c, \mu, \theta, \phi\}$.

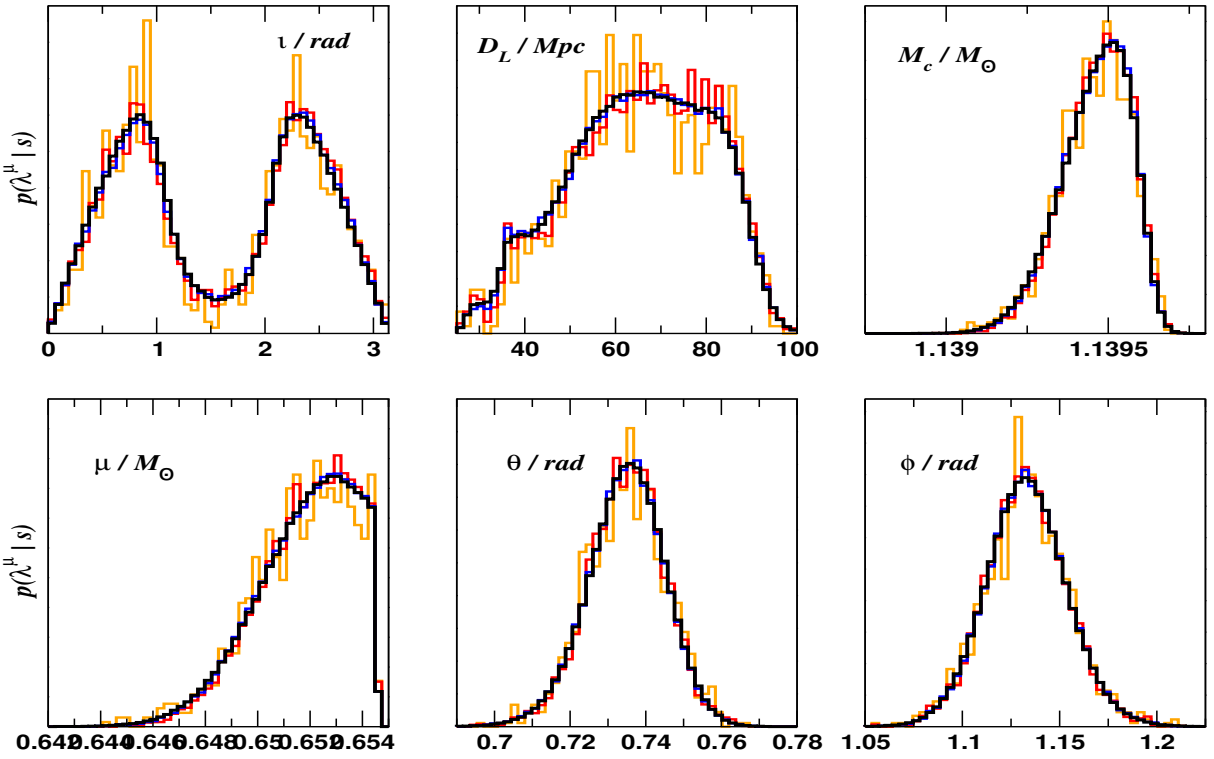

Figure 9.5: Evolution of the marginalised posterior distribution for BNS5, using a HMC chain for 10^3 (orange), 10^4 (red), 10^5 (blue) and 10^6 (black) trajectories, for the parameters $\{\iota, D_L, M_c, \mu, \theta, \phi\}$.

the computation time to obtain 5000 SISs is now 2.36 hours, which is equivalent to 1.70 SIS generated every second of CPU time.
For BNS5, we see that the autocorrelation falls to zero almost 3 orders of magnitude faster than for the DEMC chain. For the slowest mixing DEMC chain, we found that the autocorrelation dropped to zero at $\tau_{zac} = 11902$, where for the HMC chain, the zero autocorrelation crossing happens at $\tau_{zac} = 51$. Again, in terms of integrated autocorrelation, we found for the DEMC chain that $L = 2586$ produced 387 SISs. In contrast, for the HMC, we have $L = 18$ which gives us 55556 SISs. Now, if we compute the associated time to produce 5000 SISs with the HMC chain, we have a computation time of 2.02 hours which is equivalent to 1.45 SIS generated every second of CPU time.

9.2.5 Parameter Estimation

Finally, we give in Figures 9.10 and 9.11 the marginalised posterior distributions of all nine parameters for the 10^6 trajectories HMC of BNS1 and BNS5 respectively. As a reference, we also plot the marginalised posterior distributions of the 10^6 iterations DEMC chains already presented in Chapter 8.

For BNS1, for most of the parameters the marginalised posterior distributions look very similar to the ones found with the previous version of the algorithm. However, we can see some differences for ψ where the probability density between the modes is not distributed identically with a slightly stronger probability density between for ψ close to 0.5 and 2.2. For ι, we also see that there is now slightly higher values for the posterior distribution for ι between 1 and 2. As with the previous version of the algorithm, we still observe the shift of the marginalised posterior distribution of μ towards the true value.

For BNS5, we see that for all the parameters the posterior distributions are less peaked than in the case of the DEMC. In addition, for D_L we observe visible differences in the form of the posterior distribution that is slightly shifted to higher values of D_L. As for BNS1, the posterior distribution of μ is also shifted towards the true value in comparison of the posterior distribution inferred from the DEMC chain.

Finally, we present in Table 9.2 the values of the median of the posterior distributions along with the 99% credible intervals inferred from the 10^6 trajectories HMC chains for BNS1 and BNS5. For both sources, the true values are contained in the 99% credible intervals for all parameters. In addition, we see that the widths of the credible intervals is slighlty smaller than the ones obtained with the DEMC.

BNS	1	5
D_L/Mpc	43 $49.542^{+15.533}_{-25.371}$	72 $65.895^{+39.936}_{-39.936}$
$\mathcal{M}_c/M_\odot$	1.06203 $1.06199^{+0.00009}_{-0.00020}$	1.13951 $1.13947^{+0.00024}_{-0.00047}$
$\mu/M_\odot$	0.60996 $0.60917^{+0.00082}_{-0.00384}$	0.65247 $0.65203^{+0.00251}_{-0.00841}$
θ / rad	-1.35612 $-1.35655^{+0.00449}_{-0.00449}$	0.73653 $0.73515^{+0.03278}_{-0.04623}$
ϕ / deg	3.77689 $3.78338^{+0.06640}_{-0.06640}$	1.13272 $1.13274^{+0.05409}_{-0.05409}$
t_c / secs	31.91560 $31.91550^{+0.00027}_{-0.00049}$	28.38391 $28.38388^{+0.00072}_{-0.00112}$
$\Delta\Omega/sq.deg.$	0.024544	0.265134

Table 9.2: True and median chain values for a subset of parameters for BNS1 and BNS5 using a 10^6 trajectory HMC chain. The error estimates on the median values are the 99% credible intervals. We omit values of the inclination ι as the posterior distributions are bi-modal.

9.3 Conclusion

In this work, we have designed a HMC algorithm for the parameter estimation of BNS sources for the ground-based network of detectors aLIGO/aVirgo. We benchmarked the algorithm on a single BNS source first in order to have the best performance. We first derived optimal values for the free parameters of the HMC algorithm, namely the mass matrix, the step size of the algorithm and the length of the trajectory.

We then found a way to drastically reduce the computation time of the gradient of the target density at every step of the trajectory in order to have an algorithm that is competitive with other MCMC-like sampling algorithms. The first results showed us that the algorithm was capable of performing much better than a DEMC algorithm on the same BNS source. After an upgrading phase, the algorithm was then run on a variety of different BNS sources. We found that the algorithm performed much better than the DEMC in all aspects. As a reference, we found that the time to produce 5000 SISs with the HMC algorithm was in average between 10 and 20 times faster than with the DEMC. And even for the worst BNS source, the HMC algorithm was still capable of providing SISs 5 times faster than the DEMC.

As this works was an exploratory study of the performances of the HMC, all the codes were developed outside of the Ligo/Virgo data analysis pipeline LALInference. This means that it is not directly possible to do an apples-to-apples comparison with the current performances of LALInference. However, we can still approximate how the HMC should perform compared to the samplers used in LALInference. In [68], they state that the CPU time to generate a SIS is around 77.1 seconds using Nested Sampling algorithm for BNS parameter estimation using a TaylorF2 waveform model. In almost all cases, we found that the CPU time to generate a SIS reduces to the order of one second when using the HMC. In addition, we highlight that the code used to compute the waveforms and the log-likelihood are not as optimised as those currently used in LALInference. Algorithms exist within this library that can accelerate the calculation of the log-likelihood by factors of 100s-1000s. We could envisage a similar scale of speed-up once algorithms from this library are adapted into the HMC algorithm developed here. If we now compare the results with the ones obtained in [70], we find that for the parameter estimation of BNS using the TaylorF2 waveform model, their total CPU time for a single run was in average equal to 280 hours for ~ 4500 SISs with an averaged CPU time to generate one SIS of 227 s. The results we had using the HMC algorithm demonstrated that we could gain fact of 10 or more improvement in the total computation time. Once again, we highlight that these numbers should be treated carefully since the HMC algorithm was not tested in the framework of LALInference.

The gain in performances and computation time induced by using the HMC algorithm could first of all be of great importance for electromagnetic follow-up of GW sources. As for now, sky localisation is first computed using BAYESTAR [121] which produces a fast approximation of the sky position generated within seconds that are sent to electromagnetic observatories. More precise values of the sky position is then generated with full Bayesian parameter estimation using LALInference. With the HMC algorithm we developed, the decrease in computation time, and the increased convergence, could improve the follow-up observations of the sources by providing more precise sky localisation to electromagnetic observatories on the order of hours instead of days. In addition, third generation detectors of GW will have a smaller lower frequency cutoff of the detector which then increases the computation time to generate a single GW waveform. With better sampling algorithms, we could then keep the total computation time for data analysis within reasonable values.

Finally, we want to highlight that there are still a number of points that need to be investigated. First of all, the HMC algorithm is currently designed for a three-detector network where the angles of the sky are clustered in a small area of the sky. The posterior distribution with two detectors only has a more complex structure that could make the sampling process more difficult. Secondly, the results presented in this work used designed sensitivities of the detectors for which the average SNR of the BNS sources was around 50. We did not have the time to have a proper investigation of how the algorithm performs with lower SNR sources, but initial tests suggested that the HMC algorithm was still capable of providing very good performances. Finally, we have only used in this study TaylorF2 waveforms with 9 parameters. We would like to test in the future how the algorithm performed when using more complex waveform models that include more parameters like the spin of the two stars. In addition, this study was restricted to the parameter estimation of BNS but it would also be interesting to evaluate the performances of the algorithm on other type of sources formed of two black holes or a neutron star and a black hole.

We conclude with a final comment on the application of this algorithm to BNS2. Using the first version of the algorithm, plus the addition of corrections outlined at the beginning of this chapter, we had arrived at something which had an acceptance rate of 62%, but took 10 days to run. It turned out that this version of the code needed on the order of 400000 extra hybrid/numerical trajectories, of which close to 110,000 were full numerical trajectories. With the final version of the code, our algorithm had a final acceptance rate of 65%. However, it now took 4.5 days, instead of 10 to run. It still needed an addition 270000 extra hybrid/numerical trajectories, but now, only 1007 of those were full numerical trajectories. Our belief is that the the posterior distribution for BNS2 is more complicated than any of the other binaries. This is something we intend to further investigate in the future.

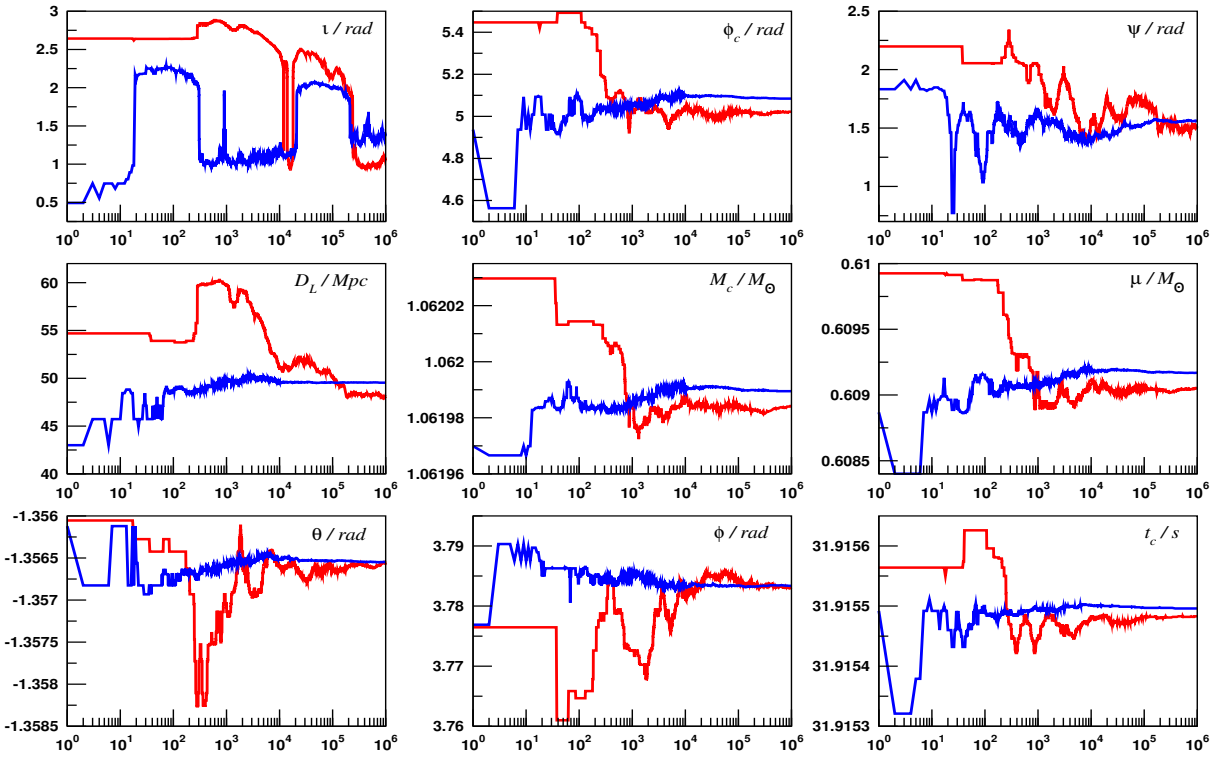

Figure 9.6: A plot of the instantaneous medians for a 10^6 trajectory HMC (blue) and 10^6 iteration DEMC (red) chain for BNS1. The true values are not plotted as we wish to focus on the magnitude of the oscillations in the convergence.

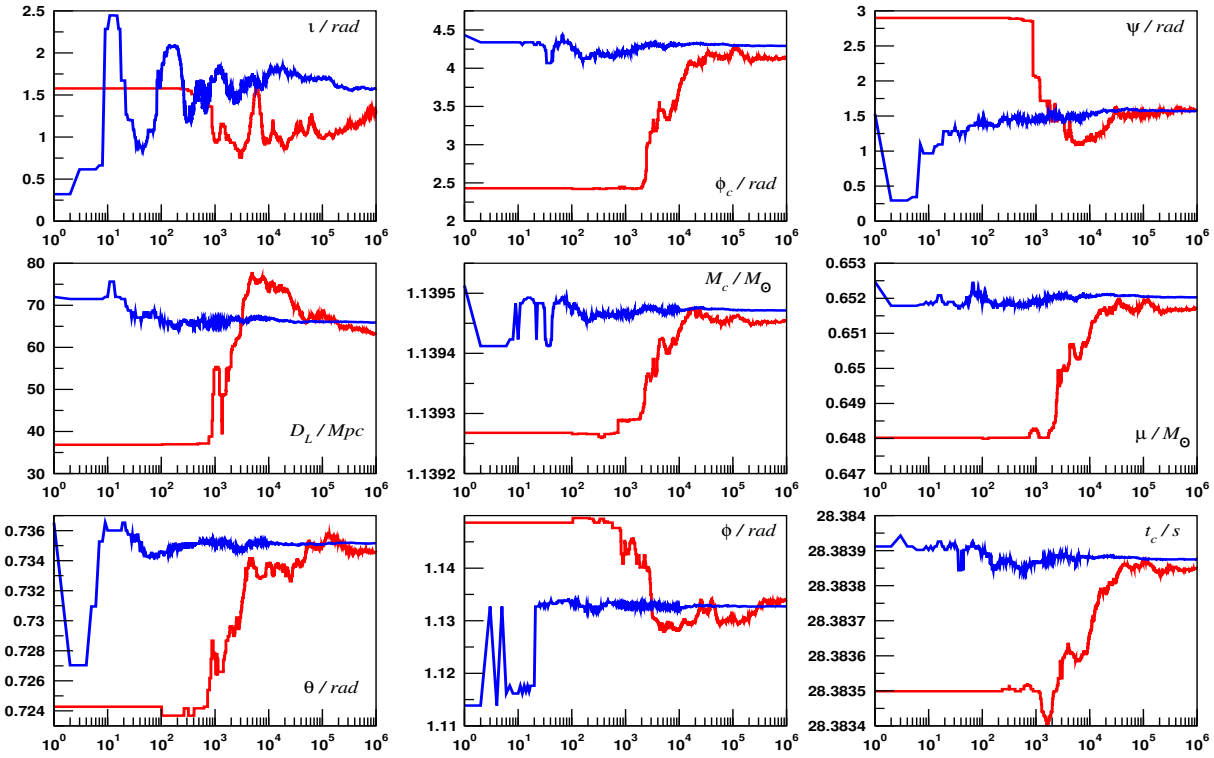

Figure 9.7: A plot of the instantaneous medians for a 10^6 trajectory/iteration HMC (blue) and DEMC (red) chain for BNS5. The true values are not plotted as we wish to focus on the magnitude of the oscillations in the convergence.

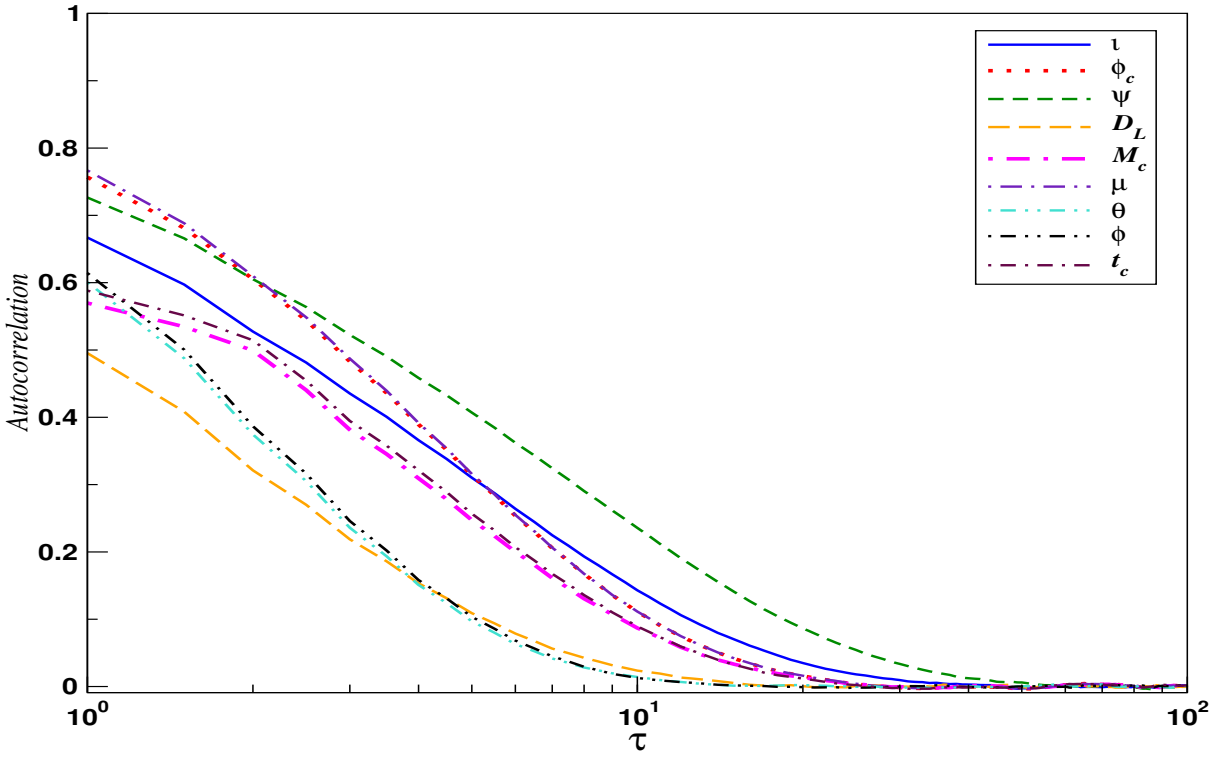

Figure 9.8: Autocorrelation as a function of lag τ for BNS1 using a 10^6 trajectory HMC. The slowest mixing chain in this case is ψ, which has zero autocorrelation at $\tau = 63$.

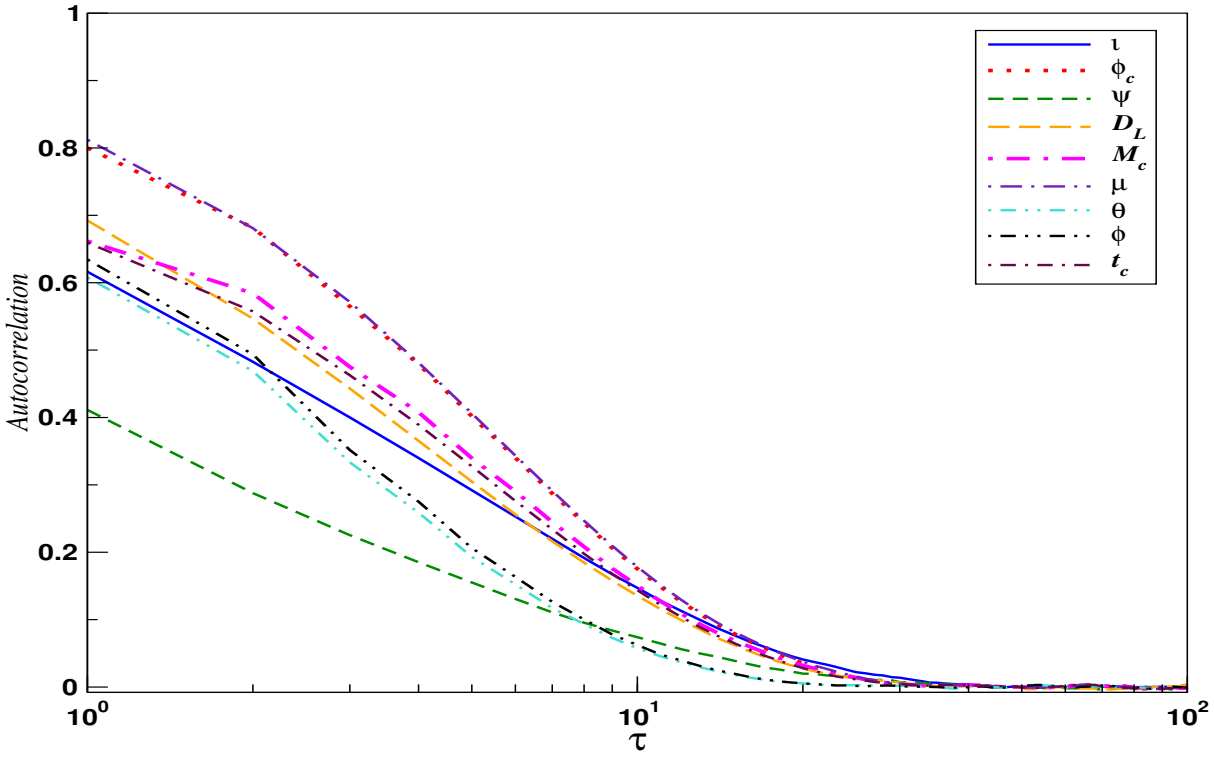

Figure 9.9: Autocorrelation as a function of lag τ for BNS5 using a 10^6 trajectory HMC. The slowest mixing chain in this case is ι which has zero autocorrelation at $\tau = 51$

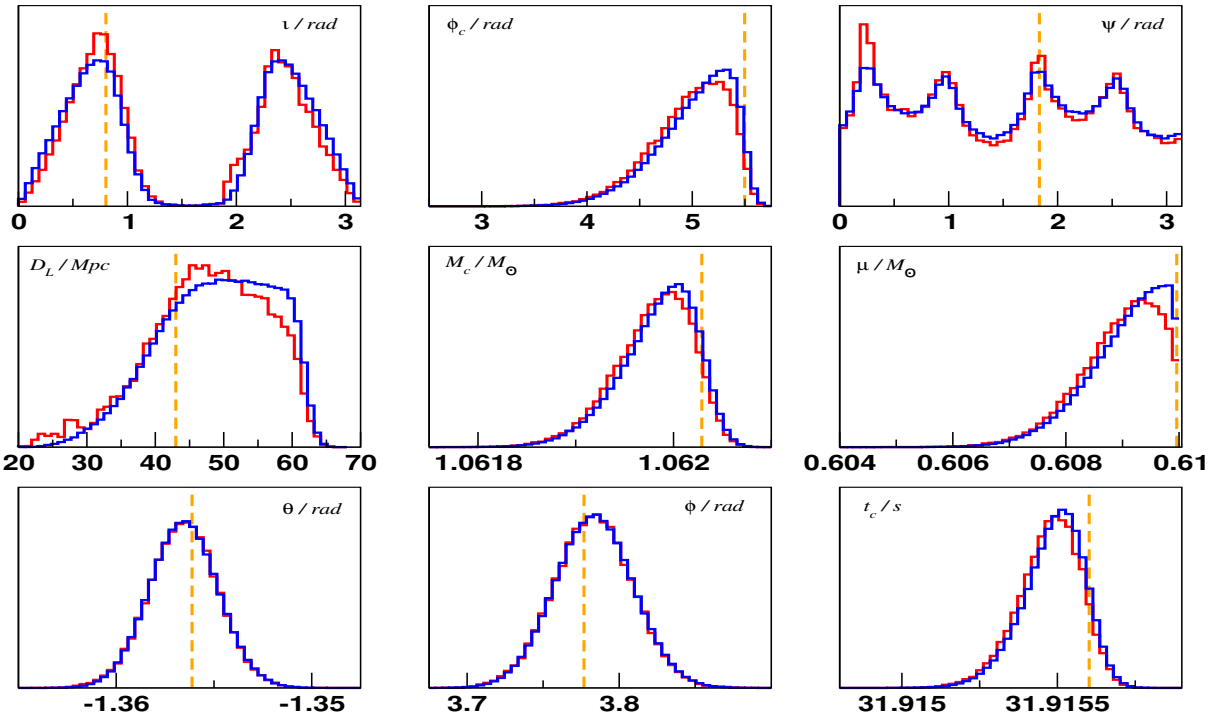

Figure 9.10: Marginalised posterior distribution of the nine parameters for BNS1 using a 10^6 iteration DEMC (red) and HMC (blue).

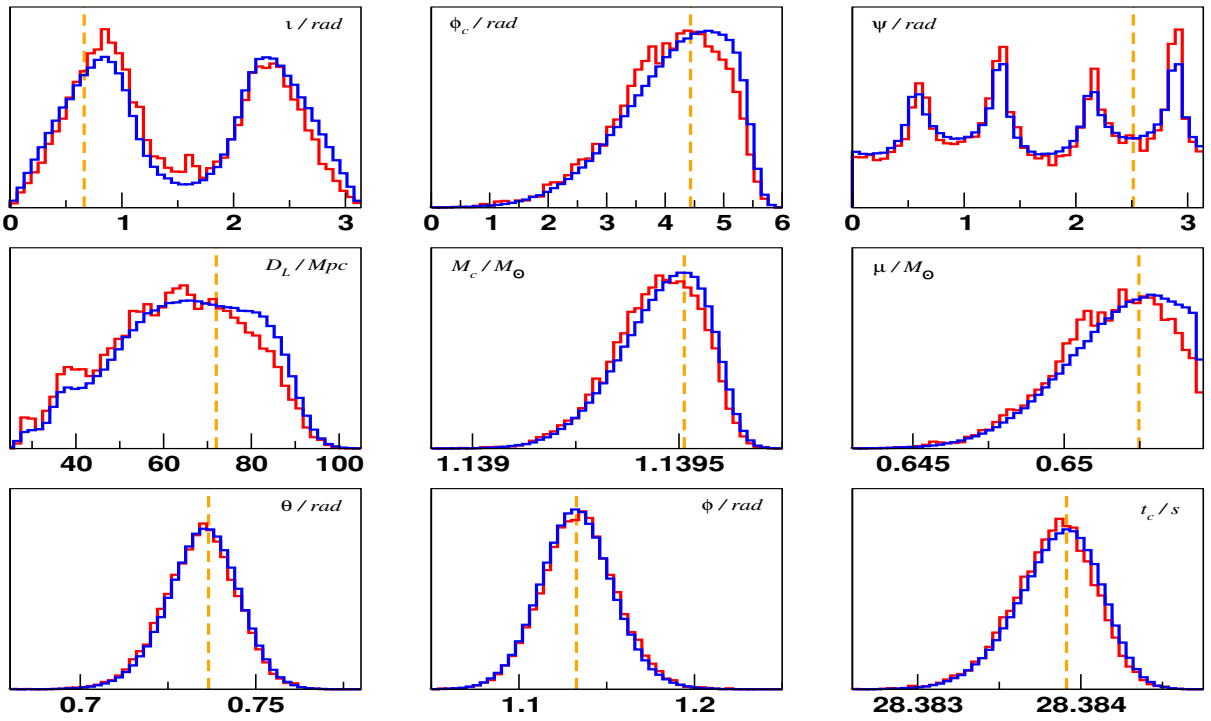

Figure 9.11: Marginalised posterior distribution of the nine parameters for BNS5 using a 10^6 iteration DEMC (red) and HMC (blue).

Chapter 10

Detection of monochromatic compact galactic binaries with eLISA using a hybrid swarm based algorithm

10.1 Introduction

Compact binary systems, composed of white dwarves, neutron stars and stellar mass black holes, are thought to be a major source of gravitational waves in our galaxy. Estimations from population synthesis models predict a possible number of close to 70 million galactic binaries in the data at any one time[122, 123].

However, identifying the parameters of all these sources is a real challenge in terms of data analysis. The first difficulty is the weak interaction between GWs and the detector, leading to very small wave amplitudes. As a consequence, all the signals will be hidden in the noise of the instrument and only a frequency analysis reveals the features of the signal in the noise background. Another issue for analysis is closely connected to the nature of gravitational wave and is often called the *Cocktail party problem* [124]. Unlike EM observations, a GW detector measures the signals in all directions simultaneously. The advantage of this simultaneous detection is to be able to constantly observe and gather information on the sources. The drawback is the confusion between signals: two binaries with extremely close frequencies cannot be analysed separately if the difference between their frequencies is smaller than the smallest accessible frequency bin $\Delta f = 1/T_{obs}$, where T_{obs} is the length of the observation period. Thus we know that in certain frequency bins, there will be more than one binary and it will be impossible to separate the two signals: only the one with the highest SNR can be recovered. While source confusion is a problem for the eLISA mission, we do not expect to have the same confusion noise from a galactic binary foreground as with the LISA configuration [125]. In addition, the power in one binary is spread over several frequency bins owing to the Doppler modulation frequency coming from the orbit of the detector around the Sun [126]. This modulation frequency is denoted as $f_m = 1/yr$.

In regards to these challenges, it has been necessary to develop algorithms dedicated specifically to parameter identification for GWs. There have been two major types of algorithms tested in the framework of space based GW data analysis: grid based and stochastic searches. Grid based algorithms use a uniform distribution of templates over the parameter space, designed to achieve a given degree of minimal match with the sources. They can either be in the form of a hierarchichal search, where regions of interest are identified separately with refinment of the grid [127], or in the form of a N sources search where metatemplates of multiple binaries are computed [128]. However, given the high computational cost of computing a uniform grid of templates in a high dimension parameter space [129], stochastic approach has been favoured in the field. A variety of such algorithms have been tested including genetic algorithms [130], tomographic reconstruction [131] and Markov chain Monte Carlo based algorithms [132, 133, 134, 135, 136, 137, 138, 139, 140]. Among all these algorithms, the MCMC based ones have been the most successful so far, especially the Block Annealed Metropolis Hastings that proved to be the most suited for detecting a full population of binaries in the Galaxy [141].

Though much progress has been made in the GB source search, it is still necessary to look for even

faster and more efficient algorithms in order to be ready to analyse real data when the mission is launched. In this article we present a first work on a swarm-based search algorithm that combines Particle Swarm Optimization (PSO, [142, 143]), Differential Evolution (DE, [76]) and Markov Chain Monte Carlo routines (MCMC, [66, 67]).

This chapter is structured as follows. In Section 10.2, we give the expressions of the waveform as seen by an eLISA-like detector in the time and Fourier domain for a monochromatic ultra compact galactic binary. In addition to that, we present some of the other features associated to this object such as the F-statistic and the multi-modal nature of the likelihood. In Section 10.3, we describe the general form of each core algorithm taking part in the method we developed for galactic binaries search. In Section 10.4, we present our strategy to design our search algorithm and how we benchmarked it on a single source search. In Section 10.5, we present the performances of our search algorithm for the analysis of two multiple sources data sets; the first one comprising 18 binaries without confusion and the second one with 30 binaries and mild confusion.

10.2 Definition of gravitational waveform for monochromatic ultra compact galactic binaries

The main difficulty in GW astronomy comes from the fact that the output of the detector is a superposition of millions of GW signals, plus a number of noise contributions. A standard technique for extracting signals from noisy data is matched filtering. The idea behind this method is as follows : given a theoretical waveform model, or template, parametrized by astrophysically motivated parameters $\{\lambda^\mu\}$, what is the parameter set that maximizes the correlation between the template and a possible individual GW signal in the data. This method is particularly suited in the framework of galactic binaries detectable by eLISA given that their signal is present throughout the whole mission.

Once a signal is detected, our next goal is to estimate the parameters for the best fit template. A number of recent studies have demonstrated that the Fisher information matrix is an unreliable tool in GW astronomy due to the fact that in a number of cases, the posterior distributions in the parameter errors are non-Gaussian (a pre-requisite for using the Fisher matrix in the first place) [64]. With this in mind, our goal is to carry out a Bayesian analysis when conducting the parameter estimation study.

10.2.1 Time domain response

The strain of the GW, $s(t)$, as seen by the detector can be written, in the low frequency approximation (LFA)[108], as a linear combination of the two GW wave polarizations $h_{+,\times}(t)$ and the detector beam pattern functions $F^{+,\times}(t)$,

$$s(t) = h_+(t)F^+(t) + h_\times(t)F^\times(t). \tag{10.1}$$

The LFA is valid when the GW wavelength is greater than the size of the detector, or conversely, where the frequency of the wave is inferior to a cutoff frequency $f^* = c/(2\pi L)$[144, 145]. For a monochromatic binary, the frequency of the wave, f_0, is related to the orbital frequency f_{orb} of the binary by

$$f_0 = 2f_{orb}. \tag{10.2}$$

For ultra compact galactic binaries, the range of GW frequencies lies between 10^{-4} and 10^{-2} Hz. Given that the current eLISA armlength is $L = 10^6$ km, the corresponding cutoff frequency is $f^* \approx 10^{-2}$ Hz, hence the validity of the LFA approximation in our study.

In the case of circular monochromatic GBs, and in the framework of general relativity, the two polarizations of the GW in Eq. (10.1) are given by

$$h_+(t) = A(1+\cos^2(\iota))\cos(\Phi(t)+\varphi_0), \tag{10.3}$$
$$h_\times(t) = -2A\cos(\iota)\sin(\Phi(t)+\varphi_0), \tag{10.4}$$

where A is the amplitude of the wave, ι is the inclination of the orbital plane, φ_0 is the initial phase and $\Phi(t)$ is the phase of the wave. The amplitude of the wave is dependent on the astrophysical source emitting the GW,

$$A = 2\left(\pi f_0\right)^{2/3}\frac{\mathcal{M}_{ch}^{5/3}}{D}, \tag{10.5}$$

Parameters	$\ln A$	i (rad)	$\ln f_0$ (mHz)	θ (rad)	ϕ (rad)	φ_0 (rad)	ψ (rad)
Range	$[-26, -21]$	$[0 - \pi]$	$[-4 - -2]$	$[0 - \pi]$	$[0 - 2\pi]$	$[0 - 2\pi]$	$[0 - \pi]$

Table 10.1: Typical range of the parameters for a compact galactic binary

where D is the distance of the source and $\mathcal{M}_{ch}$ is the chirp mass. The phase $\Phi(t)$ of a monochromatic galactic binary is

$$\Phi(t) = 2\pi f_0 \left[t + R_\oplus \sin(\theta)\cos(2\pi f_m t - \phi)\right], \tag{10.6}$$

where θ is the co-latitude, ϕ is the longitude and $R_\oplus = 1AU$ is the radius of eLISA orbit. The phase differs from a simple monochromatic process due to the motion of the detector with respect to the source inducing a Doppler motion contribution to the phase.

The beam pattern functions of the detector, $F^{+,\times}(t)$, are described in the LFA by

$$F^+(t;\psi,\theta,\phi) = \frac{1}{2}\left[\cos(2\psi)D^+(t;\psi,\theta,\phi,\lambda) - \sin(2\psi)D^\times(t;\psi,\theta,\phi,\lambda)\right], \tag{10.7}$$

$$F^\times(t;\psi,\theta,\phi) = \frac{1}{2}\left[\sin(2\psi)D^+(t;\psi,\theta,\phi,\lambda) + \cos(2\psi)D^\times(t;\psi,\theta,\phi,\lambda)\right], \tag{10.8}$$

where ψ is the polarisation angle of the wave and the full expressions for the coefficients $D^{+,\times}(t)$ are given by [144]:

$$\begin{aligned} D^+(t;\psi,\theta,\phi) = \frac{\sqrt{3}}{64}\big[&-36\sin^2(\theta)\sin(2\alpha(t)-2\lambda) + (3+\cos(2\theta)) \\ &(\cos(2\phi)\{9\sin(2\lambda) - \sin(4\alpha(t)-2\lambda)\} + \sin(2\phi)\{\cos(4\alpha-2\lambda) - 9\cos(2\lambda)\}) \\ &-4\sqrt{3}\sin(2\theta)\left(\sin(3\alpha(t)-2\lambda\ \phi) - 3\sin(\alpha(t)-2\lambda+\phi)\right)\big], \end{aligned} \tag{10.9}$$

$$\begin{aligned} D^\times(t;\psi,\theta,\phi) = \frac{1}{16}\big[&\sqrt{3}\cos(\theta)\left(9\cos(2\lambda-2\phi) - \cos(4\alpha(t)-2\lambda-2\phi)\right) \\ &-6\sin(\theta)\left(\cos(3\alpha(t)-2\lambda-\phi) + 3\cos(\alpha(t)-2\lambda+\phi)\right)\big]. \end{aligned} \tag{10.10}$$

where $\alpha(t) = 2\pi f_m t$ is the orbital phase of the center of mass of the eLISA constellation and λ defines the configuration of the arms ($\lambda = 0$ or $3\pi/2$ for a two-arm configuration).

As a consequence, for a circular monochromatic binary, we characterize the GW response by the set of seven parameters λ^μ:

$$\lambda^\mu = \{\ln A, \cos i, \varphi_0, \psi, \ln f_0, \cos\theta, \phi\}. \tag{10.11}$$

In Table 10.1, we give the range expected for a compact galactic binary according to the last stellar population models of the Galaxy. However, only a handful of binaries among the stellar population will be resolvable by the detector during the mission.

The signal given in Eq. (10.1) is evaluated in the time domain between 0 and a fixed observation time T_{obs}. The current design of eLISA mission has various scenarios with observation times being either equal to 1, 2 or 5 years. Once the signal is generated over the appropriate time length, we then use Fast Fourier Transform (FFT) algorithm to get the response in the Fourier domain. Thus this approach requires both to generate long vectors in the time domain and do a FFT, making the process quite heavy in terms of computation.

10.2.2 Fourier domain response

Given the relatively simple expression of the waveform in the time domain, one can derive the expressions directly in the Fourier domain. The underlying motivation is to speed up the algorithm, without having to first generate a full length waveform in the time domain and then Fourier transform it numerically using FFT algorithm. Moreover, most of the signal power is contained in a small number of frequency bins around the carrier frequency f_0, whose number is constrained by the power spread coming from Doppler modulation.

Thus, our goal is to find the expressions of the Fourier coefficients $\tilde{s}_n$ associated to the Fourier series of the signal $s(t)$ for a given observation time T_{obs} as

$$s(t) = \sum_n \tilde{s}_n e^{2\pi i n \frac{t}{T_{obs}}}. \tag{10.12}$$

The first step is to rewrite the expression of the signal measured by eLISA as

$$s(t) = A_+ F^+ \cos(\Phi(t)) + A_\times F^\times \sin(\Phi(t)), \tag{10.13}$$

where $A_+ = A(1+\cos^2(\iota))$, $A_\times = -2A\cos(\iota)$ and where the phase $\Phi(t)$ can be written as the sum of three contributions

$$\Phi(t) = 2\pi f_0 t + \Phi_D(t) + \varphi_0. \tag{10.14}$$

The first term is related to the binary frequency, the second one is the Doppler modulation and the last is the initial phase of the signal. If we put everything together we get

$$s(t) = \mathbb{R}\mathrm{e}\left[A_+ F^+(t) e^{2\pi i f_0 t} e^{i\Phi_D(t)} e^{i\varphi_0}\right] + \mathbb{I}\mathrm{m}\left[A_\times F^\times(t) e^{2\pi i f_0 t} e^{i\Phi_D(t)} e^{i\varphi_0}\right]. \tag{10.15}$$

If we manage to express each time function in the form of a Fourier series, we can express the Fourier coefficients $\tilde{s}_n$ as a discrete convolution of each Fourier series. In the following subsections, we detail the computation of each Fourier series.

10.2.2.1 Frequency term

We want to express the exponential of the frequency f_0 as a Fourier series,

$$e^{2\pi i f_0 t} = \sum_{n=-\infty}^{\infty} \tilde{a}_n e^{2\pi i n \frac{t}{T_{obs}}}, \tag{10.16}$$

where $\tilde{a}_n$ is the coefficient associated to the n^{th} frequency bin. The coefficient $\tilde{a}_n$ can be directly computed by

$$\begin{aligned}
\tilde{a}_n &= \frac{1}{T_{obs}} \int_0^{T_{obs}} e^{2\pi i f_0 t} e^{-2\pi i n \frac{t}{T_{obs}}} \mathrm{d}t = \frac{1}{T_{obs}} \int_0^{T_{obs}} e^{2\pi i t (f_0 - \frac{n}{T_{obs}})} \mathrm{d}t && (10.17)\\
&= \frac{1}{2\pi i (f_0 T_{obs} - n)} (e^{2\pi i t (f_0 - \frac{n}{T_{obs}})} - 1) && (10.18)\\
&= \frac{1}{2\pi i (f_0 T_{obs} - n)} e^{i\pi(f_0 T_{obs} - n)} (e^{i\pi(f_0 T_{obs} - n)} - e^{-i\pi(f_0 T_{obs} - n)}) && (10.19)\\
&= \frac{1}{2\pi i (f_0 T_{obs} - n)} e^{i\pi(f_0 T_{obs} - n)} (2i \sin(\pi(f_0 T_{obs} - n))), && (10.20)
\end{aligned}$$

which gives us

$$\tilde{a}_n = \mathrm{sinc}(\pi(f_0 T_{obs} - n)) e^{i\pi(f_0 T_{obs} - n)} \tag{10.21}$$

If we think in terms of signal analysis, given that the Fourier transform of a rectangle function is a cardinal sine, the previous result corresponds to the Fourier transform of a monochromatic function of frequency f_0 multiplied by a step function taking values between 0 and T_{obs}. Thus, owing to the fact that the signal is emitted over a limited period of time, the power of the monochromatic signal is spread across several bins of frequency around the carrier frequency f_0. Since our goal is to select and identify only the frequency bins containing significant amount of signal power, we need to estimate the individual contribution for each bin thanks to

$$|\tilde{a}_n^2| = \mathrm{sinc}^2(\pi(f_0 T_{obs} - n)). \tag{10.22}$$

On the left panel of Figure 10.1, we plot the function $\mathrm{sinc}^2(x)$ with respect to x/π. This function is highly peaked around 0 with sidebands roughly separated by π. In terms of the coefficients $\tilde{a}_n$, this means that only the terms for which the argument of the cardinal sine in Eq. (10.21) is close to 0 will contribute to the total power of the signal. However, if we want to set a proper threshold on the number of bins required, we have to compute the variation of the integrated power depending on the number of terms we take.

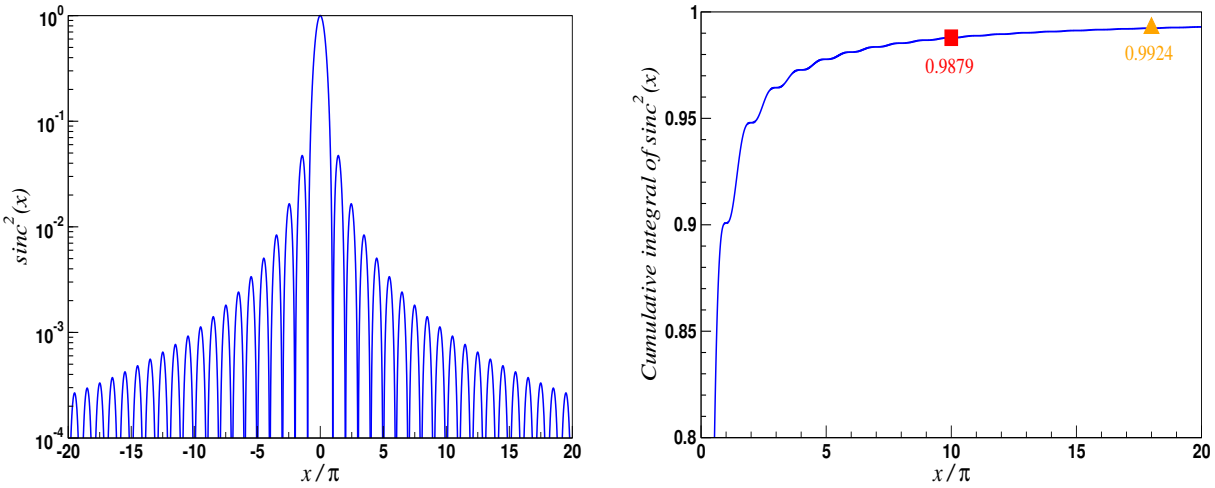

Figure 10.1: The left panel displays the squared of cardinal sine function, $\mathrm{sinc}^2(x)$, with respect to x/π. The right panel shows the value of the cumulative integral of $\mathrm{sinc}^2(y)$ between 0 and x with respect to x/π. The two symbols indicate the values of interest: the red square corresponds to $x = 10\pi$ and the green triangle corresponds to $x = 18\pi$

On the right panel of Figure 10.1, we plot the cumulated power of sinc^2 between 0 and x, $\int_0^x \mathrm{sinc}^2(y)\mathrm{d}y$, as a function of x/π. Since the total power is given by $\int_0^\infty \mathrm{sinc}^2(y)\mathrm{d}y = \pi/2$, we have renormalized the y-axis by $\pi/2$ so that it represents the percentage of power contained between 0 and x compared to the total power. In the figure, we see that 90% of the power is contained between 0 and 2π. However, precisions as high as $98 - 98\%$ are demanded in order to have good matching results between time and Fourier domain responses. In regards of the function plotted in the right panel of Figure 10.1, we have selected two different limits:

- $x = 10\pi$ with a normalized cumulative power of 0.9879
- $x = 18\pi$ with a normalized cumulative power of 0.9924

Following previous works on this matter [126], we will call these two limits the 98 and 99 percent case respectively. At the end of this section, we use these two limits to build the Fourier response of the signal $s(t)$ and compare their impact both on signal matching and computation time.

10.2.2.2 Doppler term

The Doppler modulation phase can be written as

$$\Phi_D(t) = \beta \cos(2\pi f_m t + \delta), \tag{10.23}$$

where $\beta = 2\pi f_0 R_\oplus \sin(\theta)$ and $\delta = -\phi$. Our goal is to express the exponential of the Doppler phase as a Fourier series,

$$e^{i\Phi_D(t)} = e^{i\beta\cos(2\pi f_m t+\delta)} = \sum_{n=-\infty}^{\infty} \tilde{b}_n e^{2\pi i n \frac{t}{T_{obs}}}, \tag{10.24}$$

where the coefficients $\tilde{b}_n$ are given by,

$$\tilde{b}_n = \frac{1}{T_{obs}} \int_0^{T_{obs}} e^{i\Phi_D(t)} e^{-2\pi i n \frac{t}{T_{obs}}} \mathrm{d}t. \tag{10.25}$$

In this form, the integral is not trivial owing to the cosine term in $\Phi_D(t)$. One way to solve the integration, is to express $e^{i\Phi_D(t)}$ as a harmonic series of the modulation frequency f_m,

$$e^{i\Phi_D(t)} = e^{i\beta\cos(2\pi f_m t+\delta)} = \sum_{k=-\infty}^{\infty} b_k e^{2\pi i k f_m t}, \tag{10.26}$$

using Jacobi-Anger expansion

$$e^{iz\cos(\theta)} = \sum_{n=-\infty}^{\infty} J_n(z)e^{in\theta}, \tag{10.27}$$

where z and θ are real and $J_n(z)$ is a Bessel function of the first kind of order n. Given Eq. (10.27), we have

$$\begin{aligned} e^{i\Phi_D(t)} &= e^{i\beta\sin(2\pi f_m t+\delta+\frac{\pi}{2})}, & (10.28)\\ &= \sum_{k=-\infty}^{+\infty} J_k(\beta)e^{ik(2\pi f_m t+\delta+\frac{\pi}{2})}, & (10.29)\\ &= \sum_{k=-\infty}^{+\infty} J_k(\beta)e^{ik(\delta+\frac{\pi}{2})}e^{2\pi ikf_m t}, & (10.30)\\ &= \sum_{k=-\infty}^{+\infty} b_k e^{2\pi ikf_m t}, & (10.31) \end{aligned}$$

where

$$b_k = J_k(2\pi f_0 R_\oplus \sin(\theta))e^{ik(\frac{\pi}{2}-\phi)}. \tag{10.32}$$

In this form, the exponential of the Doppler modulations is now expressed as a series of exponentials of the modulation frequencies which makes the integral in Eq. (10.25) much easier to compute.

$$\begin{aligned} \tilde{b}_n &= \frac{1}{T_{obs}}\int_0^{T_{obs}} \left(\sum_{k=-\infty}^{\infty} b_k e^{2\pi ikf_m t}\right) e^{-2\pi in\frac{t}{T_{obs}}}\mathrm{d}t & (10.33)\\ &= \sum_{k=-\infty}^{\infty}\frac{1}{T_{obs}}\int_0^{T_{obs}} b_k e^{2\pi it(kf_m-\frac{n}{T_{obs}})}\mathrm{d}t & (10.34)\\ \tilde{b}_n &= \sum_{k=-\infty}^{\infty} b_k \,\mathrm{sinc}(\pi(kf_m T_{obs}-n))e^{i\pi(kf_m T_{obs}-n)} & (10.35) \end{aligned}$$

This result is very similar with the one we had before. The main difference comes from the infinite series of Bessel function. However, this series can be reduced to a finite number of terms making it computationally affordable. We define the bandwidth of a signal to be the interval of frequencies containing most of the power of the signal. For a monochromatic binary, the empirical bandwidth associated to the Doppler modulation is given by [126]

$$B = (1+\beta)f_m. \tag{10.36}$$

As a consequence, only the frequency bins in the interval of frequency $[-B, B]$ need to be evaluated in the sum in Eq. (10.35). Even though this empirical value is taken in most of the works, we decided to increase it,

$$B_{ext} = (10+\beta)f_m, \tag{10.37}$$

so that we are sure to have all the power from Doppler modulation. This extension comes with an increased computation time, but as we will see, the gain in efficiency using Fourier domain response is so large that we decided to choose increased accuracy in signal reconstruction.

10.2.2.3 Detector term

As for Doppler modulation, it is possible to decompose the detector function into harmonics functions of the modulation frequency f_m as

$$F^{+,\times}(t) = \sum_{n=-4}^{4} p_n^{+,\times} e^{2\pi i f_m nt}, \tag{10.38}$$

using the fact that the detector function can be written as

$$D^{+,\times}(t) = \sum_{n=-4}^{4} d_n^{+,\times} e^{2\pi i f_m nt}. \tag{10.39}$$

n	a_n^+	b_n^+	$a_n^\times$	$b_n^\times$
0	$\frac{-9\sqrt{3}}{64}(3+\cos(2\theta))\sin(2\phi)$	0	$\frac{9\sqrt{3}}{16}\cos(\theta)\cos(2\phi)$	0
1	$\frac{9}{16}\sin(2\theta)\sin(\phi)$	$\frac{9}{16}\sin(2\theta)\cos(\phi)$	$\frac{-9}{8}\sin(\theta)\cos(\phi)$	$\frac{9}{8}\sin(\theta)\sin(\phi)$
2	0	$\frac{-9\sqrt{3}}{16}\sin^2(\theta)$	0	0
3	$\frac{1}{3}a_1^+$	$\frac{-1}{3}b_1^+$	$\frac{1}{3}a_1^+$	$\frac{-1}{3}b_1^+$
4	$\frac{-1}{9}a_0^+$	$\frac{-\sqrt{3}}{64}(3+\cos(2\theta))\cos(2\phi)$	$\frac{-1}{9}a_0^+$	$\frac{-\sqrt{3}}{16}\cos(\theta)\sin(2\phi)$

Table 10.2: Values of the harmonic coefficients for D^+ and $D^\times$. These values can be read directly from the expressions of D^+ and $D^\times$ in Eq. (10.10) and Eq. (10.11).

Note that the sum only needs to be taken between -4 and 4 thanks to the quadrupole approximation in the low frequency regime approximation. Since the detector functions $D^+(t)$ and $D^\times(t)$ are expressed in terms of cosine and sine of the modulation frequency, it is more convenient to work with the real series,

$$D^{+,\times}(t) = \sum_{n=0}^{4} a_n^{+,\times}\cos(2\pi i f_m n t) + b_n^{+,\times}\sin(2\pi i f_m n t). \tag{10.40}$$

Thus, the expressions for the harmonic coefficients $a_n^{+,\times}$ and $b_n^{+,\times}$ can be read directly from Eq. (10.10) and Eq. (10.11), and are reported in Table 10.2. To switch back to the complex coefficients, we use

$$d_n^{+,\times} = \frac{1}{2}(a_n^{+,\times} - i b_n^{+,\times}) \quad \text{for } n \geq 0, \tag{10.41}$$

$$d_{-n}^{+,\times} = \overline{d_n^{+,\times}} \quad \text{for } n < 0, \tag{10.42}$$

where the overbar stands for complex conjugate.

From Eq. (10.8) and Eq. (10.9), we can write

$$p_n^+ = \frac{1}{2}(\cos(2\psi)d_n^+ - \sin(2\psi)d_n^\times) \tag{10.43}$$

$$p_n^\times = \frac{1}{2}(\sin(2\psi)d_n^+ + \cos(2\psi)d_n^\times) \tag{10.44}$$

As before, we still have to compute the Fourier series from the harmonic decomposision,

$$F^{+,\times}(t) = \sum_{n=-\infty}^{+\infty} \tilde{p}_n^{+,\times} e^{2\pi i n \frac{t}{T_{obs}}}, \tag{10.45}$$

that is obtained once more through the use of cardinal sine as

$$\tilde{p}_n^{+,\times} = \sum_{k=-4}^{4} p_k^{+,\times}\mathrm{sinc}(\pi(k f_m T_{obs} - n))e^{i\pi(k f_m T_{obs} - n)}. \tag{10.46}$$

10.2.2.4 Total signal Fourier coefficients

We now have everything we need to compute the Fourier domain response of the signal $s(t)$. If we combine the expressions of the Fourier series we derived in Eq. (10.15), we find that

$$\begin{aligned} s(t) = {} & \mathbb{R}\mathrm{e}\left[A_+ \sum_m \tilde{p}_m^+ e^{2\pi i m \frac{t}{T_{obs}}} \sum_k \tilde{a}_k e^{2\pi i k \frac{t}{T_{obs}}} \sum_l \tilde{b}_l e^{2\pi i l \frac{t}{T_{obs}}} e^{i\varphi_0}\right] + \\ & \mathbb{I}\mathrm{m}\left[A_\times \sum_m \tilde{p}_m^\times e^{2\pi i m \frac{t}{T_{obs}}} \sum_k \tilde{a}_k e^{2\pi i k \frac{t}{T_{obs}}} \sum_l \tilde{b}_l e^{2\pi i l \frac{t}{T_{obs}}} e^{i\varphi_0}\right] \end{aligned} \tag{10.47}$$

$$s(t) = \mathbb{R}\mathrm{e}\left[A_+ e^{i\varphi_0} \sum_{m,k,l} \tilde{p}_m^+ \tilde{a}_k \tilde{b}_l e^{2\pi i(m+k+l)\frac{t}{T_{obs}}}\right] + \mathbb{I}\mathrm{m}\left[A_\times e^{i\varphi_0} \sum_{m,k,l} \tilde{p}_m^\times \tilde{a}_k \tilde{b}_l e^{2\pi i(m+k+l)\frac{t}{T_{obs}}}\right] \tag{10.48}$$

The previous equation suggests to define two complex quantities A_n and B_n as

$$A_n = A_+ e^{i\varphi_0} \sum_{m,k,l} \tilde{p}_m^+ \tilde{a}_k \tilde{b}_l, \tag{10.49}$$

$$B_n = A_\times e^{i\varphi_0} \sum_{m,k,l} \tilde{p}_m^\times \tilde{a}_k \tilde{b}_l, \tag{10.50}$$

where $n = m + k + l$. The signal can then be expressed as

$$s(t) = \mathbb{R}\mathrm{e}\left[\sum_n A_n e^{2\pi i n \frac{t}{T_{obs}}}\right] + \mathbb{I}\mathrm{m}\left[\sum_n B_n e^{2\pi i n \frac{t}{T_{obs}}}\right], \tag{10.51}$$

where the range of the sum for the index n depends on the ranges defined previously for the frequency, Doppler and detector terms. Finally, we can derive the complex coefficients of the Fourier series using Eq. (10.42),

$$\tilde{s}_n = \frac{1}{2} e^{i\varphi_0} \sum_k \tilde{a}_k \sum_l \tilde{b}_l \sum_m \left(A_+ \tilde{p}_m^+ + e^{i3\pi/2} A_\times \tilde{p}_m^\times\right). \tag{10.52}$$

We have now everything to generate the Fourier domain response for a monochromatic galactic binary. To test the consistency between the waveform generated directly in the Fourier domain, $\tilde{h}_F$, and the waveform generated in the time domain, $\tilde{h}_T$, one can use the overlap $\mathcal{O}_{F/T}$ to evaluate the match between these two signals,

$$\mathcal{O}_{F/T} = \frac{\langle h_F \,|h_T\rangle}{\sqrt{\langle h_F \,|h_F\rangle \langle h_T \,|h_T\rangle}}. \tag{10.53}$$

We generated one set of 10^4 binaries whose parameters were drawn uniformly accordingly to the ranges given in Table 10.1. For this set, we made several runs with various configurations where we varied both the observation time (1, 2 and 5 years) and the number of selected terms in the argument of the cardinal sines in Eq. (10.21), Eq. (10.35) and Eq. (10.46) (10 for 98% of the power and 18 for 99%).

For each of these runs, we recorded the overlaps $\mathcal{O}_{F/T}$ and the ratio of the computation time for waveform time response generation with FFT over the computation time for Fourier response generation. Thus, from the 10^4 sources overlaps and computation time ratio, we were able to derive the associated distributions for each configuration. In Figure 10.2, we plot the histograms we obtained for the 98 and 99% cases assuming 1 year (top), 2 years (middle) and 5 years of data (bottom).

The plots of the overlaps $\mathcal{O}_{F/T}$ reveal that all the overlaps were found above 0.98 regardless of the observation time. As expected, in the case of 99% power, the distributions are better: all the overlaps are superior to 0.99 and the distributions are both closer and denser around 1. As a direct consequence of this, we observe an increased computation time in the 99% case compared to the 98%. However, in both cases, the gain in computation time is always superior to 5 and can go as high as 150 for 99% and 200 for 98%. Furthermore, we observe a specific structure on the gain distribution with various peaks for given gain. A closer inspection on the results revealed that these peaks are correlated with the frequency of the binary: the gain is lower for small frequencies and higher for high frequencies. This effect can be well understood by considering the associated sampling frequency: the higher the frequency of the binary, the higher the sampling frequency and the longer the waveform vector. In this case, the time to generate such a long time vector and do a FFT is even longer hence favoring a Fourier approach where only the important terms are computed.

In light of these results, we decided to go for a higher accuracy at an acceptable price on computation time by selecting the case where we take 99% of the power with 18 terms in the argument of the cardinal sine. As a summary, we give the final ranges for all the sums in Eq. (10.52) in Table 10.3.

10.2.3 F-statistic

The F-Statistic is a method for analytically maximizing over certain parameters in the GW response. The set of waveform parameters can be split into a first set depending specifically on the source that we call intrinsic, i.e. $\{f_0, \theta, \phi\}$, and another set of extrinsic parameters, i.e. $\{\iota, \phi_0, A, \psi\}$. In the LFA, we can re-write the detector response in Eq. (10.1) in such a way that it is expressed as a sum of constant amplitudes a_i, depending only on extrinsic parameters, and time varying functions $A^i(t)$, depending only on intrinsic parameters, i.e.

$$h(t) = \sum_{i=1}^{4} a_i A^i, \tag{10.54}$$

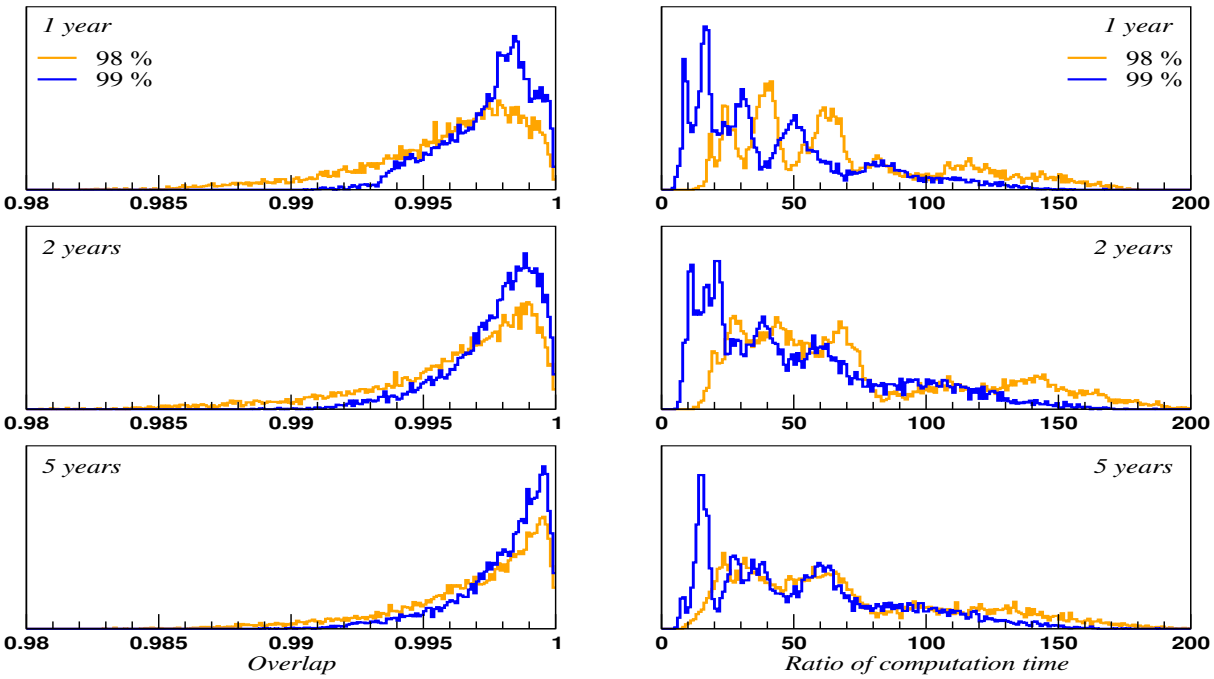

Figure 10.2: Distribution of overlap $\mathcal{O}_{F/T}$ (left) and of computation time for waveform time response generation with FFT over the computation time for Fourier response generation (right) for a set of 10^4 binaries with parameters drawn uniformly in the ranges given in Table 10.1. The observation time was fixed at 1 year (up), 2 years (middle) and 5 years (bottom). For each configuration, we tested the Fourier waveform generation with the two assumptions on the number of terms selected corresponding to 98 and 99% of the original signal power.

where we define the quantities a_i and A_i by

$$a_1 = \frac{A}{2}((1+\cos^2 i)\cos\varphi_0\cos 2\psi - 2\cos i\sin\varphi_0\sin 2\psi), \tag{10.55}$$

$$a_2 = -\frac{A}{2}(2\cos i\sin\varphi_0\cos 2\psi + (1+\cos^2 i)\cos\varphi_0\sin 2\psi), \tag{10.56}$$

$$a_3 = -\frac{A}{2}(2\cos i\cos\varphi_0\sin 2\psi + (1+\cos^2 i)\sin\varphi_0\cos 2\psi), \tag{10.57}$$

$$a_4 = \frac{A}{2}((1+\cos^2 i)\sin\varphi_0\sin 2\psi - 2\cos i\cos\varphi_0\cos 2\psi), \tag{10.58}$$

and

$$A^1 = D^+(t,\theta,\phi)\cos\Phi(t,f_0,\theta,\phi), \tag{10.59}$$

$$A^2 = D^\times(t,\theta,\phi)\cos\Phi(t,f_0,\theta,\phi), \tag{10.60}$$

$$A^3 = D^+(t,\theta,\phi)\sin\Phi(t,f_0,\theta,\phi), \tag{10.61}$$

$$A^4 = D^\times(t,\theta,\phi)\sin\Phi(t,f_0,\theta,\phi). \tag{10.62}$$

The idea is now to find the maximum of the likelihood with respect to the coefficients a_i. The reduced likelihood is given by

$$\ln\mathcal{L}(\lambda^\mu) = \langle s|h(\lambda^\mu)\rangle - \frac{1}{2}\langle h(\lambda^\mu)|h(\lambda^\mu)\rangle, \tag{10.63}$$

$$= \left\langle s|\sum_{i=1}^{4} a_i A^i\right\rangle - \frac{1}{2}\left\langle\sum_{i=1}^{4} a_i A^i|\sum_{j=1}^{4} a_j A^j\right\rangle, \tag{10.64}$$

$$= \sum_{i=1}^{4} a_i N^i - \frac{1}{2}\sum_{i=1}^{4}\sum_{j=1}^{4} a_i a_j M_{ij}, \tag{10.65}$$

Fourier coefficients	Ranges of the sum
$\tilde{a}_k$	$-18 \leq k - \left\lfloor f_0 T_{obs} \right\rfloor \leq 18$
$\tilde{b}_l$	$-(1+\beta) f_m T_{obs} - 18 \leq l \leq (1+\beta) f_m T_{obs} + 18$
$\tilde{p}_m^{+,\times}$	$-4 f_m T_{obs} - 18 \leq m \leq 14 f_m T_{obs} + 18$
$\tilde{s}_n$	$-(1+\beta) f_m T_{obs} - 4 f_m T_{obs} - 54 \leq n - \left\lfloor f_0 T_{obs} \right\rfloor \leq (1+\beta) f_m T_{obs} 4 f_m T_{obs} + 54$

Table 10.3: Ranges of the sum selected for the computation of the Fourier terms in Eq. (10.52).

where we define the vector $N^i = \langle s|A^i \rangle$ and the matrix $M_{ij} = \langle A^i|A^j \rangle$. If we now set

$$\frac{\partial \ln \mathcal{L}(\lambda^\mu)}{\partial a_k} = 0, \tag{10.66}$$

we obtain

$$N^k - \sum_{i=1}^{4} a_i M_{ik} = 0, \tag{10.67}$$

which allows us to define

$$a_j = M_{jk}^{-1} N^k. \tag{10.68}$$

If we now substitute these terms into the expression for the reduced log likelihood in Eq. (10.64), we get the F-statistic

$$\mathcal{F} = \log \mathcal{L} = \frac{1}{2} M_{ij}^{-1} N^i N^j, \tag{10.69}$$

which automatically maximizes over the extrinsic parameters.

Now, given the numerical values of the amplitudes a_i, we can solve for the extrinsic parameters according to

$$A = \frac{A_+ + \sqrt{A_+^2 - A_\times^2}}{2}, \tag{10.70}$$

$$\psi = \frac{1}{2} \arctan\left(\frac{A_+ a_4 - A_\times a_1}{-(A_\times a_2 + A_+ a_3)} \right), \tag{10.71}$$

$$\iota = \arccos\left(\frac{-A_\times}{A_+ + \sqrt{A_+^2 - A_\times^2}} \right), \tag{10.72}$$

$$\varphi_0 = \arctan\left(\frac{c(A_+ a_4 - A_\times a_1)}{-c(A_+ a_2 + A_\times a_3)} \right), \tag{10.73}$$

where

$$\begin{aligned} A_+ &= \sqrt{(a_1+a_4)^2 + (a_2-a_3)^2} + \sqrt{(a_1-a_4)^2 + (a_2+a_3)^2}, \\ A_\times &= \sqrt{(a_1+a_4)^2 + (a_2-a_3)^2} - \sqrt{(a_1-a_4)^2 + (a_2+a_3)^2}, \\ c &= \frac{\sin(2\psi)}{|\sin(2\psi)|}. \end{aligned} \tag{10.74}$$

As with the waveform generation, it is possible to speed up the computation time required to get the F-statistic by computing the A^i functions directly in the Fourier domain. The derivation of the expression for the Fourier coefficients of these functions is very similar to the one we made before in Eq. 10.2.2. Let us define the two following complex quantities

$$\tilde{C}_n = \frac{1}{2} \sum_k \tilde{a}_k \sum_l \tilde{b}_l \sum_m \tilde{d}_m^+, \tag{10.75}$$

$$\tilde{D}_n = \frac{1}{2} \sum_k \tilde{a}_k \sum_l \tilde{b}_l \sum_m \tilde{d}_m^\times, \tag{10.76}$$

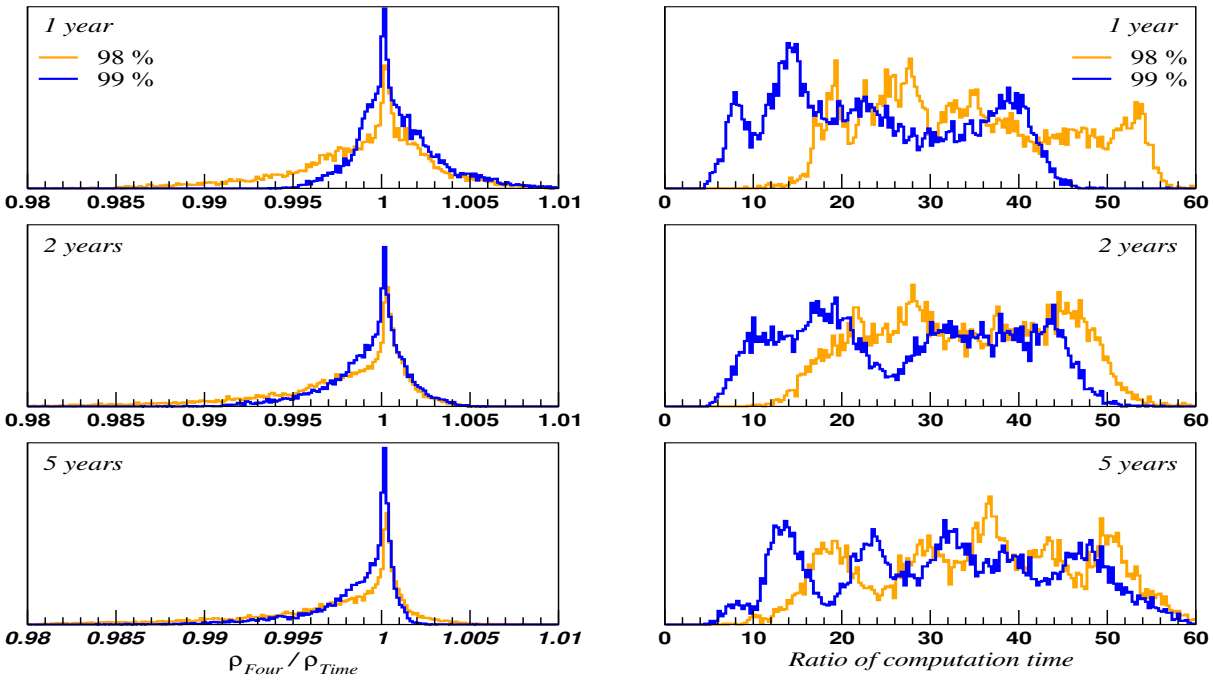

Figure 10.3: Distribution of signal to noise ratio ρ (left) and computation time (right) for F-statistic with generation in the time domain with FFT over generation in the Fourier domain for a set of 10^4 binaries with parameters drawn uniformly in the ranges given in Table 10.1. The observation time was fixed at 1 year (up), 2 years (middle) and 5 years (bottom). For each configuration, we tested the Fourier waveform generation with the two assumptions on the number of terms selected corresponding to 98 and 99% of the original signal power.

where $n = k + l + m$. Then the Fourier coefficients of the A^i functions are expressed in terms of the real and imaginary part of these complex numbers as

$$\tilde{A}^1_n = \tilde{C}^R_n - i\tilde{C}^I_n, \tag{10.77}$$
$$\tilde{A}^2_n = \tilde{D}^R_n - i\tilde{D}^I_n, \tag{10.78}$$
$$\tilde{A}^3_n = \tilde{C}^I_n + i\tilde{C}^R_n, \tag{10.79}$$
$$\tilde{A}^4_n = \tilde{D}^I_n + i\tilde{D}^R_n. \tag{10.80}$$

Once again, we compared the results of the F-statistic computed in the time and in the Fourier domain. We used the same set of binaries and the same simulation framework than described above in Section 10.2.2.4, and we recorded the values of the SNR and computation time. In Figure 10.3, we have plotted the ratios of the SNR obtained directly in Fourier over the one in the time domain (left panels) along with the subsequent computation time ratios (right panels). Regardless of the observation time, the ratios of SNR are always set between 0.98 and 1.01. As expected the distributions are more condensed around 1 for the 99% case. In terms of computation time, we still observe a large increase, although less pronounced than for waveform generation as shown in 10.2, that is always superior to 5 and that can be as high as 60 for 98% and 50 for 99%. As for waveform generation, we have observed that this increase is dependent on the binary frequency with a larger increase for high frequencies, hence the peaked structures in the plots.

We decided to use the 99% option with $n = 18$ terms for increased accuracy. The corresponding ranges of the sum for the terms $\tilde{C}_n$ and $\tilde{D}_n$ are the same than the ones given in Table 10.3.

10.2.4 Multimodal likelihood analysis

One monochromatic galactic binary is parametrized by a set of seven parameters. Since our goal is to use Bayesian inference, necessitating tools such as likelihood or SNR, this implies that we need to work with a 7-dimensional surface that depends both on the signal and the detector, here eLISA. In this section, we give insights on some interesting features of this surface, namely its symmetries and its multi-modal

nature, that explains why the analysis of such an object is difficult and requires advanced computing techniques. As a guideline for this presentation, we choose a fictitious binary with $\iota = 1.74$, $\varphi_0 = 1.664$, $A = 1.2275 \times 10^{-20}$, $f_0 = 0.52234$ mHz, $\theta = 1.273$, $\phi = 1.8845$ and where all the angles are stated in radians. The computed signal to noise ratio for this source is $\rho = 53.92$.

As we saw before, the frequency of the signal as measured by eLISA is modulated by the frequency f_m owing to the motion of the detector in a year. This means that the purely monochromatic emitted signal is now modulated and spread over several frequency bins around the carrier frequency f_0. More importantly, in terms of signal detection, this implies that a shift in frequency could wrongly mimic a signal with lower signal to noise ratio.

This effect is illustrated on the two upper panels of Figure 10.4 for the binary selected. The first upper panel is a two dimensional slice of the likelihood where all the parameters are equal to their true value except for frequency and colatitude that vary over a selected range around their true values. We clearly see two strong modes appearing for frequencies shifted roughly by $\pm 2 f_m$. The values of the signal to noise ratio at these secondary modes is lower than the one at the main mode but is still extremely high, indicating that the shifted template can mimic the true signal at a high degree of accuracy. Moreover, the associated value of colatitude for these modes is completely uncorrelated with the true value and can take any values. We observe the same characteristic on the second upper panel where this time we kept the value of colatitude at its true value while varying the value of longitude instead. The effect is even more dramatic in this case since we observe six secondary modes corresponding to various shifts in frequencies along with their corresponding values for longitude. From an algorithmic point of view, all these Doppler secondary modes can be extremely problematic if the algorithm is not able to escape one of these. Furthermore, we also see that we first need to lock on the main frequency mode before even trying to find values for colatitude and longitude, since the maximum SNR at a secondary gives back wrong values for these parameters.

Another interesting feature of the likelihood surface is illustrated on the third panel row of Figure 10.4. On this plot, we fixed all the parameters at their true values, except for longitude and colatitude. The two dimensional slice of the SNR surface reveals that once more, we have two distinct modes located at the opposite side of a sphere S^2 and related by

$$\begin{aligned} \theta &\rightarrow \pi - \theta, \\ \phi &\rightarrow \phi \pm \pi. \end{aligned}$$

The origin of this second antipodal mode comes from the shape of the detector responses $F^{+,\times}(t, \theta, \phi)$ that give nearly equal response in opposite directions. Once again, the SNR of the main mode is higher than the one of the antipodal. However, even if a search algorithm manages to find the main frequency mode, it can easily be stuck on the antipodal sky angles mode.

If we combine the two types of secondary modes that we described before, we see that there is a 'forest' of local maxima around the SNR/likelihood surface hence requiring careful algorithmic treatment in order to properly identify the main mode corresponding to the true solution.

Finally, there is another class of solutions called symmetric solutions corresponding to parameter mapping leaving the waveform truly invariant. In the response given in Eq. (10.1), the mapping

$$\begin{aligned} \psi &\rightarrow \pi - \psi \\ \varphi_0 &\rightarrow \varphi_0 \pm \pi, \end{aligned}$$

gives back the same value for the waveform. This mode is represented on the bottom panels of Figure 10.4 where we plotted the SNR surface in the case where all the parameters are equal to their true value except the initial phase φ_0 and polarisation ψ. In addition to this symmetric peak, one can observe another peak on the plot at

$$\begin{aligned} \psi &\rightarrow \psi - \pi, \\ \varphi_0 &\rightarrow \varphi_0, \end{aligned}$$

along with its symmetric solution. For the associated set of parameters (φ_0, ψ), the waveform is not invariant but the quantity $\langle s|s \rangle$ is.

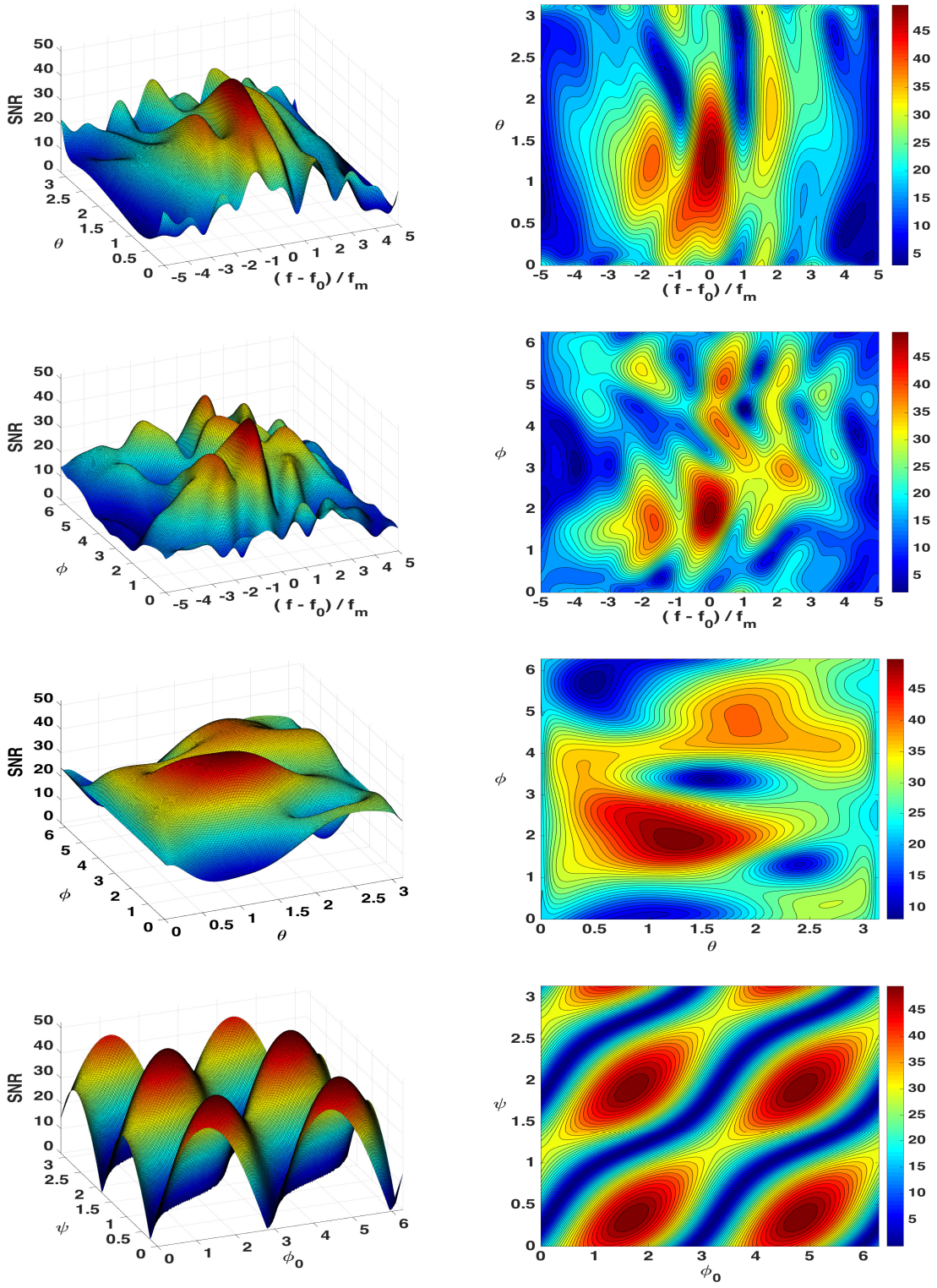

Figure 10.4: Plots of the multimodal SNR surface for one monochromatic ultra compact galactic binary illustrating the various secondary solutions. The left graphs are three dimensional illustration of the SNR surface and the right graphs are the mapped version. The upper panels illustrate the evolution of the SNR in terms of f_0 and θ keeping all the other parameters at their true value. The middle-up panels illustrate the evolution of the SNR in terms of f_0 and ϕ keeping all the other parameters at their true value. The middle-down panels illustrate the evolution of the SNR in terms of θ and ϕ keeping all the other parameters at their true value. Finally, the bottom panels illustrate the evolution of the SNR in terms of φ_0 and ψ keeping all the other parameters at their true value.

10.3 Presentation of the search algorithms

In the previous section, we saw that the problem of finding the solution requires to have algorithms capable of handling highly multi-modal surfaces. One possible solution for this is to use evolutionary algorithms (EAs). EAs refer to a group of stochastic, population based algorithms that mimic biological evolution and social behaviour to solve global optimization problems. An advantage of these algorithms is that they function with no a-priori knowledge of the problem at hand. This allows, in most cases, a very easy implementation of the algorithm. Generally, EAs display good convergence properties and have a small number of control parameters.

Previous studies in GW astronomy have used EAs, but have focused on the implementation of a single algorithm per source type [130, 146]. Our experience has shown that a possible better strategy is to use a combination of algorithms in the development of efficient, accurate search and resolution pipelines. This idea of creating hybrid EAs is not new, and has already been applied in a number of fields, including GW astronomy[147, 148].

Thus our goal has been to construct a hybrid algorithm, which predominantly uses a combination of Particle Swarm Optimization (PSO)[142, 143] and Differential Evolution to iteratively search for monochromatic galactic binaries. These two methods will be supported by a number of other techniques that will accelerate the convergence of the algorithm. In order to extract the best-fit parameters at the end of the search phase, we will use a combined Metropolis-Hastings - Differential Evolution Markov Chain. In this section we describe the PSO algorithm in its base form, before moving on, in the next section, to describing the full construction of our algorithm.

Even though the idea of using swarm intelligence and modeling has been present since a long time in computation field, the creation of the PSO is attributed to Kennedy, Eberhart and Shi. In their article written in 1995 [142], they explain their approach that led to the foundations of the algorithm as it is used nowadays. Their goal was to model human social behavior using bird flock movement as a guideline. In order to obtain a satisfactory model for their bird flock simulations, the authors proceeded iteratively with the addition of some key algorithmic concepts

- influence between adjacent neighbours
- introduction of random or stochastic components (called "craziness")
- simple equation of movements including one point of attraction for the flock
- creation of the concept of memory both for individual birds and the flock as a whole

Combining all these criteria together, the algorithm proved to be working well on bird flock simulations and even more since it appeared that it also presented abilities in optimization problems. Indeeed, when confronted with the optimization of non trivial and highly multi-modal functions such as the Schaffer's f6 function, usually used as a benchmark function for optimization performances, the PSO algorithm managed to efficiently find the global extremum. Since the publication of this paper, the particle swarm evolution has been applied to a lot of optimization problems. Thus, it has already been used in other fields of astrophyscs, such as pulsar timing [149, 150], ground based GW astronomy [151] and cosmic microwave background studies[152]. In all these works, the algorithm has been applied to find the extremum of likelihood function. As far as we know, this method has never been tested in the search for galactic binaries using a space based observatory.

Now that the foundations of PSO have been exposed, we are going to present in more details the features of the algorithm. As mentioned before, the PSO mimics swarm dynamics in nature to find the extremum of a surface. This surface is parametrized by the value of a so-called fitness function that depends on a number of model parameters. In our case, we equate the fitness function to the likelihood $\mathcal{L}(\lambda^{\mu})$. The motion of the swarm particles on the parameter surface is parametrized by a fictitious time parameter t_j, where j is the identity of the current step. Each particle is then evolved via the standard dynamical PSO equations:

$$\begin{aligned} X^i(t_{j+1}) &= X^i(t_j) + V^i(t_j), & (10.81)\\ V^i(t_{j+1}) &= wV^i(t_j) + c_1\xi_1(P^i(t_j) - X^i(t_j)) \\ &+ c_2\xi_2(G(t_j) - X^i(t_j)), & (10.82) \end{aligned}$$

where the dynamical variables for particle i are the instantaneous coordinate position $X^i(t_j) = \{\lambda_i^{\mu}\}$ and velocity $V^i(t_j)$. At each position of the particle on the parameter surface, we associate the corresponding value of the likelihood $\mathcal{L}\left(X^i(t_j)\right)$.

The velocity of the particle involves a number of quantities specific to PSO. As we explained before, a further specificity of the PSO algorithm is that each particle retains a partial memory of aspects of their personal history, while the swarm as a whole retains a sense of global history, both of which effect the velocity evolution of the particle. With this in mind, the first important quantities to define are the notions of the personal best, P^i, and group best, G, positions.

We define $P^i_{best}(t_j)$ to be the maximum value of the fitness function for particle i during the duration of its history, i.e.

$$P^i_{best}(t_j) = \max\left[\mathcal{L}\left(X^i(t_k)\right) \text{ , for } k = 0 \,..\, j\right]. \tag{10.83}$$

The personal best position P^i of a particle i at time t_j is then defined to be the position where the particle likelihood was equal to P^i_{best}

$$P^i(t_j) = X^i(t) \text{ if } \mathcal{L}\left(X^i(t)\right) = P^i_{best}(t_j). \tag{10.84}$$

Equivalently, we define $G_{best}(t_j)$ to be the maximum value of the fitness function of the swarm over its history, or in other words, the maximum among all P^i_{best}, i.e.

$$G_{best}(t_j) = \max\left[P^i_{best}(t_j) \text{ , for } i = 0 \,..\, N_p\right], \tag{10.85}$$

where N_p is the total number of particles in the swarm. The group best position is then obtained via

$$G(t_j) = P^i(t_j) \text{ if } P^i_{best}(t_j) = G_{best}(t_j). \tag{10.86}$$

Thus at any time, each particle of the swarm has a personal memory through P^i and a group memory through G.

The remaining terms in Eq. (10.82) are an inertia w, two acceleration constants (c_1, c_2), and two scaling factors (ξ_1, ξ_2). For these parameters, the standard values used in the literature are $w = 0.78$ and $c_1 = c_2 = 1.192$ and $(\xi_1, \xi_2) \in U[0, 1]$ [153]. This choice of parameters is believed to provide a good compromise between exploration and convergence for the swarm in a reasonable number of steps. However, as we will see later, these values can be made to take different values during the evolution of the algorithm.

On further investigation of Eq. (10.82), we see that there are three distinct contributions to the evolution of the velocity at each step :

- an inertial term that scales the current velocity with a factor w. If $w > 1$, the velocity will increase at each step and the particles are more likely to explore the parameter space, while for $w < 1$ the particles tend to converge to a single point in parameter space.
- an acceleration term towards the best position of the particle so far P^i. This term tends to make the particle explore the neighbourhood of a promising location in parameter space
- an acceleration term towards the best position of the swarm so far G. This term is shared between all the velocity equations of the swarm. This is the key term that creates the swarm dynamics: particles will tend to help each other to get to the best position of the group

Since the motion of the particles is governed only by the dynamical equations given above, it is essential to set boundary conditions so that the position of the particles stay within the physical parameter range of interest. Furthermore, these boundary conditions need to reflect the physical meaning of the parameters they are associated with. One common boundary used for a parameter taking value in an interval of $\mathbb{R}$ is the reflective boundary condition. In this case, if the particle crosses the physical boundary set for the parameter, the position of the particle is set at the boundary while the sign of its velocity is reversed, i.e.

$$\begin{aligned} X^i(t_{j+1}) &= X_{min} \text{ or } X_{max}, \\ V^i(t_{j+1}) &= -V^i(t_j). \end{aligned} \tag{10.87}$$

where X^i and V^i stand for the parameter position and velocity of particle i.

Finally we also set limits on the values that the velocity can take in order to prevent the particle from moving outside the physical boundary of the problem. If the velocity computed at time t_{j+1} is superior to a set maximum velocity V_{max}, the velocity is then set at this limit, i.e.

$$V^i(t_{j+1}) = \begin{cases} V_{max} & \text{if } V^i(t_{j+1}) \geq V_{max} \\ -V_{max} & \text{if } V^i(t_{j+1}) \leq -V_{max} \end{cases} \tag{10.88}$$

where $V_{max} = \frac{1}{4}(X_{max} - X_{min})$, X_{min} and X_{max} being the superior and inferior limit respectively for the given parameter.

As PSO was originally formulated to mimic flocks of birds, there is no real evolution as the population moves as a whole. At each point in time, the position of a particle is influenced by all other members of the swarm. This is contrast to most EAs. As we have seen above, the PSO is quite a simple algorithm as it is dependent on only three control parameters (w, c_1, c_2). The final dependency is the number of particles in the swarm, N_p. However, a downside of PSO is that there is no guide to choosing the optimum value of N_p. As we will see later, one needs to be careful when choosing N_p in order to achieve a balance between accuracy and runtime of the algorithm.

10.4 Building a hybrid swarm algorithm

10.4.1 Parameter space and detection threshold

In this section, we will now present how the previous algorithms can be combined in the framework of the detection and resolution of galactic binaries.

In Section 10.2.1, we have seen that the gravitational waveform of a circular monochromatic binary is parametrized by a set of seven parameters. This sets the dimensionality of the monochromatic galactic binary problem at $7 \times N$, where N is the total number of ultra compact binary sources in the Galaxy. In Section 10.2.3, we have seen that it is possible to maximise the likelihood over four out of the seven parameters. The total dimension of the space is then reduced to $3 \times N$.

While the F-Statistic is very useful in finding the maximum of the likelihood surface, it should not be used when mapping the posterior density. In fact, the maximisation of the extrinsic parameters prevents proper exploration of the posterior density and, as a consequence, parameter estimation can be incorrect [124]. However, unless otherwise stated, we will use the F-Statistic with all algorithms in the search phase of the pipeline.

To set a SNR detection threshold for eLISA, we ran a series of null tests. In this case, we assume that the output of the detector is composed of instrumental noise only, i.e. $s(t) = n(t)$. It has been shown that previous null tests for inspiralling supermassive black hole binaries in eLISA provide a detection threshold of $\rho = 10$ [154].

In this case we search the detector output using monochromatic waveforms. To conduct the null test, we used a modified DEMC algorithm. To encourage the movement of the chain we use a combination of simulated annealing as described in Chapter 6 and thermostated annealing [107, 134, 135]. As for simulated annealing, thermostated annealing replaces the factor of $1/2$ in the likelihood with an inverse temperature $\gamma = 1/(2T)$ that is defined by

$$\gamma = \begin{cases} \frac{1}{2} & 0 \leq \rho \leq \rho_0 \\ \\ \frac{1}{2}\left(\frac{\rho}{\rho_0}\right)^{-2} & \rho > \rho_0 \end{cases}, \tag{10.89}$$

where ρ is the SNR and ρ_0 is a threshold that needs to be chosen. In this case we took $\rho_0 = 1$ as we wanted the chain to begin exploring as quickly as possible in the beginning. To extract the threshold SNR, we then need to cool the surface down slowly using simulated annealing. For the null tests, we used 5×10^4 iterations each of thermostated annealing, simulated annealing and standard DEMC.

We ran the above algorithm 50 times, with different starting configurations. In each case, the algorithms returned "detections" with SNRs of $5.9 \leq \rho \leq 8.7$. To account for the possibility of higher values from different noise realisations, we thus decided to set the detection threshold at $\rho = 9$.

10.4.2 Single source detection

Our initial goal was to investigate and define regions of functionality for each component of the search pipeline. This 'top-down' approach would allow us to confidently detect a source within a small region of parameter space surrounding the binary, and then expand the size of the search space, solving problems as we go. With this in mind, we defined a search region around the binary in each case by $\lambda^i \pm n\Delta\lambda^i$, where $n > 1$. In our initial investigations, we defined $\Delta\lambda^i = \sigma_i$, where σ_i is the standard deviation calculated using the FIM.

We used two criteria to assess if the algorithm was working for a given width n: does it recover the SNR of the source and are the recovered parameters less than 3σ away from the true solution? If those

two conditions were met for repeated simulations, we increased the width of the search space. Thus we were able to control both how our algorithm works and observe exactly when it breaks down.

In order to test the performance of the algorithms, we started with a simple search where the data contained a single binary and an observation time of one year. Two different sources were selected for this first test phase. The first one was a fiducial low frequency source which allowed us to reduce runtime of the algorithm during development, and solve problems in our initial investigations quickly. The parameters for this binary were $\iota = 1.74, \phi_0 = 1.664, A = 1.2276 \times 10^{-20}, \psi = 1.884, f_0 = 5.2234 \times 10^{-4}\, Hz, \theta = 1.273$ and $\phi = 1.8845$, where the angular values are in radians. The signal had an SNR of $\rho \approx 54$. The second source was the WD-WD galactic binary, RXJ0806.3+1527. This source is one of the main verification binary candidates for eLISA and should be the one recovered with the highest SNR [155]. The parameters this binary are $\iota = 0.663, \phi_0 = 5.857, A = 6.378 \times 10^{-23}, \psi = 1.741, f_0 = 6.2203 \times 10^{-3}\, Hz, \theta = 1.162$ and $\phi = 3.612$ where the angles are in radians [156]. For one year of data, the SNR for RXJ is $\rho \approx 25$.

10.4.2.1 Initial search with PSO

The initial step in the construction of our algorithm was to implement the PSO to see how well it performs, on its own, in the detection of galactic binaries. There were two motivations to use PSO as a baseline algorithm :

- in comparison with MCMC based algorithms, the performances of PSO seemed to be more efficient in finding the maximum of complicated posteriors [152]

- the swarm-like feature of the algorithm, such as the personal and group best position, seemed to be powerful and could easily be adapted, or complemented, in other situations.

The set of parameters used for a single source PSO search is $\lambda^\mu = \{\ln f_0, \theta, \phi\}$. Regarding the boundary conditions, we applied the reflective boundary condition for frequency as given in Eq. (10.87). In terms of sky angles, we have seen that the colatitude θ stands in $[0, \pi]$ while the longitude is in $[0, 2\pi]$. However the search space cannot be divided into distinct intervals. In fact, these two parameters give the position of a point on a two sphere $\mathcal{S}^2$ of radius 1. That is why we decided to apply boundary conditions where particles can move freely across the boundary. In order to keep the sky angles in the range we set, we first map the sky angles on Cartesian coordinates (x,y,z) as

$$x = \sin(\theta)\cos(\phi), \tag{10.90}$$
$$y = \sin(\theta)\sin(\phi), \tag{10.91}$$
$$z = \cos(\theta), \tag{10.92}$$

and we map back the set of coordinates (x,y,z) to colatitude and longitude as

$$\theta = \mathrm{acos}(\phi), \tag{10.93}$$
$$\phi = \mathrm{atan}(y/x). \tag{10.94}$$

This transformation ensures that the sky angles are in their range.

For the first application of PSO, we focused solely on the low frequency source. We initially used 10 particles with standard control parameter values of $w = 0.72$, $c_1 = c_2 = 1.192$ for a total number of steps of 500. Using $\Delta\lambda^i = \sigma_i$, for every chosen value of width n, we launched ten simulations with different initial conditions. In this configuration, the PSO works well up to $n = 10^2$ (see table below), where one of the simulations did not converge to the true solution. This value of n corresponds to a $\sim 3f_m$ frequency bandwidth, where some of the secondary peaks in the likelihood, caused by the Doppler modulation, are now accessible to the particles. In the failed simulation, the swarm was not able to escape a secondary mode in the likelihood surface. In fact, for this number of σ, the range of sky angles accessible is now large enough to include the antipodal mode presented in Section 10.2.4.

Search domain	Convergence	$\left\| \left(f_{0,min/max} - f_0\right)/f_m \right\|$	$[\theta_{min}, \theta_{max}]$	$[\phi_{min}, \phi_{max}]$
±5 sigma	100%	0.064	[1.821, 1.947]	[1.181, 1.365]
±10 sigma	100%	0.128	[1.757, 2.011]	[1.089, 1.457]
±20 sigma	100%	0.257	[1.631, 2.137]	[0.906, 1.640]
±30 sigma	100%	0.385	[1.504, 2.264]	[0.722, 1.824]
±40 sigma	100%	0.514	[1.378, 2.390]	[0.539, 2.007]
±50 sigma	100%	0.642	[1.251, 2.517]	[0.355, 2.191]
±75 sigma	100%	0.963	[0.935, 2.833]	[0.000, 2.650]
±100 sigma	90%	1.284	[0.619, 3.142]	[0.000, 3.109]

Table 10.4: Convergence of PSO algorithm for various sizes of search space. The search space is symmetric around the true solution and its width is given in terms of sigma (from Fisher matrix) and real values of (f_0,θ,ϕ). The symmetric range of frequencies is given in modulus with respect to the true frequency in units of modulation frequency f_m. The values of θ and ϕ are given in radians

This result indicates that the particle swarm optimizaton in this form, is not able to find the global optimum in all cases. One possible solution for this problem is to increase the number of particles, N_p, thus giving the swarm more opportunities to find the main mode. This solution is not costless since it also increases the total computation time. This is why we decided to keep this as a last resort solution, and instead introduce a new control on the swarm. In order to fine-tune the behavior of the swarm we now allow the constant inertia w to vary over time, i.e. $w = w(t_j)$. In the dynamical equations, a value of inertia superior to 1 increases the exploration of the swarm, while convergence of the swarm is improved for inertias inferior to 1. A good compromise between exploration and convergence can be found by introducing a type of "inertia annealing". This method is not new and has already applied in other works, albeit in a different form [150]. Starting with an initial inertia w_i, we cool the inertia according to

$$w(i) = \begin{cases} w_f 10^{\log_{10}(\frac{w_i}{w_f})(1-\frac{i}{T_w})} & \text{if} \quad 0 \leq i \leq T_w \\ \\ w_f & \text{if} \quad i > T_w, \end{cases} \tag{10.95}$$

where $w_i = 1.2$, the final inertia is $w_f = 0.78$ and $T_w = 500$ is the cooling time (which we take to be equal to the total number of steps in the algorithm). With this new annealed version of the algorithm, the swarms now successfully found the source up to a value of $n = 10^4$. For this value of n, the search space now covers the full sky and is $\sim 300 f_m$ wide in frequency.

Beyond this value of n, we seemed to come to a natural limit in the PSO algorithm. The efficiency of the PSO is based on the ability of members of the swarm to find good positions in parameter space, and drag the whole set of particles towards them. One of the fundamental requirements is then to have a good exploration of the parameter space so that the swarm is not confined in some small part of the likelihood surface. For medium size widths such as the one with $n = 10^4$, we noticed that for some initial conditions, the swarm was not able to explore a large enough space to find the area of interests. In fact, for this range of parameter, the swarm has now access to all the secondary maxima described in Section 10.2.4 namely the subpeaks in frequency around the true carrier frequency combined with all the antipodal solutions corresponding to all these subpeaks. It is known that PSO can very quickly converge to secondary maxima, especially on a complicated surface. Once there, it becomes very difficult to move the algorithm on to a better solution. This is why we decided to introduce a DE step that would provide the swarm a greater ability to explore the parameter space.

10.4.2.2 Combining PSO with DE

In the form described earlier, the PSO algorithm is expressed in terms of evolution of a generation G to the next generation $G + 1$. This proper structure of the DE algorithm can be easily adapted to a swarm by replacing the notion of generation G by the position of the entire swarm at a specific time t_G. Thus evolving a generation G to $G + 1$ is equivalent to evolve the swarm from t_G to t_{G+1}. Thanks to that, we can now combine the two algorithms and control the evolution of the swarm by using either the PSO equations given in Eq. (10.82) or the DE equation given in Eq. (4.54).

The main objective for the mixing of these two algorithms was to introduce larger movements for the swarm in order to avoid getting stuck in one of the many local optima of the likelihood surface. For the

good integration of the two algorithms we identified a couple of points that needed to be adjusted in the DE algorithm. First of all, we decided to keep the structure of G and P^i inherent to the swarm but missing in the DE algorithm. Thus, at the end of each DE step, we decided to keep recording the positions of the best positions of the swarm such that a block of DE moves will improve the next dynamical equations of the swarm for the following PSO block. The other choice we made was to simplify the structure of the DE equation by first getting rid of the crossover step and by simplifying the mutation equation:

$$U^i(G+1) = X^i(G) + \gamma \left[X^k(G) - X^l(G)\right] \tag{10.96}$$

where U^i is the trial vector and i, k, l are mutually different and where we set the differential weight at its optimal value, $\gamma = 2.38/\sqrt{2D}$. This simplification of DE is motivated by the fact that the PSO is already a large source of mixing and crossover in the swarm. Finally, we also decided to use the thermostated and simulated annealing schemes for the DE.

We implemented our algorithm using two series of PSO and DE blocks of 125 steps each for a total number of 500 steps. As we have already described, both the PSO and the DE have their own specific annealing schemes. As we alternate between the PSO and the DE, we also need to alternate between the annealing programs. To ensure that the use of these schemes was optimal, they were implemented in the following manner : both the inertia annealing (for the PSO) and the thermostated/simulated annealing (for the DE) were run simultaneously. This ensured that each annealing scheme cooled over a long period. It also meant that when one scheme was in use, the other was running in the background, ensuring that when it came back into use, it would be at the same level as if it was being applied to a single algorithm pipeline only. So, for the PSO, we set $T_w = 500$. This means that at the end of the algorithm, the inertia is equal to the final value of $w_f = 0.78$, even though the actual total number PSO steps is 250. During the DE phase, we used a thermostated annealing scheme during the first 125 steps, with a threshold of $\rho_0 = 5$ to encourage movement in the parameter space. We then set $T_c = 375$ for the simulated annealing phase, meaning that at the end of the pipeline, the heat is unity. In this new combined configuration, the algorithm managed to always find the source for $n = 10^5$, which corresponds to a frequency bandwidth of $\sim 3000 f_m \approx 10^{-4}$Hz

At this point, we chose to test the algorithm on a full 1mHz frequency band to observe its performance. The interest here is to see if, given the width of the signal in comparison to the width of the search space, the algorithm would find the source, or repeatedly get stuck on strong peaks in the noise, i.e. a kind of "needle in an empty field" problem. Since the frequency band is now much wider, we decided to keep the structure of the previous algorithm but it quickly became clear that we finally needed to increase the number of particles in the swarm.

We thus changed the number of particles from 15 to 40 to facilitate a wider exploration (we will comment further on the number particles required below). In this configuration, the algorithm was doing quite well, but was not able to satisfy the two convergence criteria for all the simulations.

In fact, we observed two main weaknesses in the movement of the swarm: in the case where the source has not yet been detected, the group best position G may be lying on a distant noise peak. A particular particle may land in the vicinity of the source and in so doing, improve its personal best position P^i. However, as $\mathcal{L}(P^i) < \mathcal{L}(G)$, the group best position will never move into the area of interest. Furthermore, while P^i stays close to the source, the particle itself wanders off to explore the rest of the parameter space. If a new personal best position is found, it can actually pull P^i away from the source, delaying detection. In the second case, a source can be detected (i.e. its SNR beats the threshold), but the group best position ends up on a secondary solution. In order to improve on this current solution, we have to wait until one of the particles lands at random on a better solution and thus improving $\mathcal{L}(G)$.

Upon investigation, the culprit in both cases was the width of the search window. A wide search window allows for a very diffuse swarm. This means that any chance of a local exploration is dramatically reduced. In this case, particles will visit the region of parameter space close to the source once, and then maybe never again after. One solution would be to over-populate the search space with particles, increasing the possibility that certain regions would be visited by many particles, many times, during the runtime of the algorithm. However, this would lead to a very slow algorithm. So, in order to solve these two issues, while keeping the number of particles small, we introduced two further features to the algorithm that enforce a more local exploration and reduces the width of the search space.

10.4.2.3 Uphill Climber

The motivation for this type of move is fact that the swarm algorithm's lack of local exploration possibilities. In the previous test cases, the frequency band was small enough that the swarm was able to locally

explore good positions on the likelihood surface. For wider frequency bands, the swarm is more diffuse, and the PSO-DE blocks do not provide an opportunity for the swarm to locally explore the personal best positions P^i.

The idea was then to implement a new block of MCMC-like movements that would be related, not to the individual swarm particles, but directly to the P^i. Thus, at some time t_j, we stop the global movement of the swarm and locally explore all P^i positions that have been spotted so far with the PSO and DE blocks.

The uphill climber (UC) scheme was first introduced in [148] as a greedy criterion proposal, i.e. if the new solution has a higher fitness than the starting solution, move there immediately. In the current context, this move is implemented as follows : for each $P^i(t_j)$, and as we are using the F-Statistic, calculate the FIM on the 3D projected subspace at that point, i.e. iteratively project the D-dimensional FIM onto a $D-1$ subspace according to

$$\Gamma^{(D-1)}_{\mu\nu} = \Gamma^{(D)}_{\mu\nu} - \frac{\Gamma^{(D)}_{\mu\kappa}\Gamma^{(D)}_{\nu\kappa}}{\Gamma^{(D)}_{\kappa\kappa}}. \tag{10.97}$$

As with the MCMC algorithm, use the eigenvalues and eigenvectors of the projected matrix to construct jump proposals, and accept or reject according to the greedy criterion $\mathcal{L}(\lambda^{new}) > \mathcal{L}(\lambda^{old})$.

In order for the UC to be effective, it needs to be implemented a number of times successively as the acceptance rate is usually quite low ($< 10\%$). Although low, and probably coming from the fact that we do not update the eigenvectors and eigenvalues after a move has been accepted, if the UC is used N_{UC} times in sequence (with $N_{UC} \geq 10^2$), it can easily move the position of P^i between $10-30$ f_m in frequency. This greatly accelerates the convergence of the algorithm. To make the UC fit into the overall structure of the pipeline, and especially with the annealing schemes, we count the N_{UC} iterations as one step in the algorithm. This means that we take the thermostated/simulated annealing temperature at t_j and keep it constant during the UC moves.

10.4.2.4 Swarm Strengthening and Culling

The second feature was introduced due to the fact that the likelihood surface has many secondary peaks owing to the multi-modal nature of the detector response. One of the strongest secondary solutions comes from the Doppler modulation of the phase. For some simulations, the swarm was able to get close to the source but was stuck on one of these Doppler induced peaks. As mentioned above, when the frequency band is large, even with a local exploration of the likelihood surface, only a handful of particles explore the area of interest, and those that do so may converge to one of the secondary solutions. In order to prevent this situation, we decided to add a second phase in the overall algorithm where we strengthen a good solution by moving half of the the swarm to an interval of frequency around G given by $[f_0(G) \pm 10 f_m]$. The size of this band is motivated by the assumption that, if this feature is used late enough in the pipeline, the principal mode should not be further away than a few f_m from the secondary solutions.

Furthermore, while it is clear that an increased number of particles is necessary in the early stages of the pipeline to ensure a wide exploration of the parameter space, later on, some of these particles can converge to low fitness regions of the likelihood surface. Seeing as they never evolve to the region of interest during the runtime of the pipeline, we cull the 50% of the swarm that is not drawn to the region around G. This also helps speed up the runtime of the algorithm in the second phase of the pipeline.

10.4.2.5 Single source search over a 1 mHz band

As the primary goal of this study is accuracy, rather than speed, the final version of our algorithm is as follows : the pipeline is now 1250 steps long, beginning with 40 particles. In the first phase, the structure is two blocks of 150 steps each of PSO and DE, followed by 200 steps of UC. For the PSO we use an inertia annealing scheme with $T_w = 750$, while for DE and UC we use a thermostated annealing scheme. At the end of this phase, we move the swarm that is now 20 particles in size to the vicinity of G and kill the other 20 particles. In the second phase of the pipeline, we have a structure of 75 steps of PS0, followed by 75 steps of DE and 100 steps of UC. For DE and UC, we use a simulated annealing scheme with an initial temperature equal to the temperature at the end of the first phase and a cooling time of $T_c = 250$.

The search algorithm was then tested for the detection of the low frequency source, using a 1 mHz search band. We ran 10 simulations with different initial conditions, and in each case recovered the source according to both detection criteria. For this source, the total runtime for the search algorithm was 1

	ι	φ_0	A	ψ	f_0/mHz	θ	ϕ	SNR
Fiducial source	1.74	1.664	1.2275×10^{-20}	1.884	0.52234	1.273	1.8845	53.92
	1.734	4.873	1.2525×10^{-20}	0.3268	0.52233953	1.259	1.900	53.95
	1.735	4.873	1.2508×10^{-20}	0.3268	0.52233953	1.258	1.901	
RXJ0806.3	0.663	5.8565	6.378×10^{-23}	1.74	6.2202766	1.1624	3.6116	24.93
	1.143	1.785	9.769×10^{-23}	0.742	6.2202745	1.1600	3.6236	25.53
	1.120	1.737	9.550×10^{-23}	0.746	6.2202751	1.1590	3.6213	

Table 10.5: Values recovered for the two sources both at the end of the search and parameter estimation phases. The first line is the true value, the second the one at the end of the search phase and the third the recovered value from parameter estimation. The values for RXJ0806.3+1527 are taken from [156]. We provide the recovered results for two simulations that end up on the response invariant symmetric solution $(\phi_0, \psi) \rightarrow (\phi_0 \pm \pi, \psi \pm \pi/2)$.

minute on an Intel Xeon 2.6GHz processor (15 minutes when using time domain responses). We then ran the algorithm on the more realistic source, namely the WD-WD verification binary RXJ0806.3+1527. For this source, and due to the much higher sampling rates required, the total time for the search algorithm was 12.5 minutes (3 hours when using time domain responses). Again, in all simulations, the algorithm was able to find the global solution. In Table 10.5 we again present both the injected and recovered parameter values at the end of the search algorithm for the two sources. We should point out that the recovered results presented are for two simulations that ended up on a $(\phi_0, \psi) \rightarrow (\phi_0 \pm \pi, \psi \pm \pi/2)$ symmetric solution that leaves the response $s(t)$ invariant as shown in Section 10.2.4.

In Figure 10.5, we plot the frequency evolution for one of the swarm particle $X^i_{f_0}$ when searching for RXJ0806.3+1527. In addition to that, we also plot the associated personal best position $P^i_{f_0}$ along with the group best positionG_{f_0}. We see that the movement of the particle is covering all the 1 mHz band frequency during the first part of the research, namely between 1 and 600 steps, which indicates a good exploration of the combined PSO-DE algorithm. It seems that there are no variations during the second part of the research (600 - 750 steps) but this is due to a scale effect owing to the restriction of the search space width from 1 mHz to $20f_m$ around G. In addition to the movement of the particles $X^i_{f_0}$, we see that $P^i_{f_0}$ is also evolving and exploring a wide range of frequencies through specific positions. Finally, this plot reveals the various structure of the algorithm where we see medium exploration up to 0.5 mHz during the two PSO blocks (0-150 and 300-450 steps), and larger explorations up to 1 mHz during the DE blocks (150-300 and 450-600).

In order to better understand the movement of the swarm as a whole, we have extended the previous graph in Figure 10.6, by adding the evolution of sky angles (θ, ϕ) and SNR along with two additional particles. The upper panels display the particles movements, the middle ones their associated personal best positions and the bottom ones the group best position. In all these graphs, the true values is represented with a dashed orange line. Once again, we observe that the algorithm is able to explore the whole parameter space for all the three search parameters allowing the swarm to spot interesting positions through the personal best positions. However, we see that the signal to noise ratios for the particles in the first part of the search never gets above 6. The major gain in SNR for the personal best position comes from the inclusion of the UC block that drag both the P^i and G to increased SNR. This is clearly illustrated on the graph at step 300 where the personal best position SNR of particle 1 goes from 5 to 16 during the UC block, making it the new group best position. This again justifies the inclusion of this type of move. Secondly, we observe that the algorithm first needs to lock on frequencies before trying to find the good values for sky angles. This is entirely due to the specific nature of the likelihood surface as described in Section 10.2.4. As an example, we see that after moving the swarm close to the source (> 600 steps), the personal best colatitude position for particle 3 is not entirely locked on the true values inducing a reduced SNR of 20 compared to the true SNR of 25. Finally this plot is a good illustration of the various levels of dynamics and evolution of the swarm. From top to bottom, the particle movements are responsible for most of the swarm exploration that enables to find positions of interest. Then the personal best positions still have a good exploration potential though reduced around some specific positions, whether they are strong noise peaks or signals. However their exploration is way more accurate than the ones from the particles with higher SNR. And finally, the group best position is only exploring the most promising position with high level of accuracy.

We should take some time here to talk about both the number of particles in the swarm, N_p, and the stopping criterion used for the pipeline. For the algorithm described above, we initially used 40 particles,

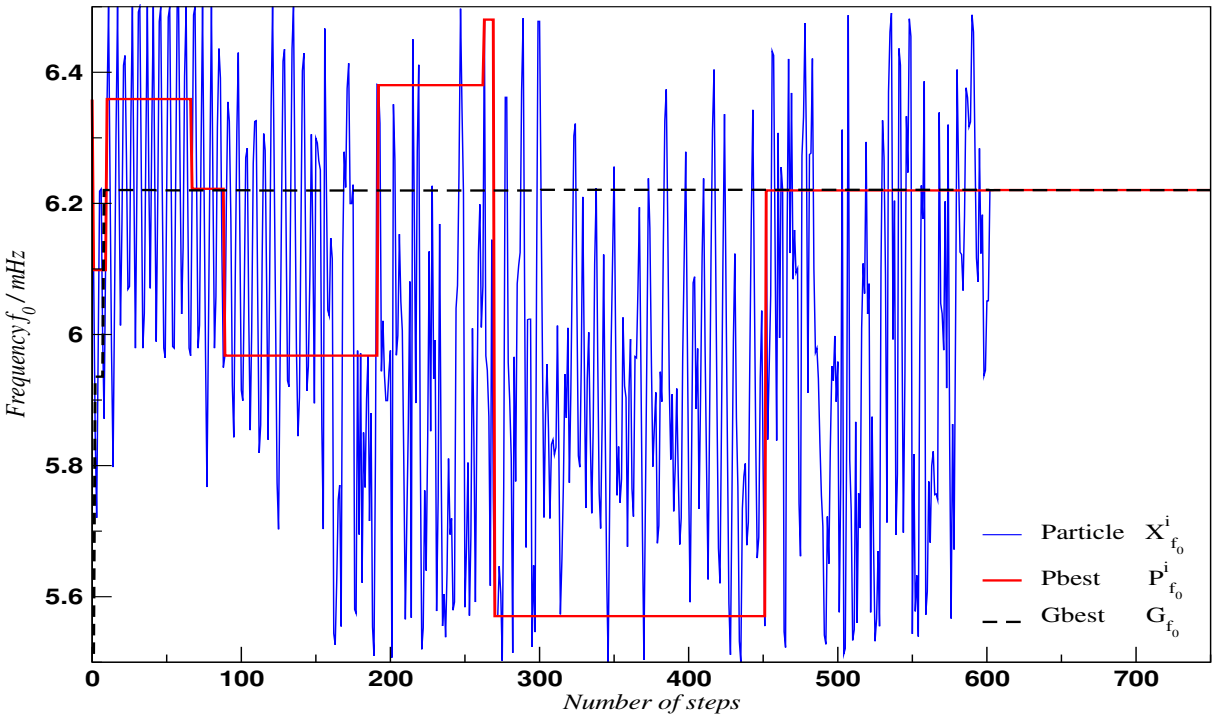

Figure 10.5: A plot of the movement of frequency for one particle from the swarm, $X^i_{f_0}$, with its personal best position, $P^i_{f_0}$, and the group best position, G_{f_0}, during the search for RXJ0806.3+1527 as a function of the number of steps in the algorithm.

before culling the swarm to 20. In order to test for accelerated convergence, we ran the algorithm with different numbers of initial particles, up to $N_p = 10^2$. In no circumstance did we observe an acceleration of convergence that would convince us start with more than 40 particles. As expected however, we did see an increase in the runtime of the pipeline, that came without a corresponding increase in convergence or accuracy. As with most search algorithms, there is no pre-defined stopping criterion. This case is identical. An investigation of quantities such as P^i or G did not suggest any obvious way of using these parameters as stopping criteria. In this particular study, as we know the true values a-priori, the number of steps in the algorithm was chosen to ensure that we always found the true source. We intend to investigate a more rigorous stopping criterion in a future work.

10.4.2.6 Parameter estimation

In keeping with recent articles on supermassive black hole binaries [64, 157], we also carried out a full statistical analysis using Markov chains for each source. Starting from the recovered values obtained at the end of the search algorithm, we ran a 10^6 steps combined MCMC and DEMC chain. From the chain, we can then draw the distributions for the binary parameters and use Bayesian tools such as chain mean/median and credible intervals. For sky angles θ, ϕ, instead of presenting the values of the recovered angles (mean/median), we will rather plot the orthodromic distance between the true and recovered sky positions as given by the Vincenty formula

$$\Delta\sigma = \arctan\left(\frac{\sqrt{\left(\cos\phi_R \sin\Delta\theta_L\right)^2 + \left(\cos\phi_T \sin\phi_R - \sin\phi_T \cos\phi_R \cos\Delta\theta_L\right)^2}}{\sin\phi_T \sin\phi_R + \cos\phi_T \cos\phi_R \cos\Delta\theta_L}\right), \tag{10.98}$$

where we define the the true longitude ϕ_T and latitude θ_T of the source, the recovered longitude ϕ_R and latitude θ_R, and $\Delta\theta_L = \theta_L^T - \theta_L^R$, and the value of $\Delta\sigma$ is in radians. Similarly, we will rather use positional resolution of the source instead of credible intervals for the sky angles. We can define an error box in the sky according to [108]

$$\Delta\Omega = 2\pi\sqrt{\Sigma^{\theta\theta}\Sigma^{\phi\phi} - \left(\Sigma^{\theta\phi}\right)^2}, \tag{10.99}$$

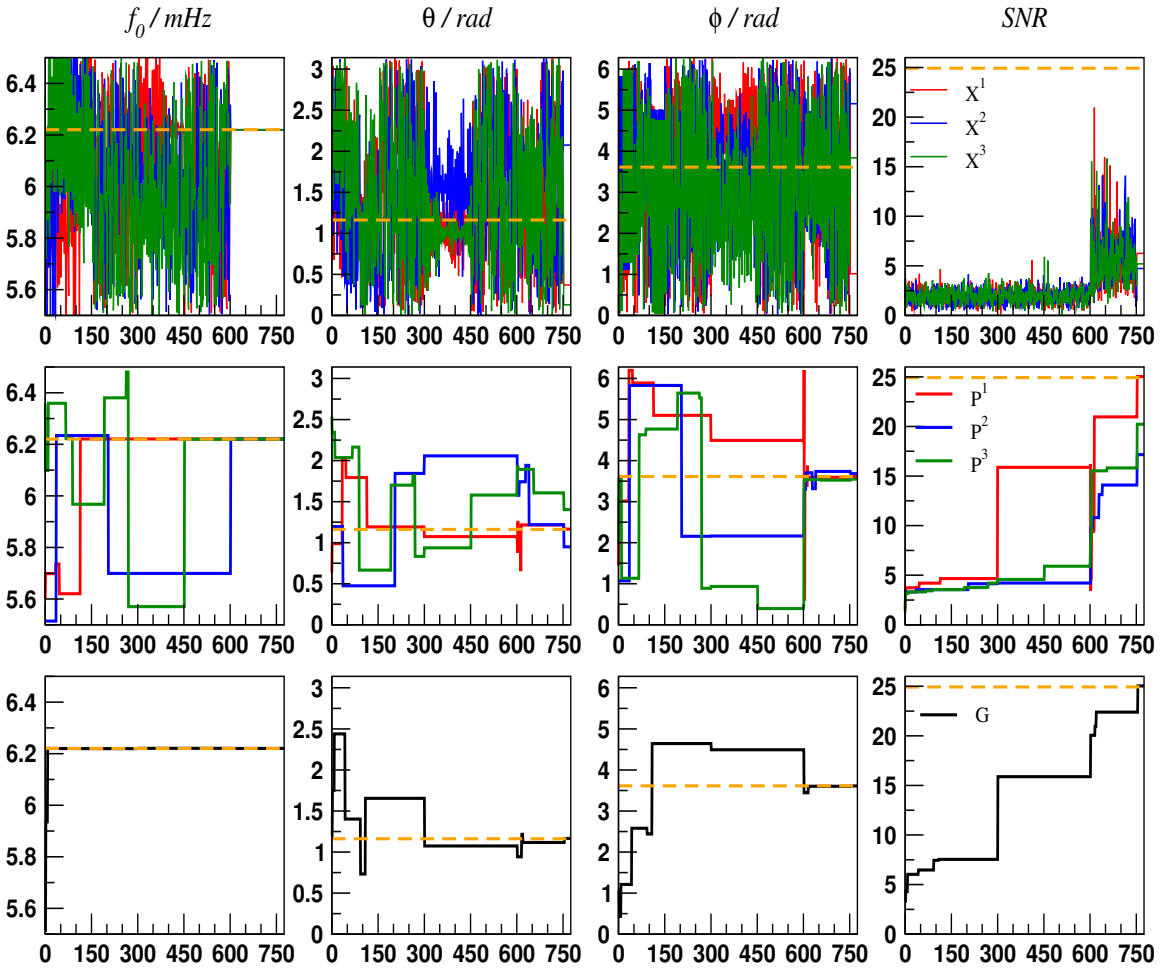

Figure 10.6: A plot of the movement for parameters (f_0, θ, ϕ) and SNR for three particle from the swarm, X^i, with its personal best position, P^i, and the group best position, G, during the search for RXJ0806.3+1527 as a function of the number of steps in the algorithm. In each cell the true values are represented by the dashed (orange) lines.

where

$$\Sigma^{\theta\theta} = \langle \Delta\cos\theta \Delta\cos\theta \rangle \,, \tag{10.100}$$

$$\Sigma^{\phi\phi} = \langle \Delta\phi \Delta\phi \rangle \,, \tag{10.101}$$

$$\Sigma^{\theta\phi} = \langle \Delta\cos\theta \Delta\phi \rangle \,, \tag{10.102}$$

and $\Sigma^{\mu\nu} = \langle \Delta\lambda^\mu \Delta\lambda^\nu \rangle$ are elements of the variance-covariance matrix, calculated directly from the DEMC chains themselves.

For the low frequency source, all distributions had low values of skewness and kurtosis. However, for RXJ0806.3+1527, the posterior distributions were highly skewed and/or had a high kurtosis for a number of parameters. In Figure 10.7, we plot the distributions of the seven parameters for RXJ0806.3+1527. We can see that while most of the distributions are close to be Gaussian, the distributions for amplitude and inclination are indeed skewed. The statistical analysis from the chain gives values of skewness of -2.29 and -0.62 for inclination and amplitude respectively. In addition to that, the chains also present high values of kurtosis for these parameters; 13.58 for inclination and 1.34 for amplitude. This demonstrates that the FIM should be avoided as a parameter estimation tool for compact galactic binaries and the analysis requires to use Bayesian analysis. In Table 10.5, we give the values that we recovered at the end of the parameter estimation for the two sources.

Finally, we checked how good our posterior sampling is given our finite number of 10^6 steps for the chain. In Figure 10.8, we plotted the evolution of the cumulated chain mean as a function of the number of steps for (ι, A, f_0, θ, ϕ). We observe that the means present large fluctuations for chain sizes between

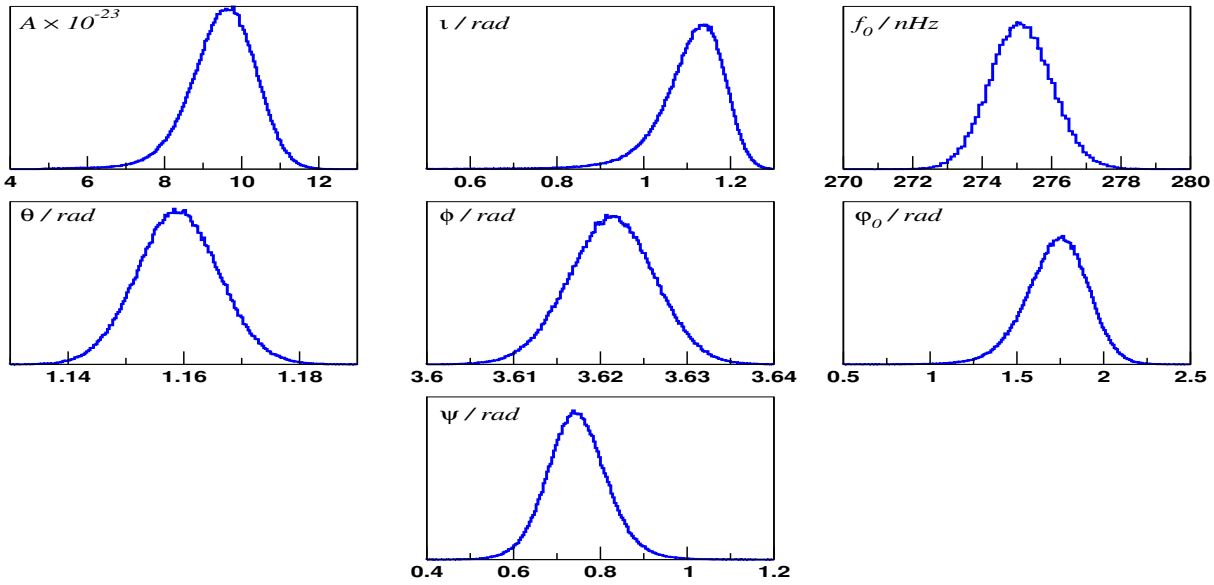

Figure 10.7: Parameter distribution from the MCMC/DEMC chain of RXJ0806.3+1527. The amplitude is scaled by 10^{-23} and the frequency is given in terms of delta where the frequency is defined by $f_0 = 6.22 + \delta$ mHz with δ in nHz. All the other parameters are in radians. The order of the parameters from left upper panel to right lower panel is: A, ι, f_0,θ, ϕ,φ_0 and ψ

1 and 10^5 steps. For instance, in this step range, the inclination mean is covering the interval between 0.4 and 0.5 before getting stable around 0.44. It is clear then that we need at least a 10^5 steps long chain in order to have a stable value of the mean for all the parameters hence indicating a good sampling of the posterior.

A complementary test is to draw the evolution of one parameter distribution from the chain as a function of the number of steps. In Figure 10.9, we have plotted the chain histograms for the colatitude of RXJ0806.3+1527 using various sizes of chains. For a 10^6 chain long, we get back the distribution presented in Figure 10.7. We see that both the form and density of the histograms are wrong when down sampling the posterior. In fact, with 10^3 and 10^4 points, the width of the distribution is reduced and the distributions have not smooth with peaks. The 10^5 chain gives back a much better distribution but is still far from being smooth with still small overdensities appearing. This study shows, as expected, that a non sufficient long chain will give wrong values for median and credible intervals. These two tests prove that it is indeed necessary to use at least 10^6 steps long chain MCMC in our parameter estimation.

10.5 Multi sources search

10.5.1 Presentation of the problem

After the single source search, we decided to test our algorithm in a situation closer to reality where the data set contains several binaries. We have built two data sets with different attributes

- The first set of data contains 18 sources in a frequency band of 1 mHz width from 0.5 to 1.5 mHz. The SNR of the sources have been taken between 10.5 and 28. The minimum distance in frequency between two sources is $273 f_m$, meaning that there is no confusion between the sources, but there is plenty of opportunity for the algorithm to get stuck on a strong peak in the noise.

- The second data set contains 30 sources in a frequency band of $10^3 f_m \approx 30 \mu Hz$, centered at 1 mHz. The SNR of the sources have been taken between 10.9 and 35. The minimum distance in frequency between two sources is $9 f_m$, meaning that there is now a mild confusion between the sources.

At the end of the pipeline described above, we are faced with two practicalities that require different treatments. The first is that we need to have a way of quickly and accurately subtracting the recovered source, before moving onto the search for the next source. The other is a full statistical analysis of the recovered source.

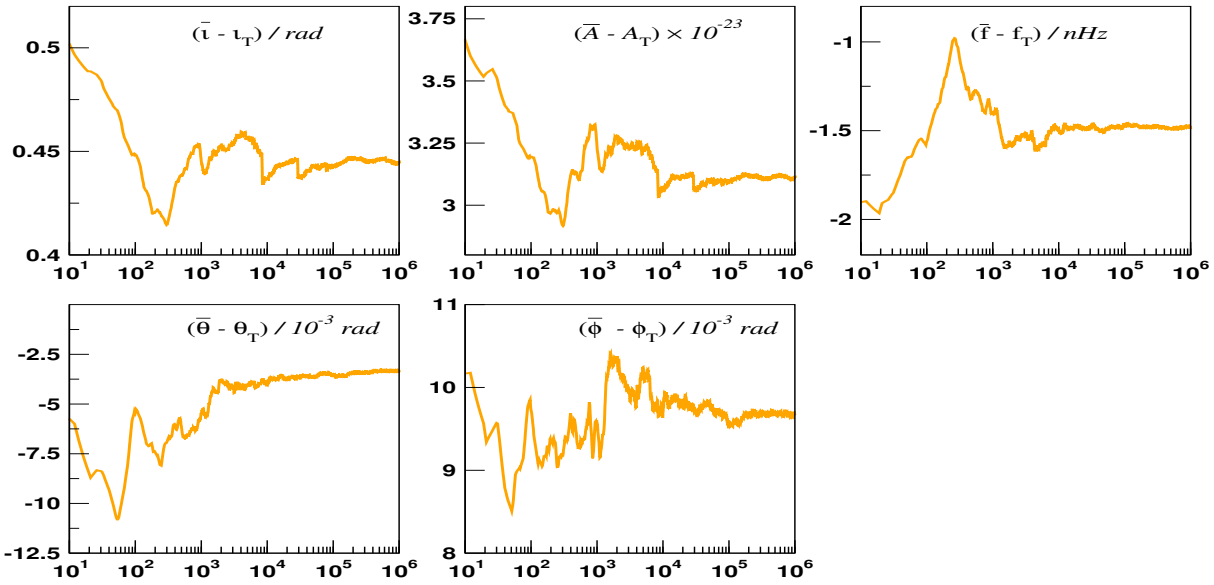

Figure 10.8: Plot showing the relative difference between the mean of the chain and the true value for RXJ0806.3+1527 set of parameters (ι, A, f_0, θ, ϕ) in terms of chain length.

To complete the first task, at the end of the search phase, we run a 4×10^4 iteration DEMC with a constant inverse temperature of $\gamma = 1/2$. This gives the chain a chance to become statistically independent. We then run a further 2×10^4 iterations where we superfreeze the chain to extract the Maximum Likelihood Estimator (MLE)[124]. This is done by using a simulated annealing phase, where we use an inverse temperature of $\gamma = 1/(2T)$, starting with $T_i = 1$ and ending with $T_f = 10^{-5}$. At the same time, to conduct the statistical analysis and calculate the credible intervals, we run a 10^6 iteration DEMC, where we neglect the first 2×10^4 iterations as 'burn-in'.

10.5.2 Data Set 1

For the first data set with no confusion, we detected 17 sources at the main frequency peak and one source on a secondary shifted frequency peak. The recovered values at the end of the search phase can be found in Table D.1 in Appendix D. In Figure 10.11, we plot the (true subtracted) recovered median values for the parameters (A, ι, f_0), i.e. $\Delta\lambda = \tilde{\lambda}_R - \lambda_T$, as well as the 99% credible intervals, as a function of the recovered binary. We also plot the sky error box $\Delta\Omega$, the orthodromic distance between the true and recovered sky positions $\Delta\sigma$, as well as the SNR. For all the following graphs, we present both the version where we used the time and Fourier domain responses. Depending on the generation method, the order in which we found the binaries changed. However, both methods give concording results indicating a good agreement in the two methods. From now on, all the results are discussed using the results we had from Fourier generation.

For the amplitude and inclination, all the injected source values are contained in the credible intervals, except for source 6. For this source, the binary has almost no inclination. This leads to an extremely high anti-correlation between the parameters of -0.949. An inspection of the correlation matrices for the binaries using the information from the chains, we noticed that a high correlation value between amplitude and inclination leads to larger credible intervals. For this data set, we found nine binaries with correlations between A and ι superior to 0.9 in absolute value. In addition, most of the credible intervals for inclination are not symmetric around the recovered values, due to the high skewness and/or kurtosis of the posterior distributions. This confirms that credible intervals are indeed important to use in this situation. In terms of frequency, all the frequencies are recovered in the interval except for binary number 10. A closer inspection reveals that the median frequency from the chain is at 3.95σ away from true frequency, where σ is the standard frequency deviation from the chain. However, the signal to noise ratio of the recovered signal is the same as the source and the residual is below the noise. This means that we indeed have a detection. In the majority of cases, the recovered median frequencies were within 7 nHz, or $0.2 f_m$ of the true values. We should point out that source 18 is not represented in the frequency cell. We will tackle this binary separately below.

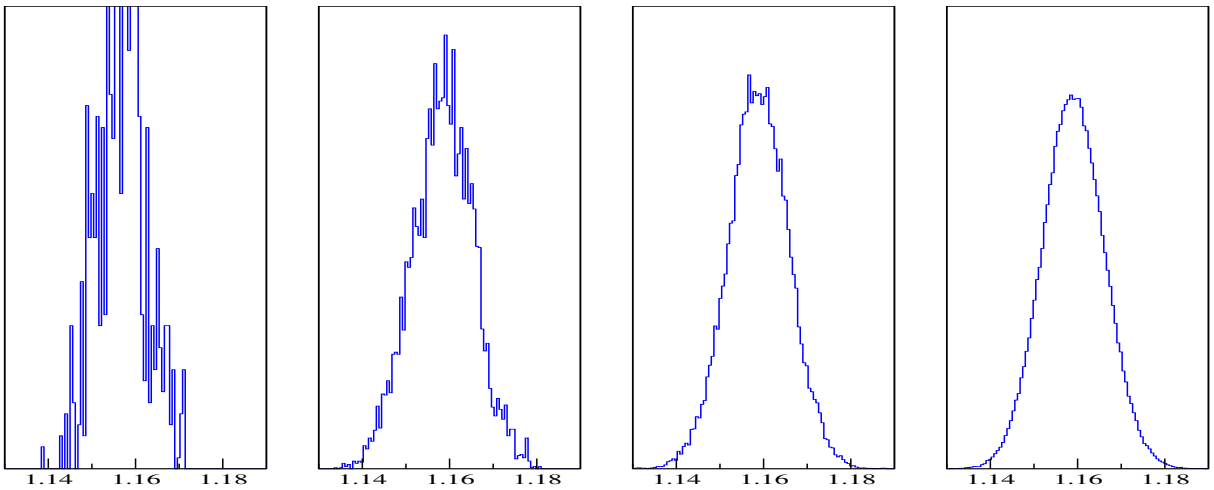

Figure 10.9: Plot showing the evolution of the distribution obtained from the chain for the colatitude of RXJ0806.3+1527 using 10^n steps long MCMC/DEMC chain with $n = 3, 4, 5, 6$.

If we now focus on the bottom row of Figure 10.11, we can see that, as expected, the sky error box tends to grow as a function of diminishing SNR. In almost all cases, the sky error box is smaller than 100deg^2, with some of the high SNR sources having error boxes of $\sim 10\,\text{deg}^2$. One binary (source 11) has a large error box ($\sim 150\,\text{deg}^2$). On inspection, we observed a large imprecision in the resolution of co-latitude for this source, but the recovered value was within a 3σ of the true value. For the orthodromic distance, we can see that the recovered sky solutions are usually within $\sim 10\,\text{deg}$ of the true value. In the cell representing the signal-to-noise ratios, it is interesting to see that most of the binaries have been found in order of SNR. This essentially means that our search algorithm is able to locate the brightest signal on a 1 mHz band and finds the maximum of the fitness function in a noisy background.

The final source (#18), where we ended up on a secondary Doppler peak, was the binary with the lowest injected SNR of the set ($\rho_T = 10.58$). In all of our simulations, the MLE frequency extracted at the end of the search phase corresponded to a frequency that was $\sim 3f_m$ away from the true value. However, the recovered SNR from the MLE ($\rho_R = 11.08$) was actually higher than the injected value due to noise. It has been pointed out in the literature that this situation can happen, especially for sources with SNR close to the threshold [130]. As a sanity check, we ran 10 simulations with different noise realizations and tried to find this single source. For each simulation we managed to find the source on the true frequency peak, confirming that it was indeed a problem of noise realization, and not a fundamental problem with the algorithm.

To further test how we had done, we also computed the overlap between the true and recovered signals, where the overlap between h_T and h_R is given by

$$\mathcal{O} = \frac{\langle h_T \,|h_R\rangle}{\sqrt{\langle h_T \,|h_T\rangle \,\langle h_R \,|h_R\rangle}}. \tag{10.103}$$

The 17 fully recovered sources had overlaps above 0.9, with values as high as 0.999 for the highest SNR source. For source 18, the overlap was 0.82.

We can see what this means graphically by plotting the residual of the subtracted data set against the original signal and the instrumental noise. The residual power is an indication of how much we have disturbed the data set through the imperfect subtraction of a source. We define the residual as

$$\tilde{r}(f) = \tilde{n}(f) - \left(\tilde{s}(f) - \sum_{i=1}^{N_s} \tilde{h}_i^{MLE}(f)\right). \tag{10.104}$$

with N_s the number of recovered sources. In Figure 10.14, we plot the power spectra for the total signal with noise $\tilde{s}(f)$, the instrumental noise only $\tilde{n}(f)$ and the residual $\tilde{r}(f)$. If the residual is below the level of the noise, then the substraction process has been successful. For the first data set, we can see that the level of the residual is always below the level of noise even for the binary where the recovered frequency was $3f_m$ away.

We also carried out one final extra check to test the performance of the subtraction process. We ran 40 simulations, using different initial configurations, on the binary subtracted residual data set. If we hadn't properly extracted all binaries, we should in this case "detect" another source. However, all simulations ended up with a "detection" with signal to noise ratio inferior to the SNR threshold. Moreover, as some of the returned solutions were clustered around specific frequencies, we inspected these values and confirmed that they were pure noise peaks at more than $50 f_m$ away from any injected frequency signals.

10.5.3 Data set 2

For the second data set, the algorithm managed to find all sources on the $10^3 f_m \approx 30\mu$Hz band. The recovered values at the end of the search phase can be found in Table D.2 in Appendix D. In Figure 10.13, we again plot the credible intervals, sky errors and SNR. For this data set, all the true values lie in the 99% credible intervals for amplitude, inclination and frequency. Once again we observe that the widths of the credible intervals for ι and A depend on the values of these parameters and their correlation. For instance, the largest credible interval for amplitude (binary 15) corresponds also to a high correlation between amplitude and inclination of 0.951 and a high asymmetry in the posterior density. In terms of frequency, the recovered values are very good and all lie at less than 3 nHz or 0.1 f_m from true frequency.

For the sky positions, the sky error boxes have expected values. For this data set, the orthodromic distance between injected and recovered sky positions is lower, with nearly all sources recovered within 5 deg of the true value. Once again the size of the sky error box increases as the signal-to-noise ratio decreases, and the binaries we again found in order of decreasing SNR. In this case, and probably due to the smaller search band, the overlaps between the recovered and the true signals were once again extremely good and higher than 0.95 for all binaries.

In Figure 10.15, we again see that the residual power is below the noise, suggesting a minimal disturbance of the data set during source subtraction. And once again, running a sanity-test search on the source subtracted data set, we found that all simulations converged to a noise peak. Moreover, in this case as the noise realization was different, all of our simulations converged to the same frequency peak, which returned an SNR value of $\rho = 5.52$, clearly below the threshold for detection.

10.6 Conclusion

In this chapter, we have explained how we developed a hybrid swarm-based algorithm for the detection of gravitational waves emitted by ultra compact monochromatic galactic binaries, mixing together evolutionary algorithms such as particle swarm optimization and differential evolution, with Markov Chain Monte Carlo methods. We demonstrated the ability of this algorithm to detect a single source on a 1 mHz band using a fiducial low frequency source and the verification binary RXJ0806.3+1527 in the framework of the future eLISA mission. We then showed how this search algorithm was able to perform well in the situation where the signal contains several sources. We used two data sets containing 18 and 30 sources on frequency bands equal to 1 mHz (no confusion) and $10^3 f_m \approx 30\mu$Hz (mild confusion) respectively. We successfully recovered all sources, and using a full Bayesian analysis, we demonstrated that the median values for all the recovered binaries were within a 99% credible interval of the injected values. This demonstrated, that while unoptimized, the algorithm works well in the iterative search for GB sources.

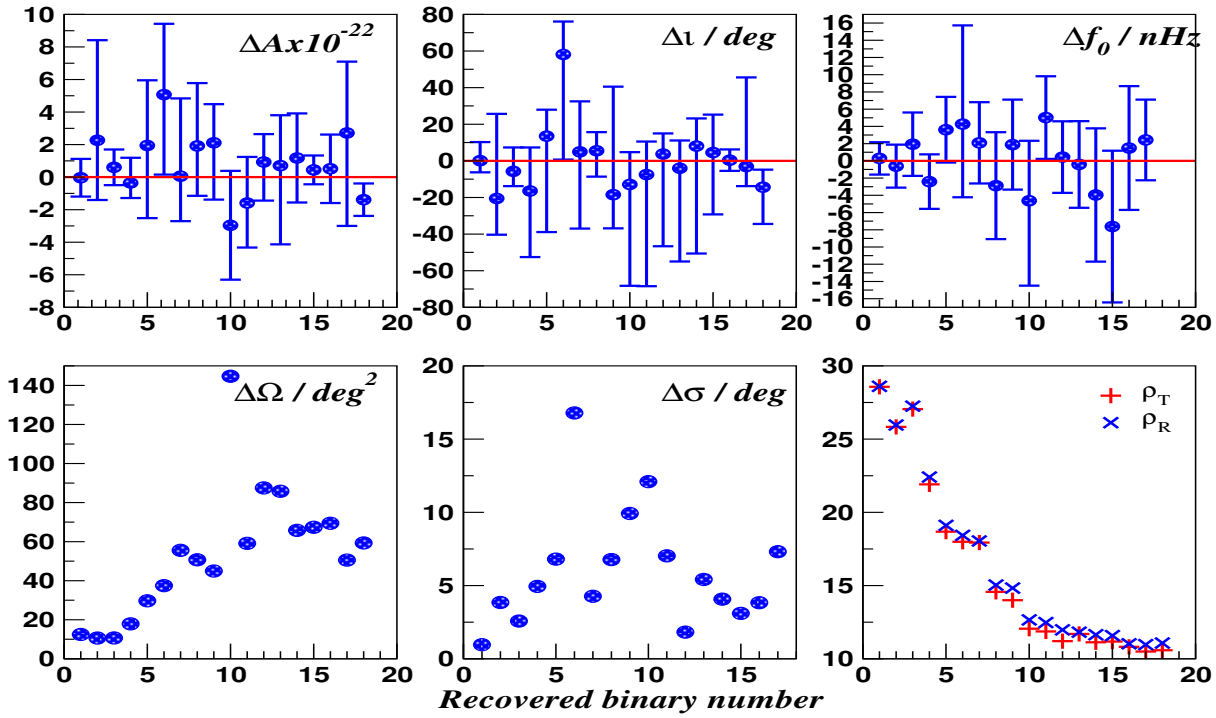

Figure 10.10: Results obtained from the recovered sources in data set 1 using time domain waveform responses. In the top row, we present the 99% credible intervals and (true-subtracted) median values for amplitude (top left), inclination (top middle) and frequency (top right). In the bottom row, we present the values of the sky error boxes, the orthodromic distance and the values of the true and recovered signal-to-noise ratios.

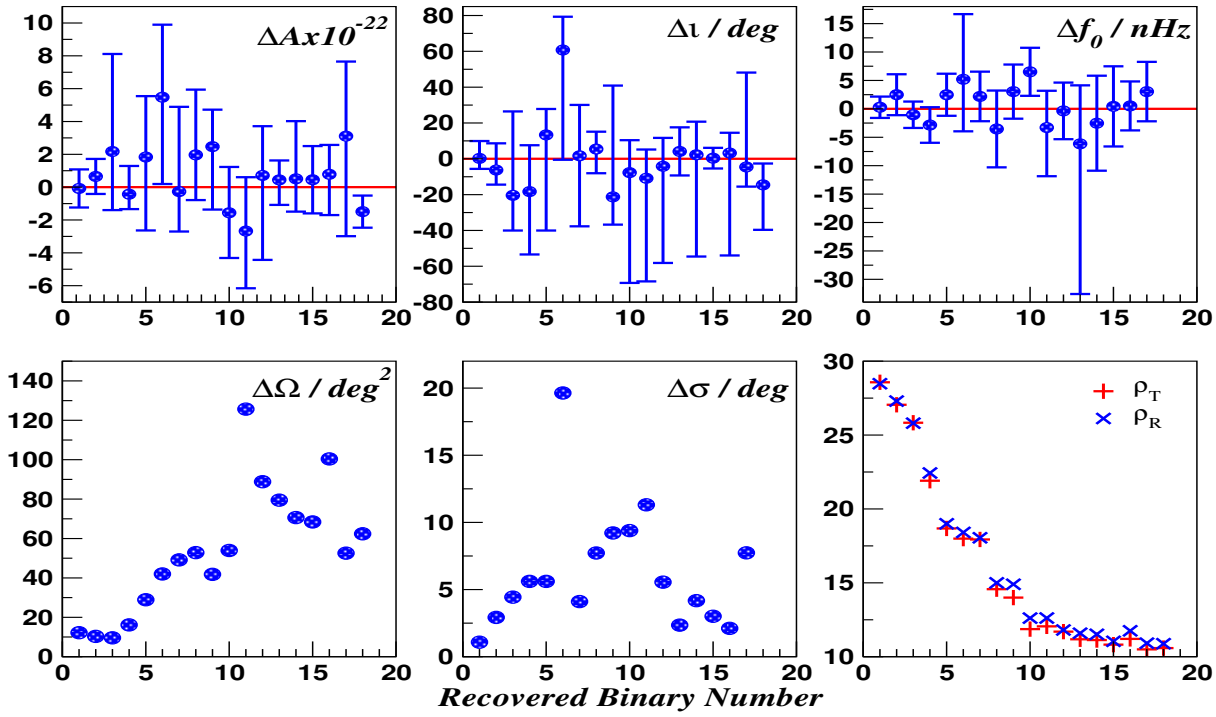

Figure 10.11: Results obtained from the recovered sources in data set 1 using Fourier domain waveform responses. In the top row, we present the 99% credible intervals and (true-subtracted) median values for amplitude (top left), inclination (top middle) and frequency (top right). In the bottom row, we present the values of the sky error boxes, the orthodromic distance and the values of the true and recovered signal-to-noise ratios.

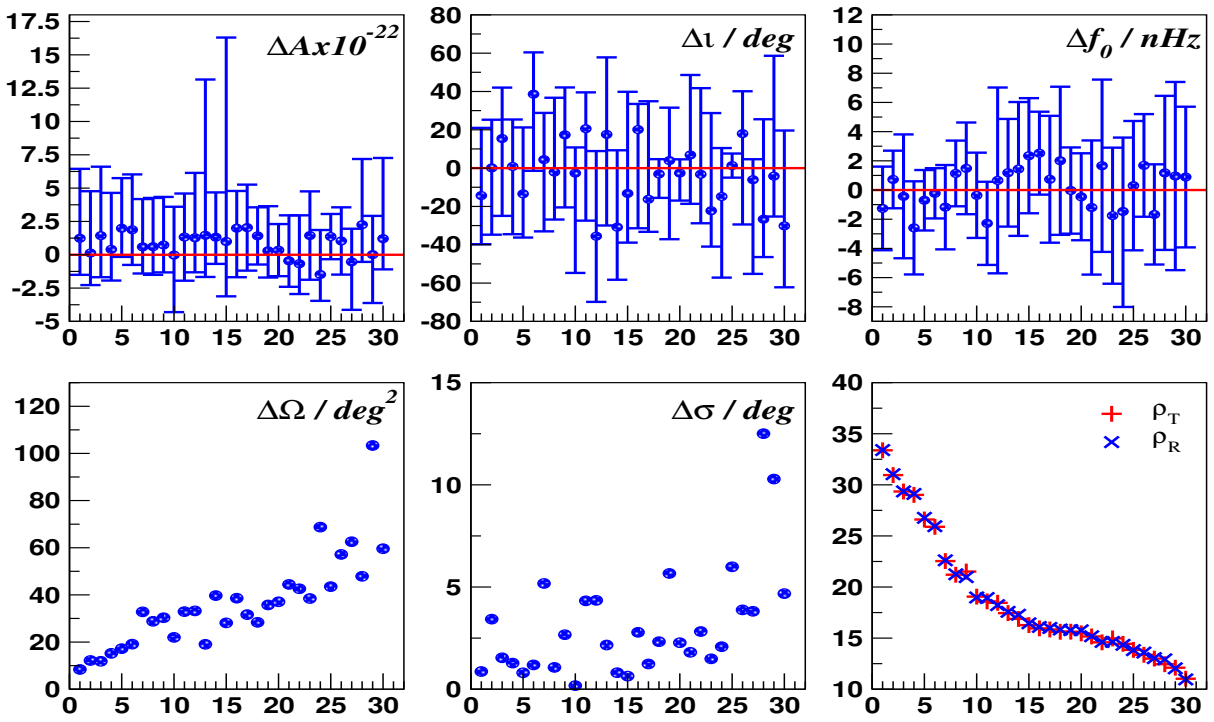

Figure 10.12: Results obtained from the recovered sources in data set 2 using time domain waveform responses. In the top row, we present the 99% credible intervals and (true-subtracted) median values for amplitude (top left), inclination (top middle) and frequency (top right). In the bottom row, we present the values of the sky error boxes, the orthodromic distance and the values of the true and recovered signal-to-noise ratios.

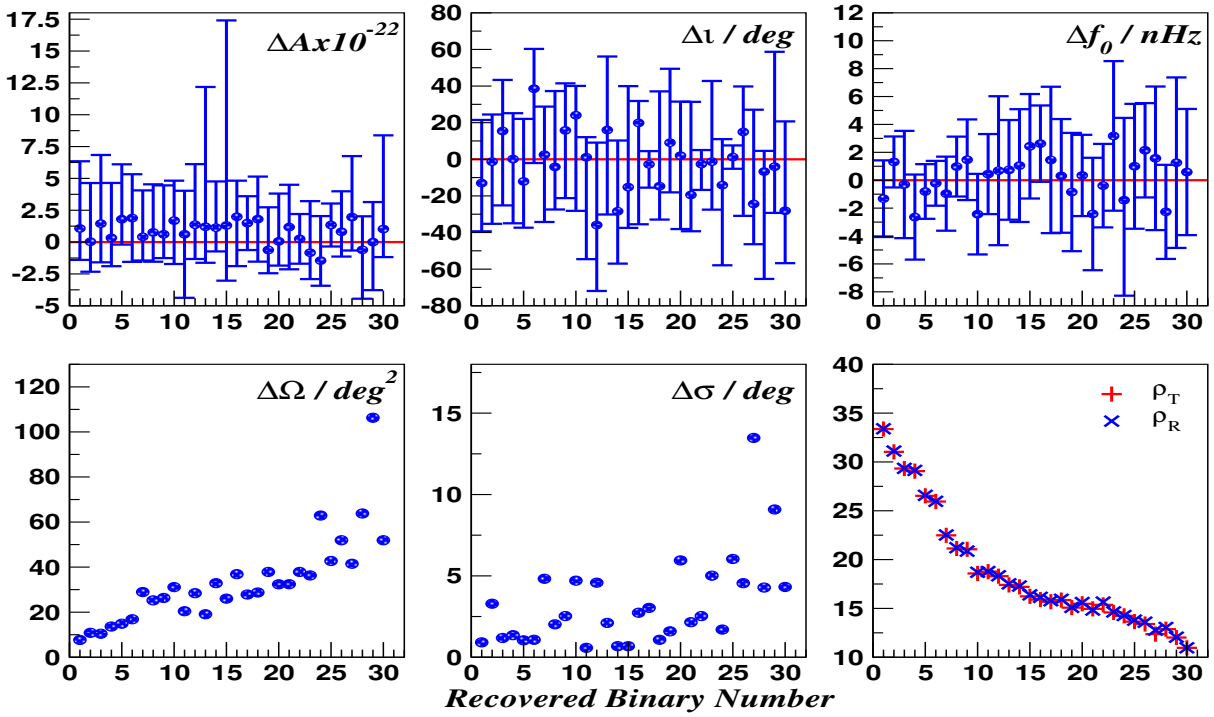

Figure 10.13: Results obtained from the recovered sources in data set 2 using Fourier domain waveform responses. In the top row, we present the 99% credible intervals and (true-subtracted) median values for amplitude (top left), inclination (top middle) and frequency (top right). In the bottom row, we present the values of the sky error boxes, the orthodromic distance and the values of the true and recovered signal-to-noise ratios.

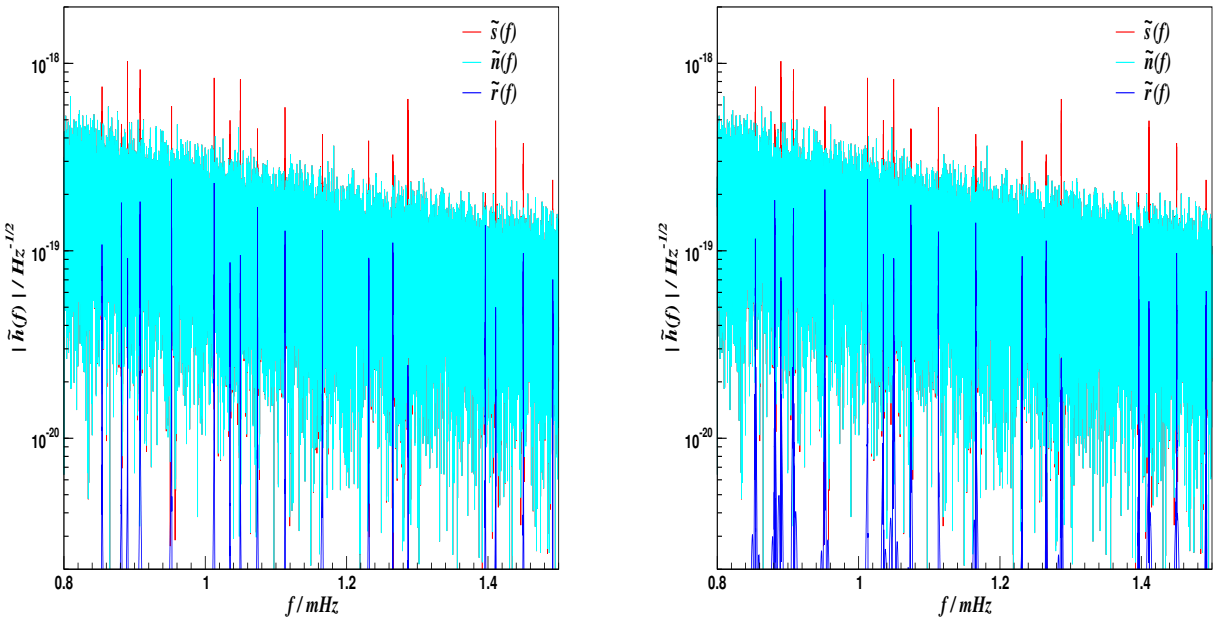

Figure 10.14: A plot of the power spectra for the injected data set, the instrumental noise and the residual for data set 1 using time (left) and Fourier (right) domain responses

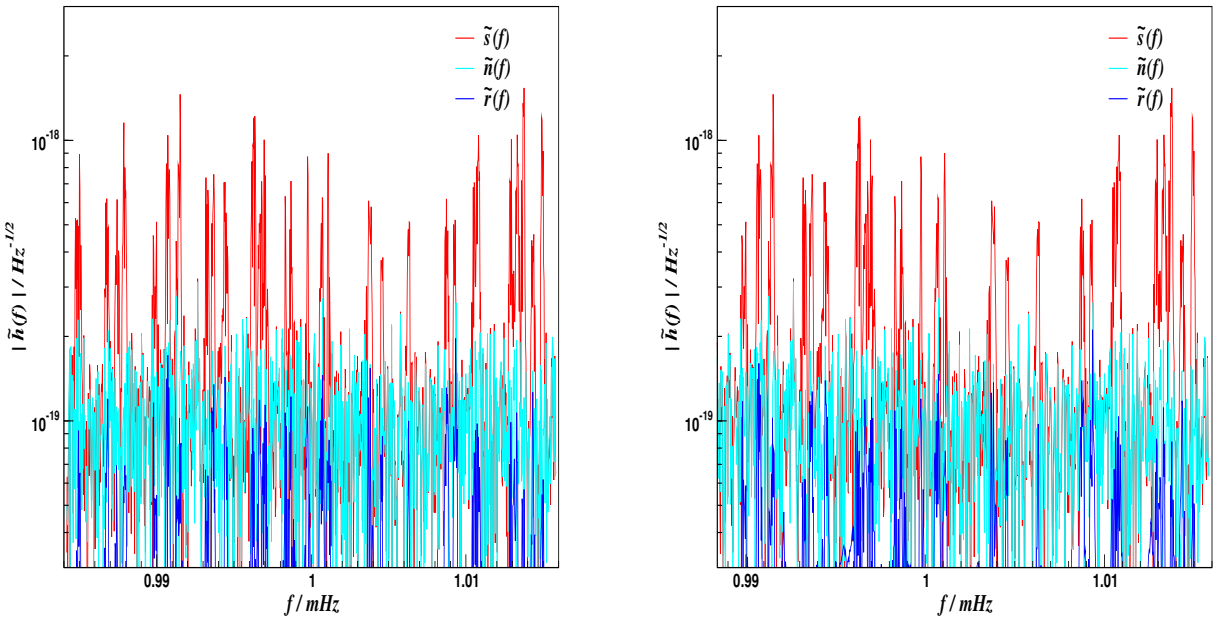

Figure 10.15: A plot of the power spectra for the injected data set, the instrumental noise and the residual for data set 2 using time (left) and Fourier (right) domain responses.

Conclusions

The work of this thesis was dedicated to the development of a number of algorithmic techniques applied to data analysis for gravitational waves. In addition, this work was split in two different projects that were respectively connected to ground-based and space-based detectors of GWs. Chronologically speaking, the first project was dedicated to the implementation of a search algorithm for monochromatic galactic binaries detected by LISA. The second subject of research was the main project of this thesis and consisted in designing a parameter estimation algorithm using a Hamiltonian Monte Carlo algorithm for binary neutron stars detected by the network of detectors advanced LIGO and advanced Virgo.

First of all, regarding the development of the parameter estimation algorithm for ground-based detectors, we successfully applied the HMC in the case of binary neutron stars coalescences modeled using Taylor F2 waveforms. We first showed how we fine-tuned the free parameters of HMC, namely the step size ϵ, the length of the trajectory l and the mass matrix $M_{\mu\nu}$, in order to have an efficient algorithm both in terms of exploration and acceptance rate. At this point, the efficiency of our algorithm was good but the computation time was prohibitive, due to the fact that the gradient of the target density needs to be evaluated numerically at each step of the trajectory. This is the reason why, the next part of this project was dedicated to tackle the problem of finding a fitting method to reduce the computation time of the algorithm. With the introduction of the fit, the algorithm was then divided into three phases. In phase I, the gradients of the target density are computed numerically and values of the points from the accepted trajectories are recorded in order to build the fit in phase II. Finally, in phase III, we use the fit derived in phase II to approximate the gradient and speed-up the computation time. We found out that the polynomial fitting method at the cubic order already developed for LISA, did not manage to produce accurate values for the fit in our case which resulted in a large decrease in acceptance rate in phase III. After an in depth-analysis, we noticed that the multi-modality feature of the posterior distribution made the cubic approximation fit fail to provide a good approximation of the gradients with respect to inclination ι, luminosity distance D_L and polarisation angle ψ.

A significant amount of this thesis work was then dedicated to the development of a working fit method for the gradients with respect to the three latter parameters. We implemented a variety of methods that all failed to produce a good fit. First, we tried to increase the order of the polynomial fit up to the quartic and quintic order but the acceptance rate remained very low in phase III. In order to have a better representation of the bimodality of the posterior distribution, we then decided to build two separate cubic approximation depending on the value of inclination. This method managed to slightly improve the acceptance rate in phase III but did not mange to reach the acceptance rate we had with numerical gradients in phase I. Finally, we tried to implement a radial basis functions method but ran into problems of memory allocation and computation time due to the inversion of large square matrices. All the details regarding these different methods are given in chapter 8 of this thesis.

The solution we found for the problem was to use a local fit method based on look-up tables. After some fine tuning analysis of the fit method, we proved that this local fit method both managed to produce a good approximation of the gradient and to keep the acceptance rate high in phase III of the algorithm. With the local fit method we developed, our HMC algorithm was able to have reasonable computation time and could then be tested in a real parameter estimation scenario. To do that, we developed beforehand a Differential Evolution Monte Carlo algorithm in order to have a way to compare the performances of our HMC algorithm. In terms of binary neutron star sources, we decided to use a set of 10 sources coming from an earlier publication of the Ligo/Virgo collaboration. We then ran both the DEMC and HMC algorithms and compared their respective performances. We found out that the HMC algorithm was able to generate statistically independent samples at a much higher rate than the DEMC algorithm. However, for some sources we ran into some troubles with the algorithm that were either connected with problems with the approximation of the gradients or problems with the scalings coming from the mass matrix. These failures motivated us to upgrade our HMC algorithm in various ways as

fully described in chapter 9. Among these upgrades, we introduced hybrid trajectories in phase III where the gradient is approximated for some of the parameters and computed numerically for the others. We also solved some issues related with the inversion of the Fisher Information matrix that produced wrong scalings for the mass matrix.

Once again, we ran the algorithm on the set of 10 binary neutron star sources and recorded various performances of the algorithm as presented in chapter 9. This time, the algorithm was capable of performing well on all sources. In addition, we found that the rate at which the algorithm produces statistically independent samples was at least 5 times faster than the DEMC and could be as high as 20 times faster in the best case scenario. So far these results could not be used to do an apples to apples comparison with the LALInference library currently used by the LIGO/Virgo collaboration. However, these are very promising since the average CPU time to produce a single statistically independent sample with our algorithm was around 1 second, while studies in the litterature reported that the time with LALInference was between 77 and 227 seconds.

A number of questions still need to be addressed in the future and will led to various future research works. First of all, we would like to implement the algorithm within the LALInference library using the data analysis tools developed by the Ligo/Virgo collaboration. We could then see how the algorithm compares with the other algorithms of LALInference. The next step would then be to use more advanced waveform models for binary neutron stars that include for instance matter effect such as tidal deformation. Then, we want to test how the HMC algorithm works with other compact binary sources that contain at least one black hole. These two latter cases both require in depth analysis since we will need to study both how the HMC tackle the extra parameters included in the waveforms along with how the fit behaves in these cases.

The second project of this thesis was dedicated to the development of a search algorithm for monochromatic galactic binaries for the future space-based observatory LISA. The number of parameters needed to describe this waveform is seven, but we can analytically maximise the log-likelihood over four parameters using the F-statistic so that the dimension of the search space reduces to three parameters: the frequency of the GW, the colatitude and longitude of the source in the sky. In chapter 10, we also derived the expression of the Fourier coefficients associated to the time domain waveform, and we have shown that the match between the time and Fourier waveforms was superior to 0.99. Similarly, we derived the Fourier coefficients for the F-statistic and showed the good agreement between the expressions obtained in the time and Fourier domains.

We then decided to construct a search algorithm based on an evolutionary algorithm called Particle Swarm Optimisation or PSO. In the algorithm framework, we consider a population of candidate solutions on the parameter space and evolve the population according to behaviors observed in Nature. In Chapter 10, we showed that when we apply this algorithm in our case, we managed to obtain good results but reached somewhat a limit of the algorithm when the size of the frequency band for the search is greater than $10^4 f_m$. In this case, we found that our algorithm presented both problems in terms of local and global exploration that could not be solved easily only using PSO. This is the reason why, we decided to introduce Differential Evolution jumps and so-called Uphill Climber steps where we accept the jumps using a greedy criterion. With the inclusion of these both algorithms, our search algorithm was capable of finding a single source on a 1 mHz frequency interval.

To further test the algorithm, we constructed two data sets designed to test various aspects in the algorithm. The first data set tested the ability of the algorithm to search for multiple sources on a large frequency band of 1 mHz, which was not done at the time this thesis work started. The data set contained 18 sources and the minimum distance in frequency between the sources was 273 f_m, indicating that we did not have any confusion in this case. The second data set tested how the algorithm behaves in the case where we have mild confusion between the sources with a minimum distance in frequency between the sources of $8f_m$. The set contained 30 sources that were spread this time on a much smaller frequency band of $10^3 f_m$.

For both data sets, the search was done sequentially meaning that we first ran our evolutionary search algorithm sequentially, extracted the value of the maximum of the log-likelihood using a superfrozen chain and finally subtracted the source using the maximum of the log-likelihood to search for the next source. In addition, we also ran a parameter estimation phase using a DEMC algorithm in order to extract the posterior distribution for all sources. We found out that our algorithm was able to properly recover all sources for the two data sets. In chapter 10, we showed that the spectrum of the residual was below the noise and that the true value for the parameters was recovered within the 99% credible interval derived from the parameter estimation runs.

A number of other investigations are planned for the future. First we would like to test how the

algorithm behaves in the case of high confusion between the sources. As the LISA mission was thought to be launched in a reduced version called eLISA with only two arm links at the time the study was done, we would like to test the algorithm using the current design of the LISA mission. This study was also restricted only to monochromatic galactic binaries with a constant frequency f_0. It would be interesting to see how our evolutionary algorithm performs when we use a model that includes the derivative of the frequency with respect to time. Similarly, we would be interested to use models where the orbit of the binary is eccentric instead of purely circular. Finally, we want to run the algorithm on a full population of galactic binaries modeled in order to see how it performs in a data analysis situation close to what will have when the mission is launched.

While the two projects conducted during this thesis and presented before do not completely overlap, there are a number of strong connections between the two studies. First of all, in both cases we studied sources of gravitational waves that were binaries formed of compact objects with matter, meaning white dwarfs and neutron stars. In the case of LISA, the compact binary is in a period of its life where the two objects orbit steadily around each other with almost constant frequency. As the two objects get close, the frequency of the gravitational waves increases and the binary moves in a frequency band of the ground-based detectors until they merge at frequency close to 1 kHz. In both projects, the work of this thesis was focused on gravitational wave data analysis whether it it related to searches (LISA) or parameter estimation (LIGO/Virgo and LISA). Secondly, the techniques applied in each case could be transferred between the two projects. As an example, it would be of great interest to develop a HMC algorithm for parameter estimation of galactic binaries with LISA and see how it compares to the regular MCMC methods such as DEMC that were applied for this project. Similarly, applying the search algorithm developed in the case of LISA for the search of sources observed by the ground-based detectors could lead to interesting results that may improve the current searches algorithm used by the LIGO/Virgo collaboration. Finally, this thesis work could also be of interest for other data analysis problems encountered in other fields of physics or science that use Bayesian analysis and need advanced algorithmic techniques to solve their problems.

Appendices

Appendix A

Differential Evolution Markov Chain Results

In this Appendix we present the results from the DEMC chains for all binary systems in our test study.

A.1 Global Analysis

These tables represent the performance diagnostics from the chains for all ten binaries.

BNS	AR/%	t_{run}/hr	τ_{zac}	L	SIS	t_{5000}/hr
1	17.44	4.67	10450	2522	396	58.97
2	7.53	4.22	13998	2747	364	57.97
3	10.29	4.08	12670	3537	283	72.09
4	17.20	3.91	5932	2441	410	47.68
5	7.28	3.49	11902	2586	387	45.09
6	15.01	2.96	10598	1357	737	20.08
7	18.23	3.49	3098	1137	879	19.85
8	9.93	2.91	10475	1931	518	28.09
9	11.46	3.30	4196	1664	601	27.45
10	12.55	3.49	11450	3117	321	54.36

Table A.1: DEMC chain diagnostics for ten BNS sources using 10^6 iterations. Column two gives the acceptance rate at the end of the chain. Column three gives the run-time for the chains in hours. Columns four to six give the lag at which the autocorrelation of the slowest mixing chain goes to zero, τ_{zac}, the integrated autocorrelation length of this chain, L, and the number of statistically independent samples, SIS, based on this chain. The final column gives the effective time needed to accumulate 5000 SISs.

BNS	ι/deg	ϕ_c/deg	ψ/deg	D_L/Mpc	$\mathcal{M}_c/M_\odot$	$\mu/M_\odot$	θ / rad	ϕ / deg	t_c / secs
1	5703	1263	8939	10450	950	1082	1029	1122	542
2	13998	1645	1330	9517	2076	1708	2334	1270	1439
3	12670	1158	1597	3984	1239	1169	847	956	1131
4	5932	431	924	3529	687	659	4215	3829	796
5	11902	1570	2105	5168	1752	1578	2693	2723	1831
6	9449	1239	650	10598	703	1248	1181	1222	754
7	2469	697	813	3098	714	700	858	772	697
8	8590	2834	1491	8318	1488	2842	10231	10475	3439
9	4196	1146	989	2874	1093	1083	883	1084	1058
10	11450	1402	778	7932	1328	1292	991	1278	1374

Table A.2: Zero auto-correlation lags for all parameter chains for each of the ten BNS sources in our study. We observe that in most cases, the slowest mixing chains are either ι or $\ln D_L$. However, for BNS 8, the slowest mixing chains are the sky angles θ and ϕ.

A.2 Autocorrelation

In this section, we plot the autocorrelation as a function of lag for binaries 2-4, and 6-10.

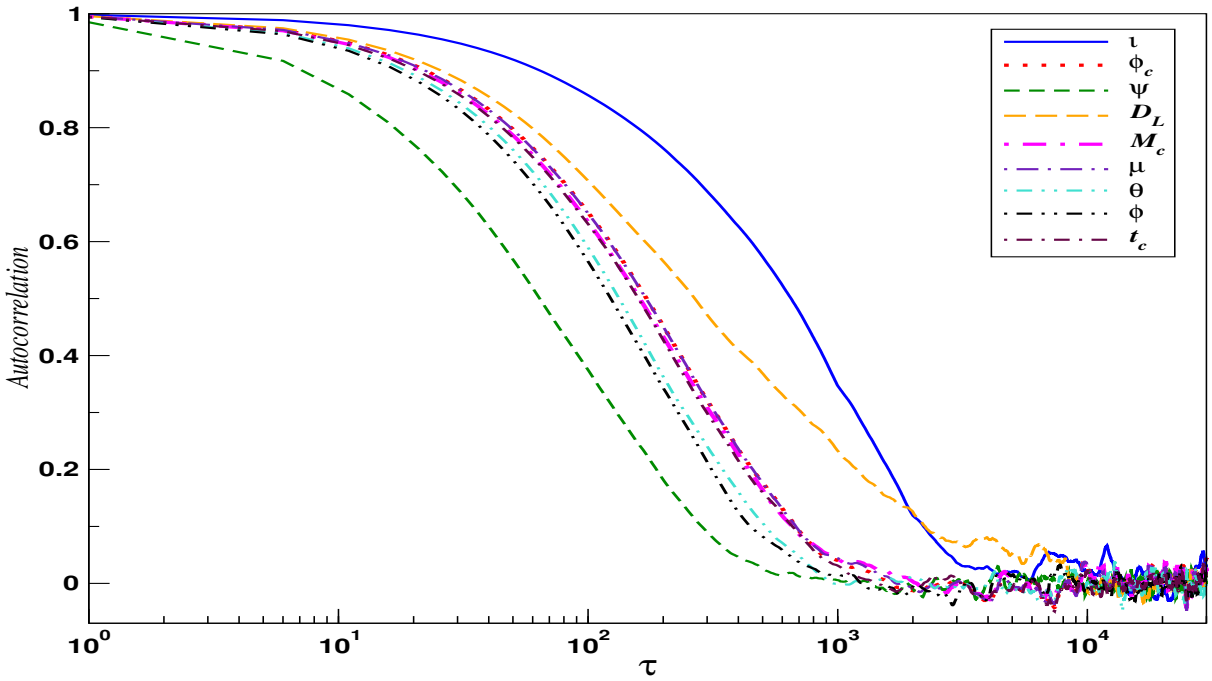

Figure A.1: Autocorrelation as a function of lag τ for BNS2 using a 10^6 iteration DEMC. The slowest mixing chain in this case is ι, which has zero autocorrelation at $\tau = 13998$.

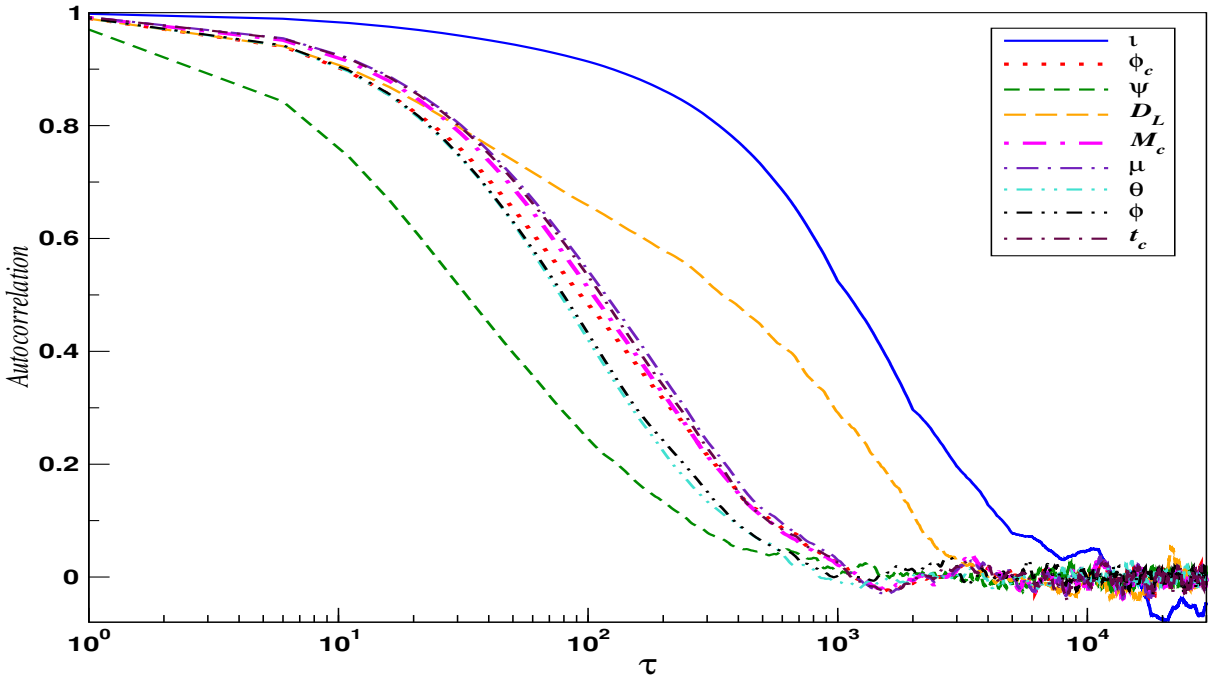

Figure A.2: Autocorrelation as a function of lag τ for BNS3 using a 10^6 iteration DEMC. The slowest mixing chain in this case is ι which has zero autocorrelation at $\tau = 12670$

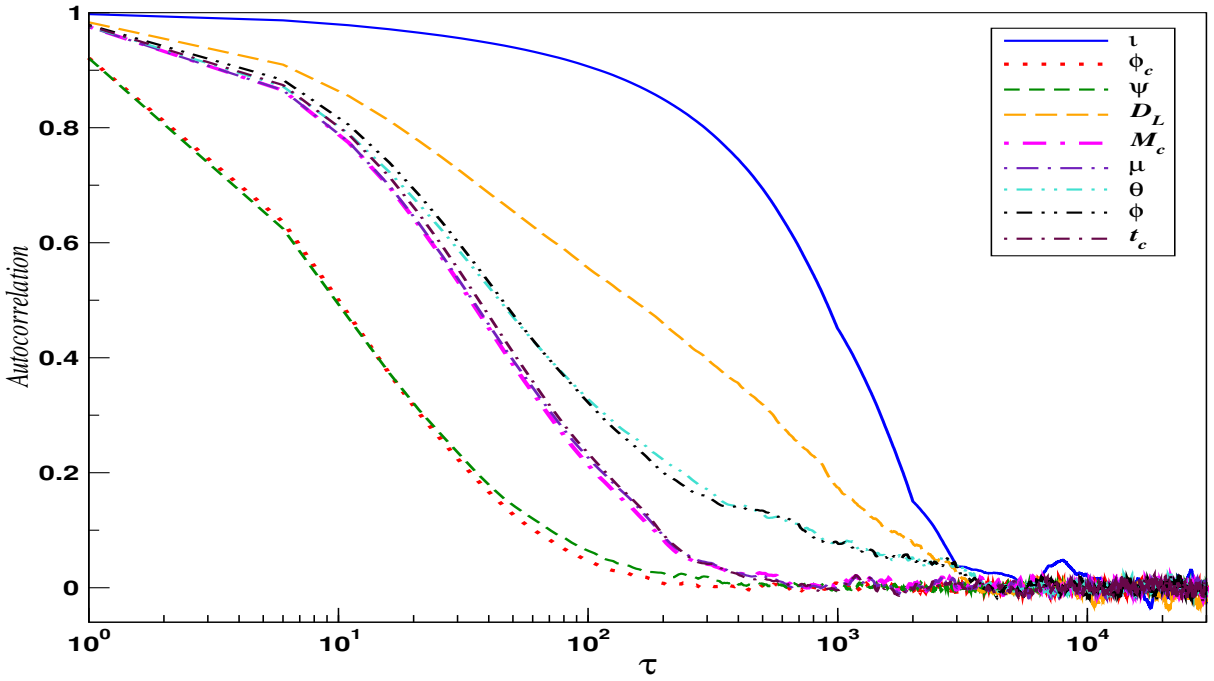

Figure A.3: Autocorrelation as a function of lag τ for BNS4 using a 10^6 iteration DEMC. The slowest mixing chain in this case is ι, which has zero autocorrelation at $\tau = 5932$.

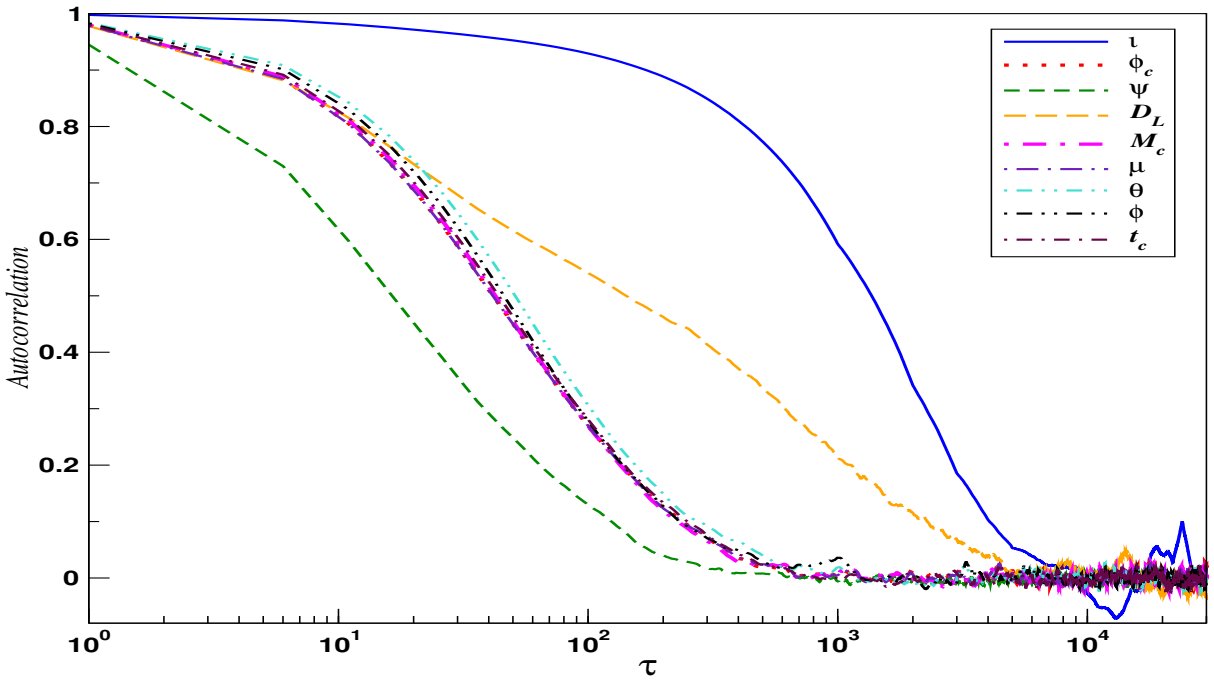

Figure A.4: Autocorrelation as a function of lag τ for BNS6 using a 10^6 iteration DEMC. The slowest mixing chain in this case is D_L which has zero autocorrelation at $\tau = 10598$

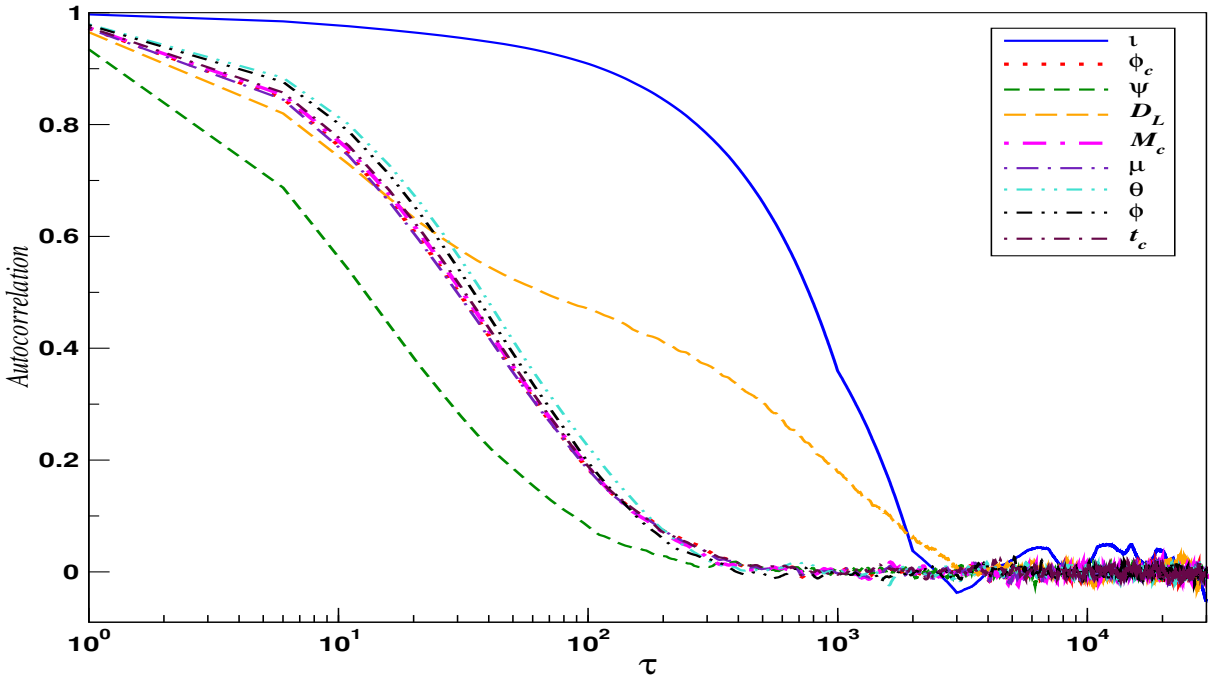

Figure A.5: Autocorrelation as a function of lag τ for BNS7 using a 10^6 iteration DEMC. The slowest mixing chain in this case is D_L, which has zero autocorrelation at $\tau = 3098$.

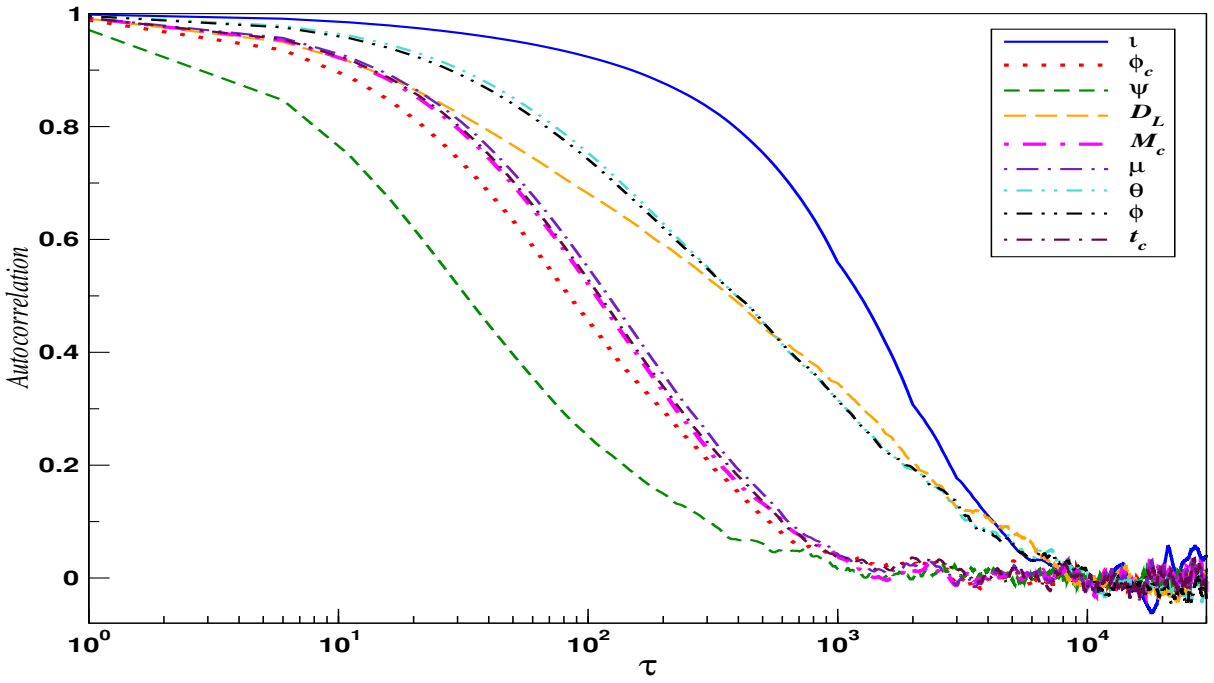

Figure A.6: Autocorrelation as a function of lag τ for BNS8 using a 10^6 iteration DEMC. The slowest mixing chain in this case is ϕ which has zero autocorrelation at $\tau = 10231$

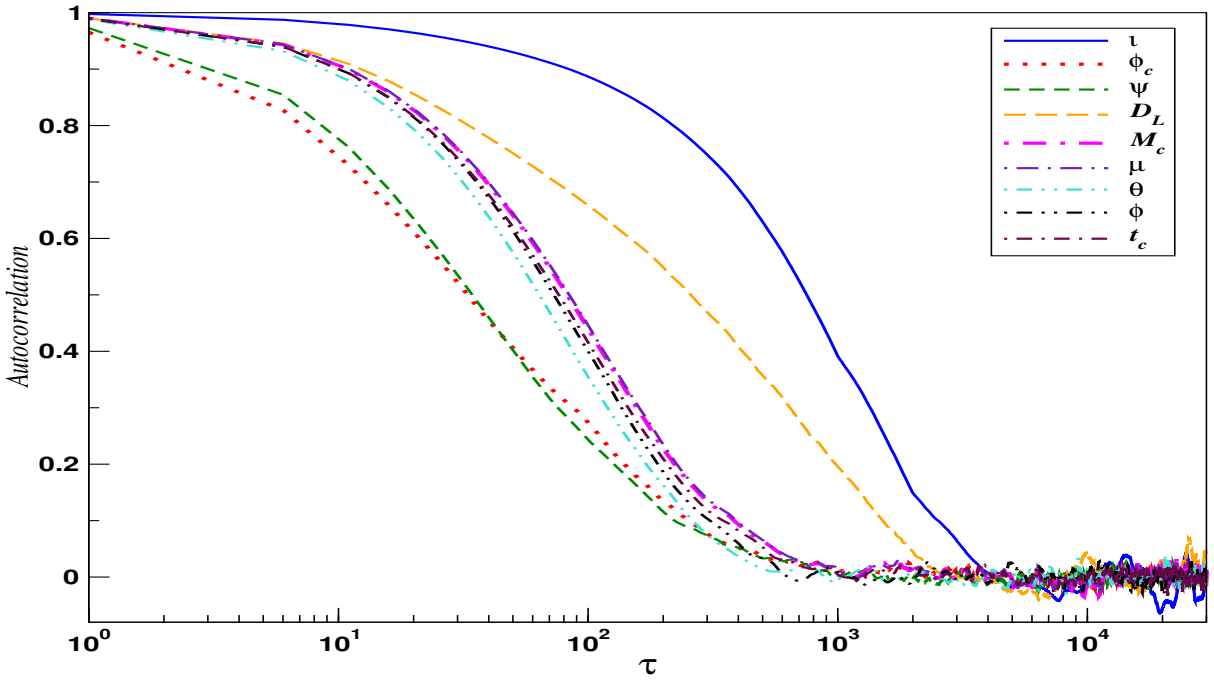

Figure A.7: Autocorrelation as a function of lag τ for BNS9 using a 10^6 iteration DEMC. The slowest mixing chain in this case is ι, which has zero autocorrelation at $\tau = 4196$.

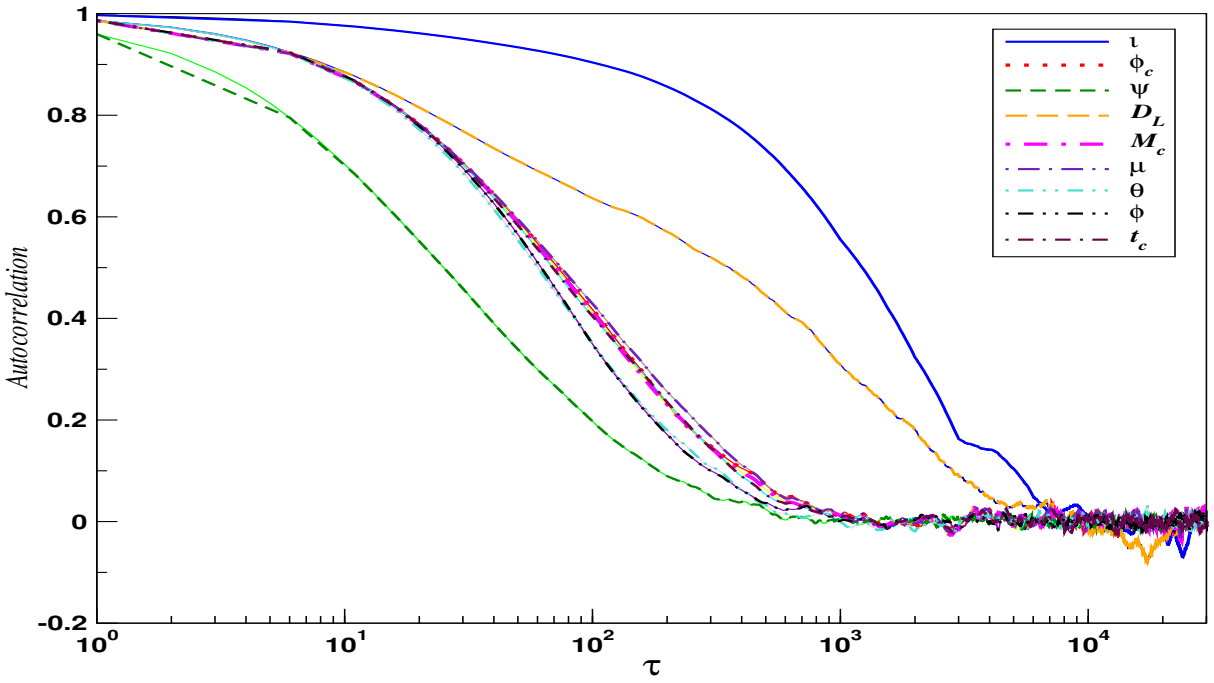

Figure A.8: Autocorrelation as a function of lag τ for BNS10 using a 10^6 iteration DEMC. The slowest mixing chain in this case is ι, which has zero autocorrelation at $\tau = 11450$

A.3 Median and credible intervals

In this section, we give the values of the medians and credible intervals inferred from a 10^6 iteration DEMC for binaries 2-4, and 6-10.

BNS	2	3	4	6
D_L/Mpc	41 $34.445^{+23.431}_{-23.431}$	84 $70.685^{+27.022}_{-41.264}$	57 $50.628^{+23.846}_{-23.846}$	46 $40.158^{+15.438}_{-15.438}$
$\mathcal{M}_c/M_\odot$	1.11743 $1.11741^{+0.00008}_{-0.00012}$	1.13486 $1.13481^{+0.00013}_{-0.00021}$	1.15865 $1.15862^{+0.00017}_{-0.00035}$	1.17959 $1.17953^{+0.00010}_{-0.00018}$
$\mu/M_\odot$	0.64132 $0.64099^{+0.00081}_{-0.00195}$	0.65134 $0.65050^{+0.00132}_{-0.00349}$	0.66412 $0.66361^{+0.00190}_{-0.00644}$	0.67749 $0.67663^{+0.00087}_{-0.00290}$
θ / rad	0.89535 $0.89580^{+0.01658}_{-0.01658}$	-0.53756 $-0.53675^{+0.01193}_{-0.01193}$	0.15708 $0.15862^{+0.09354}_{-0.06367}$	0.29147 $0.29381^{+0.09572}_{-0.06078}$
ϕ / deg	3.90779 $3.90799^{+0.01210}_{-0.01210}$	1.74358 $1.74420^{+0.01019}_{-0.01019}$	2.94786 $2.94760^{+0.02032}_{-0.01364}$	1.85179 $1.85130^{+0.01712}_{-0.01712}$
t_c / secs	29.32578 $29.32574^{+0.00016}_{-0.00022}$	28.58003 $28.57995^{+0.00027}_{-0.00044}$	27.60872 $27.60866^{+0.00041}_{-0.00076}$	26.79953 $26.79945^{+0.00020}_{-0.00032}$
$\Delta\Omega/sq.deg.$	0.053024	0.042752	0.096227	0.084043

Table A.3: True and median chain values for a subset of parameters for BNS2, BNS3, BNS4 and BNS6 using a 10^6 iteration DEMC chain. The error estimates on the median values are the 99% credible intervals. We omit values of the inclination ι as the posterior distributions are bi-modal.

BNS	7	8	9	10
D_L/Mpc	87 $82.326^{+30.836}_{-30.836}$	68 $52.333^{+19.180}_{-24.605}$	77 $60.779^{+35.647}_{-35.647}$	83 $69.059^{+37.498}_{-37.498}$
$\mathcal{M}_c/M_\odot$	1.14038 $1.14030^{+0.00015}_{-0.00029}$	1.18668 $1.18664^{+0.00015}_{-0.00025}$	1.16575 $1.16572^{+0.00020}_{-0.00030}$	1.14042 $1.14037^{+0.00011}_{-0.00022}$
$\mu/M_\odot$	0.65496 $0.65355^{+0.00143}_{-0.00510}$	0.68095 $0.68014^{+0.00145}_{-0.00414}$	0.66758 $0.66737^{+0.00222}_{-0.00504}$	0.65500 $0.65401^{+0.00099}_{-0.00371}$
θ / rad	1.02451 $1.02168^{+0.01555}_{-0.01555}$	-0.34558 $-0.35193^{+0.91517}_{-0.60311}$	1.22871 $1.22523^{+0.02086}_{-0.02086}$	-0.45204 $-0.45143^{+0.01609}_{-0.01609}$
ϕ / deg	6.02488 $6.02360^{+0.01867}_{-0.01867}$	4.84678 $4.84435^{+0.02850}_{-0.06436}$	2.11185 $2.11452^{+0.02083}_{-0.02083}$	1.35787 $1.35796^{+0.01324}_{-0.01324}$
t_c / secs	28.35058 $28.35041^{+0.00036}_{-0.00063}$	26.53244 $26.53237^{+0.00029}_{-0.00046}$	27.32872 $27.32871^{+0.00038}_{-0.00057}$	28.34895 $28.34884^{+0.00025}_{-0.00042}$
$\Delta\Omega/sq.deg.$	0.113962	0.419093	0.200214	0.043870

Table A.4: True and median chain values for a subset of parameters for BNS7, BNS8, BNS9 and BNS10 using a 10^6 iteration DEMC chain. The error estimates on the median values are the 99% credible intervals. We omit values of the inclination ι as the posterior distributions are bi-modal.

Appendix B

Hamiltonian Markov Chain Results

In this Appendix we present the results from the HMC chains for all binary systems in our test study.

B.1 Global Analysis

These tables represent the performance diagnostics from the HMC chains for all ten binaries.

BNS	AR/%	t_{run}/hr	τ_{zac}	L	SIS	t_{5000}/hr
1	85.79	27.76	63	17	58823	2.36
2	85.99	105.44	146	19	52631	10.01
3	81.16	26.15	85	13	76923	1.70
4	85.34	24.97	69	12	83333	1.50
5	83.33	22.48	51	18	55556	2.02
6	80.59	25.30	80	10	100000	1.26
7	89.50	25.46	45	9	111111	1.14
8	80.69	23.61	226	21	47619	2.48
9	76.16	28.12	164	15	66667	2.11
10	76.95	26.68	68	12	83333	1.60

Table B.1: HMC chain diagnostics for ten BNS sources using 10^6 trajectories. Column two gives the acceptance rate at the end of the chain. Column three gives the run-time for the chains in hours. Columns four to six give the lag at which the autocorrelation of the slowest mixing chain goes to zero, τ_{zac}, the integrated autocorrelation length of this chain, L, and the number of statistically independent samples, SIS, based on this chain. The final column gives the effective time needed to accumulate 5000 SISs.

BNS	ι/deg	ϕ_c/deg	ψ/deg	D_L/Mpc	$\mathcal{M}_c/M_\odot$	$\mu/M_\odot$	θ / rad	ϕ / deg	t_c / secs
1	46	28	63	21	29	29	25	18	27
2	80	140	55	109	126	140	45	47	146
3	85	47	69	50	50	49	25	24	49
4	69	27	32	39	45	44	57	57	44
5	51	50	44	46	50	50	36	35	50
6	80	47	18	29	47	47	25	24	47
7	45	26	12	16	26	26	21	22	27
8	67	226	91	130	88	89	153	156	92
9	50	54	51	164	55	55	22	30	60
10	68	46	34	46	45	45	27	37	49

Table B.2: Zero auto-correlation lags for all parameter chains for each of the ten BNS sources in our study. We observe that in most cases, the slowest mixing chains are either ι, ψ or D_L. However, for BNS2 and BNS8, the slowest mixing chains are the time at coalescence t_c and the phase at coalescence ϕ_c.

B.2 Autocorrelation

In this section, we plot the autocorrelation as a function of lag for binaries 2-4, and 6-10 using a 10^6 trajectory HMC.

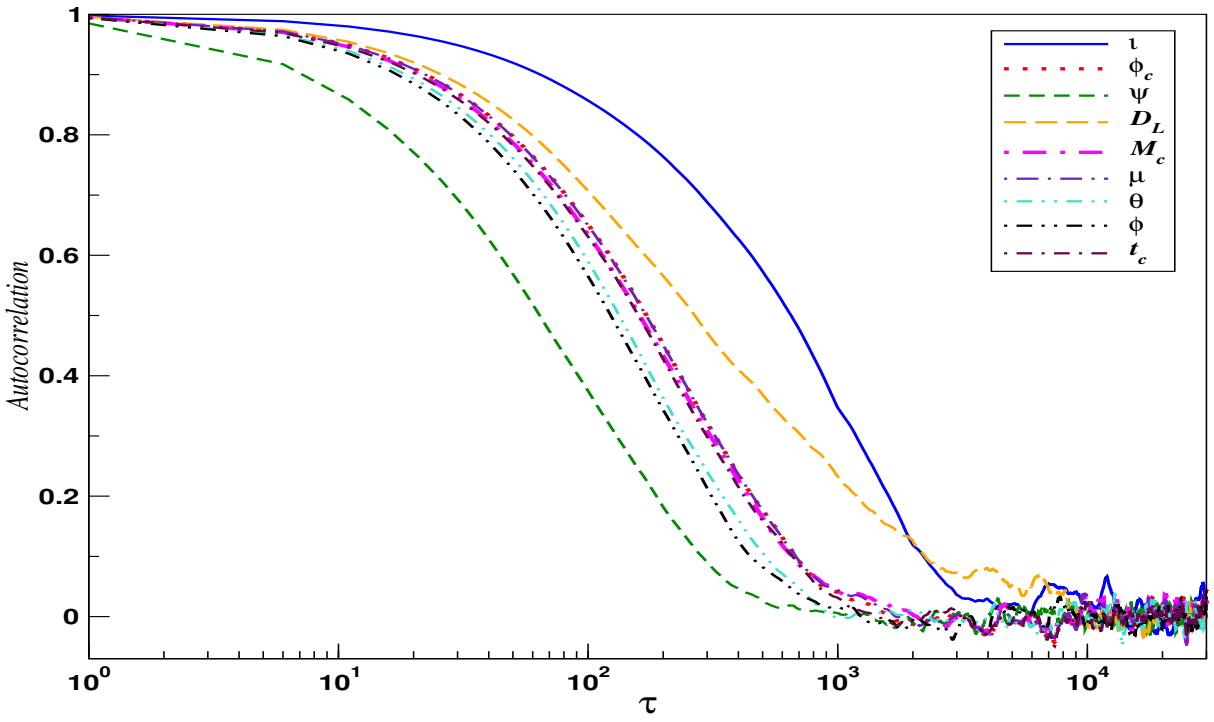

Figure B.1: Autocorrelation as a function of lag τ for BNS2 using a 10^6 trajectory HMC. The slowest mixing chain in this case is t_c, which has zero autocorrelation at $\tau = 146$.

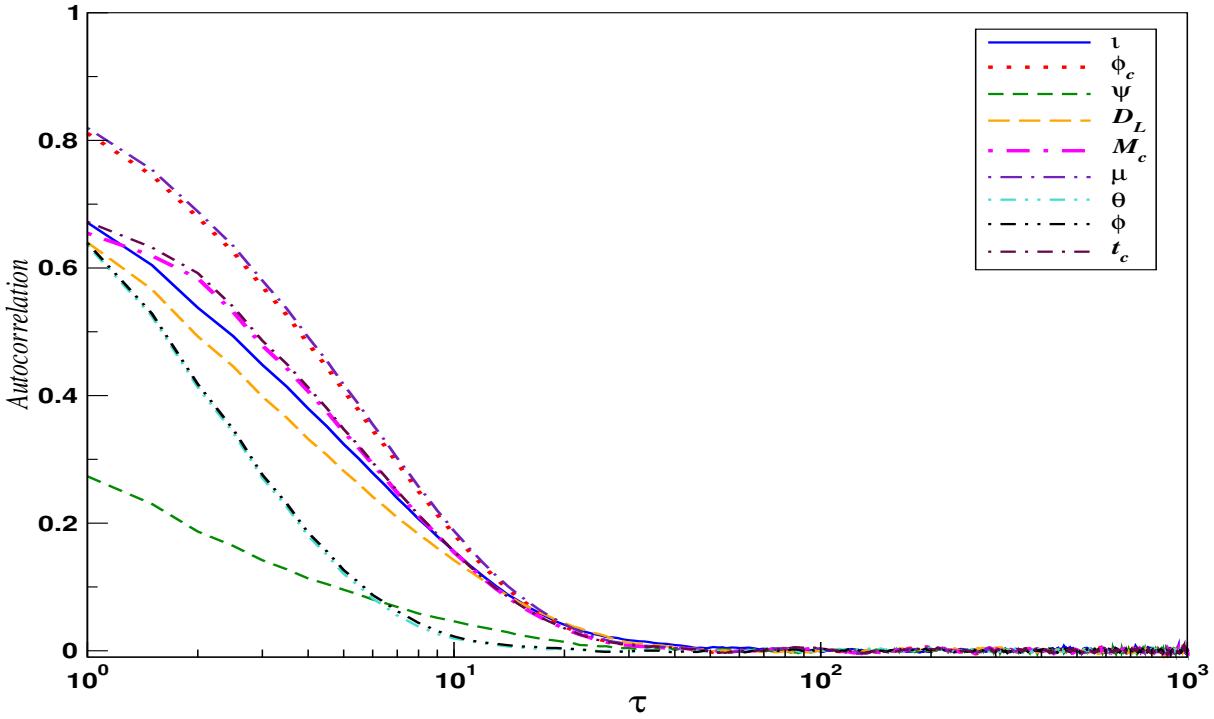

Figure B.2: Autocorrelation as a function of lag τ for BNS3 using a 10^6 trajectory HMC. The slowest mixing chain in this case is ι which has zero autocorrelation at $\tau = 85$

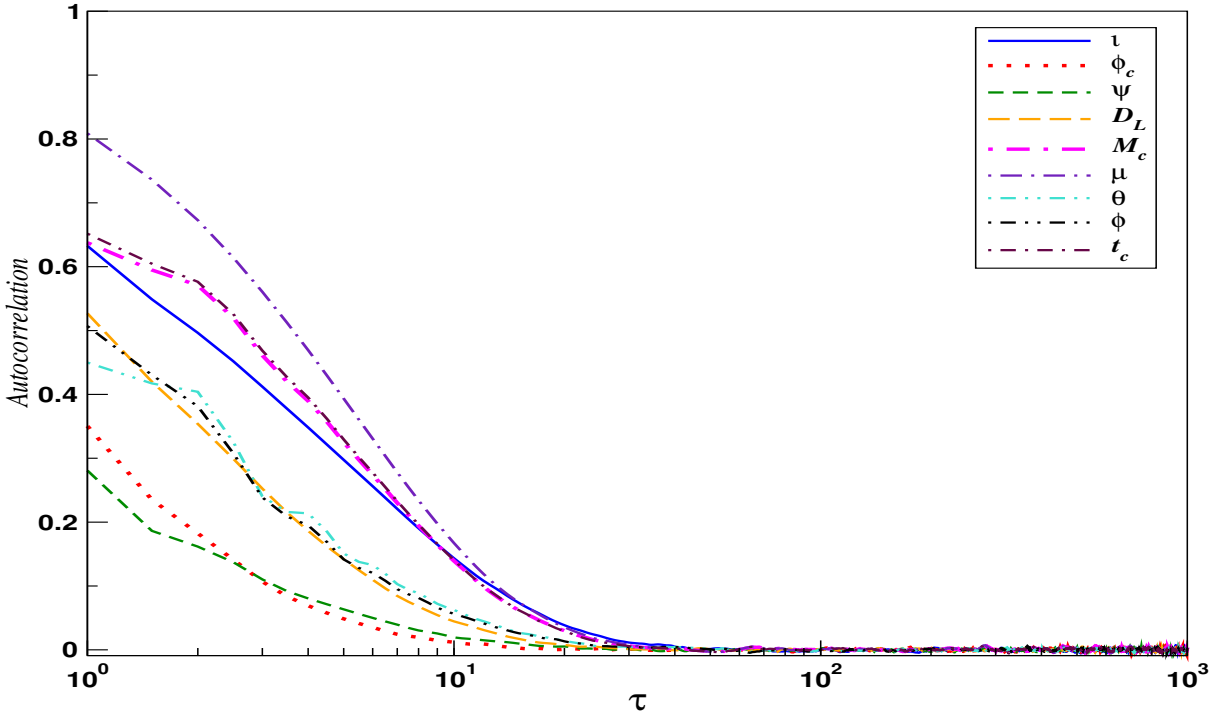

Figure B.3: Autocorrelation as a function of lag τ for BNS4 using a 10^6 trajectory HMC. The slowest mixing chain in this case is ι, which has zero autocorrelation at $\tau = 69$.

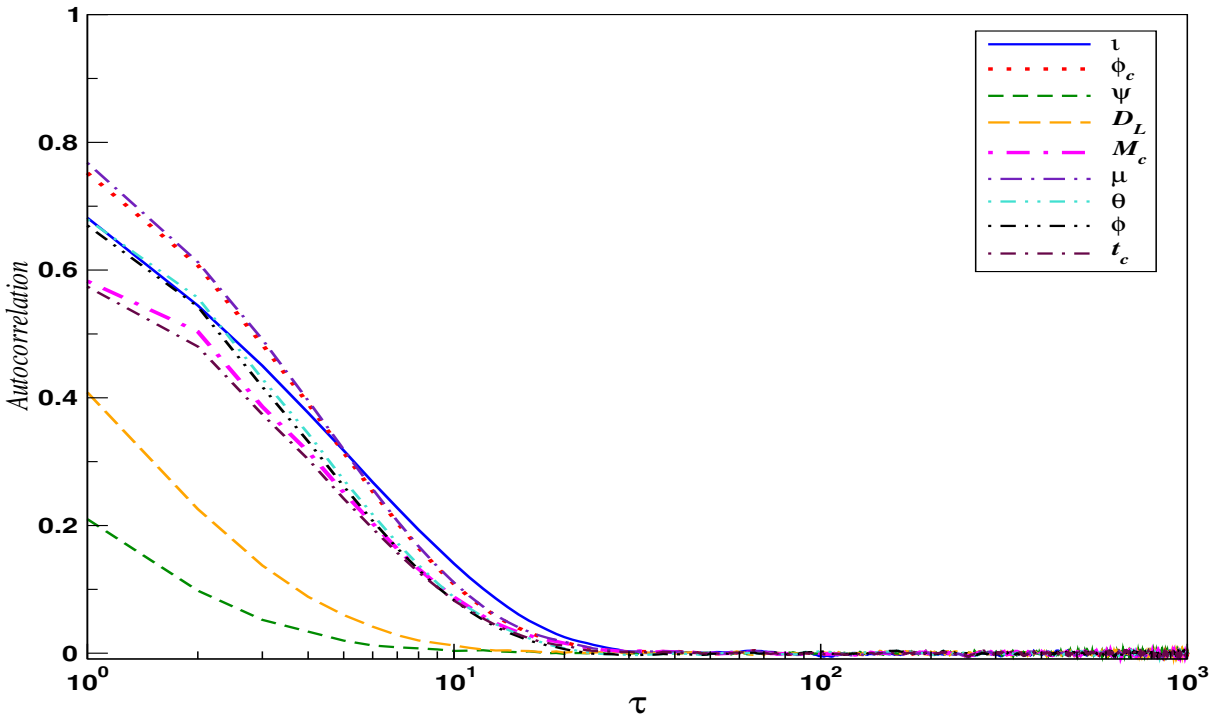

Figure B.4: Autocorrelation as a function of lag τ for BNS6 using a 10^6 trajectory HMC. The slowest mixing chain in this case is ι which has zero autocorrelation at $\tau = 80$

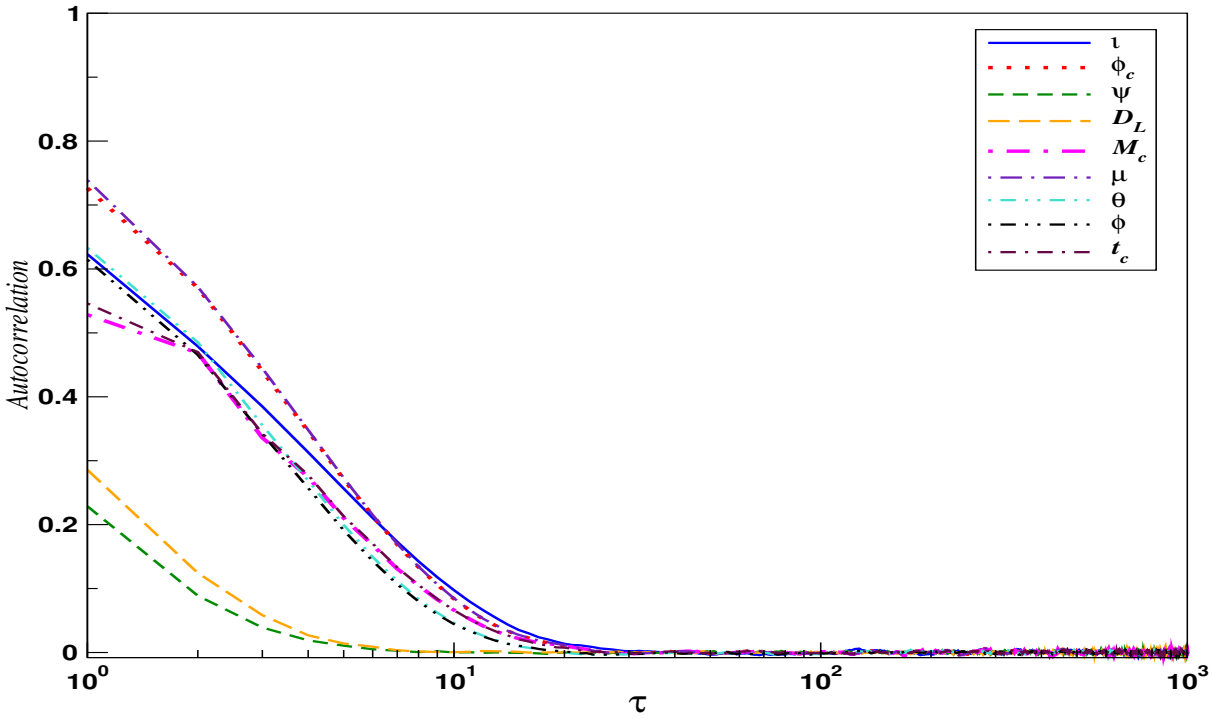

Figure B.5: Autocorrelation as a function of lag τ for BNS7 using a 10^6 trajectory HMC. The slowest mixing chain in this case is ι, which has zero autocorrelation at $\tau = 45$.

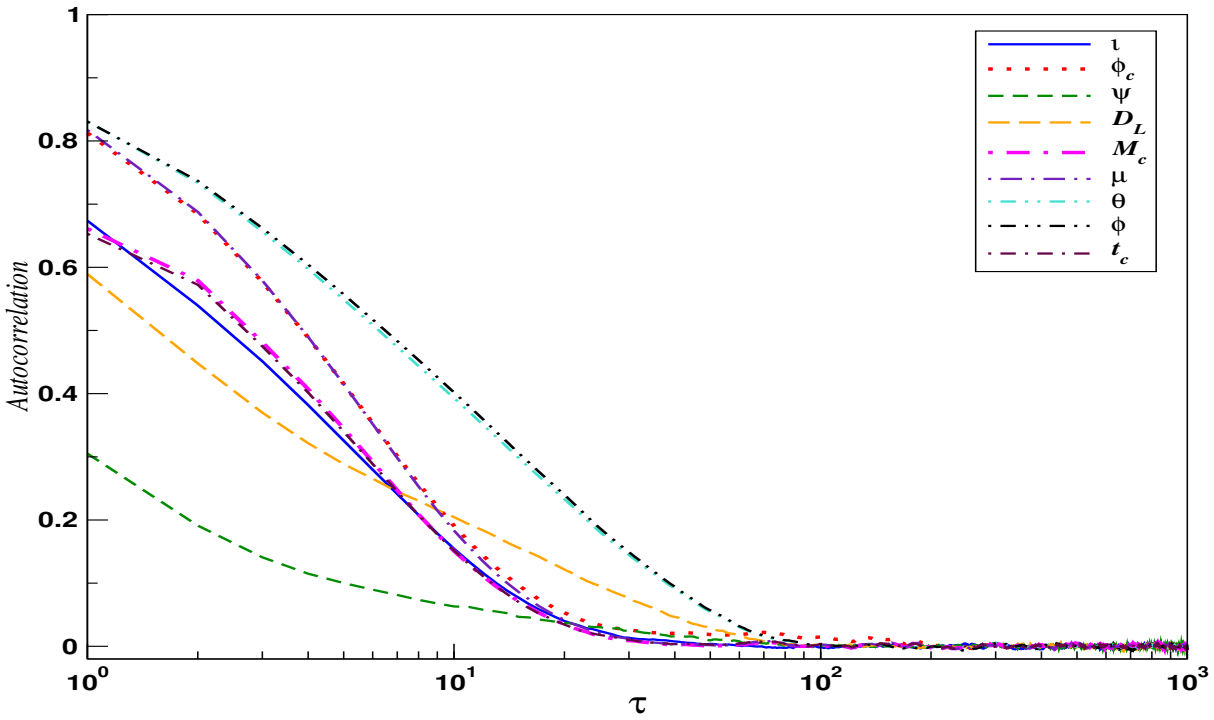

Figure B.6: Autocorrelation as a function of lag τ for BNS8 using a 10^6 trajectory HMC. The slowest mixing chain in this case is ϕ_c which has zero autocorrelation at $\tau = 226$

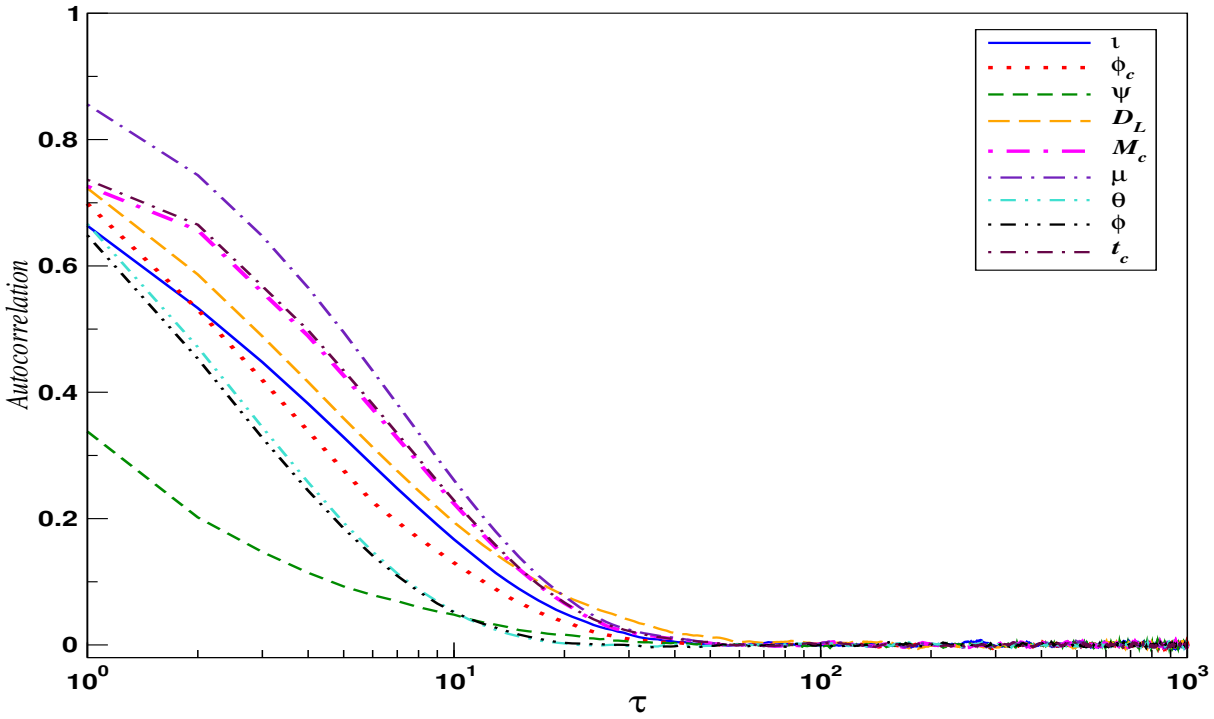

Figure B.7: Autocorrelation as a function of lag τ for BNS9 using a 10^6 trajectory HMC. The slowest mixing chain in this case is D_L, which has zero autocorrelation at $\tau = 164$.

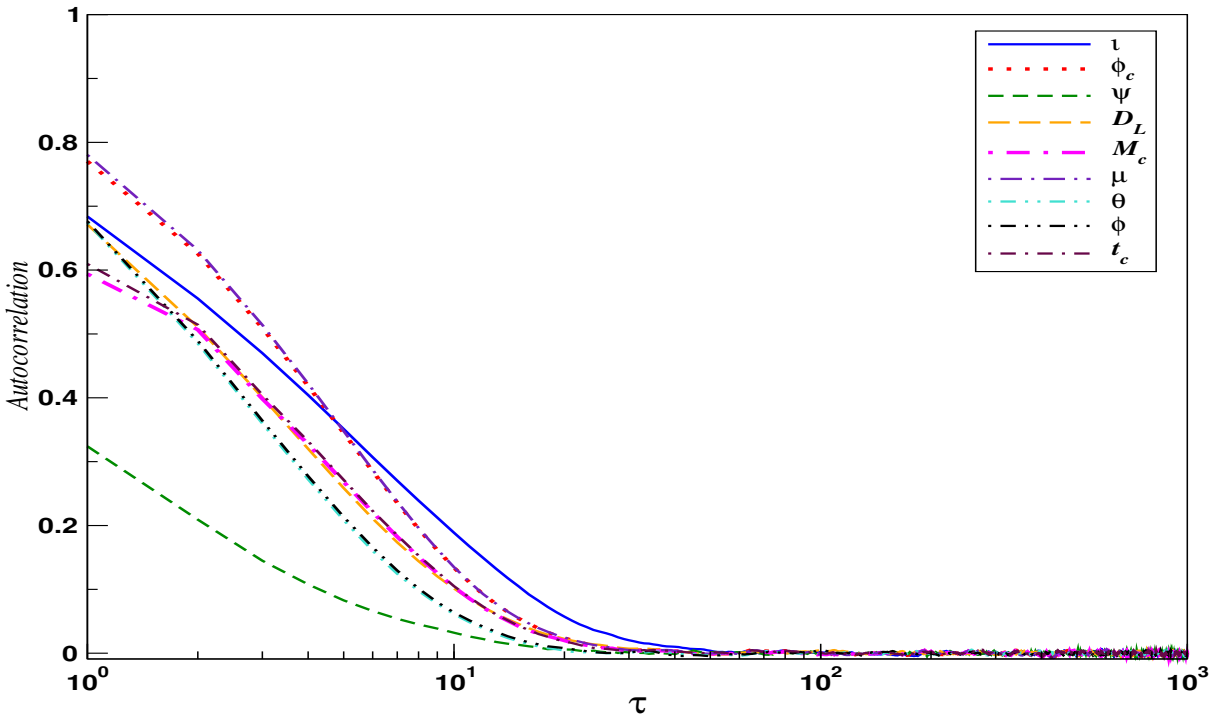

Figure B.8: Autocorrelation as a function of lag τ for BNS10 using a 10^6 trajectory HMC. The slowest mixing chain in this case is ι, which has zero autocorrelation at $\tau = 68$

B.3 Median and credible intervals

In this section, we give the values of the medians and credible intervals inferred from a 10^6 trajectory HMC for binaries 2-4, and 6-10.

BNS	2	3	4	6
D_L/Mpc	41	84 $72.407^{+25.550}_{-44.479}$	57 $52.528^{+20.000}_{-28.325}$	46 $41.446^{+15.231}_{-15.231}$
$\mathcal{M}_c/M_\odot$	1.11743	1.13486 $1.13482^{+0.00012}_{-0.00024}$	1.15865 $1.15863^{+0.00018}_{-0.00038}$	1.17959 $1.17954^{+0.00010}_{-0.00022}$
$\mu/M_\odot$	0.64132	0.65134 $0.65068^{+0.00115}_{-0.00380}$	0.66412 $0.66382^{+0.00170}_{-0.00659}$	0.67749 $0.67675^{+0.00075}_{-0.00321}$
θ / rad	0.89535	-0.53756 $-0.53674^{+0.01168}_{-0.01168}$	0.15708 $0.15797^{+0.09891}_{-0.07640}$	0.29147 $0.29350^{+0.09518}_{-0.06690}$
ϕ / deg	3.90779	1.74358 $1.74421^{+0.01020}_{-0.01020}$	2.94786 $2.94750^{+0.02382}_{-0.01433}$	1.85179 $1.85134^{+0.01676}_{-0.01676}$
t_c / secs	29.32578	28.58003 $28.57997^{+0.00029}_{-0.00046}$	27.60872 $27.60868^{+0.00040}_{-0.00079}$	26.79953 $26.79946^{+0.00021}_{-0.00036}$
$\Delta\Omega/sq.deg.$		0.041013	0.089284	0.080206

Table B.3: True and median chain values for a subset of parameters for BNS2, BNS3, BNS4 and BNS6 using a 10^6 trajectory HMC. The error estimates on the median values are the 99% credible intervals. We omit values of the inclination ι as the posterior distributions are bi-modal.

BNS	7	8	9	10
D_L/Mpc	87 $85.294^{+31.371}_{-31.371}$	68 $53.279^{+18.992}_{-25.772}$	77 $61.269^{+36.463}_{-36.463}$	83 $73.462^{+39.300}_{-39.300}$
$\mathcal{M}_c/M_\odot$	1.14038 $1.14031^{+0.00015}_{-0.00034}$	1.18668 $1.18664^{+0.00015}_{-0.00026}$	1.16575 $1.16573^{+0.00019}_{-0.00037}$	1.14042 $1.14037^{+0.00011}_{-0.00023}$
$\mu/M_\odot$	0.65496 $0.65376^{+0.00122}_{-0.00598}$	0.68095 $0.68029^{+0.00131}_{-0.00401}$	0.66758 $0.66751^{+0.00208}_{-0.00617}$	0.65500 $0.65416^{+0.00085}_{-0.00366}$
θ / rad	1.02451 $1.02171^{+0.01566}_{-0.01566}$	-0.34558 $-0.35107^{+0.92122}_{-0.59792}$	1.22871 $1.22504^{+0.02087}_{-0.02087}$	-0.45204 $-0.45160^{+0.01603}_{-0.01603}$
ϕ / deg	6.02488 $6.02358^{+0.01871}_{-0.01871}$	4.84678 $4.84470^{+0.03073}_{-0.06734}$	2.11185 $2.11453^{+0.02059}_{-0.02059}$	1.35787 $1.35803^{+0.01336}_{-0.01336}$
t_c / secs	28.35058 $28.35043^{+0.00039}_{-0.00074}$	26.53244 $26.53238^{+0.00031}_{-0.00048}$	27.32872 $27.32872^{+0.00040}_{-0.00070}$	28.34895 $28.34885^{+0.00027}_{-0.00045}$
$\Delta\Omega/sq.deg.$	0.113939	0.392069	0.203070	0.042152

Table B.4: True and median chain values for a subset of parameters for BNS7, BNS8, BNS9 and BNS10 using a 10^6 iteration HMC. The error estimates on the median values are the 99% credible intervals. We omit values of the inclination ι as the posterior distributions are bi-modal.

Appendix C

Posterior distribution DEMC and HMC

In this section, we plot the posterior distributions for binaries 2-4, and 6-10 using a 10^6 iteration DEMC (red) and a 10^6 trajectory HMC (blue). The true values are represented by the orange dashed lines.

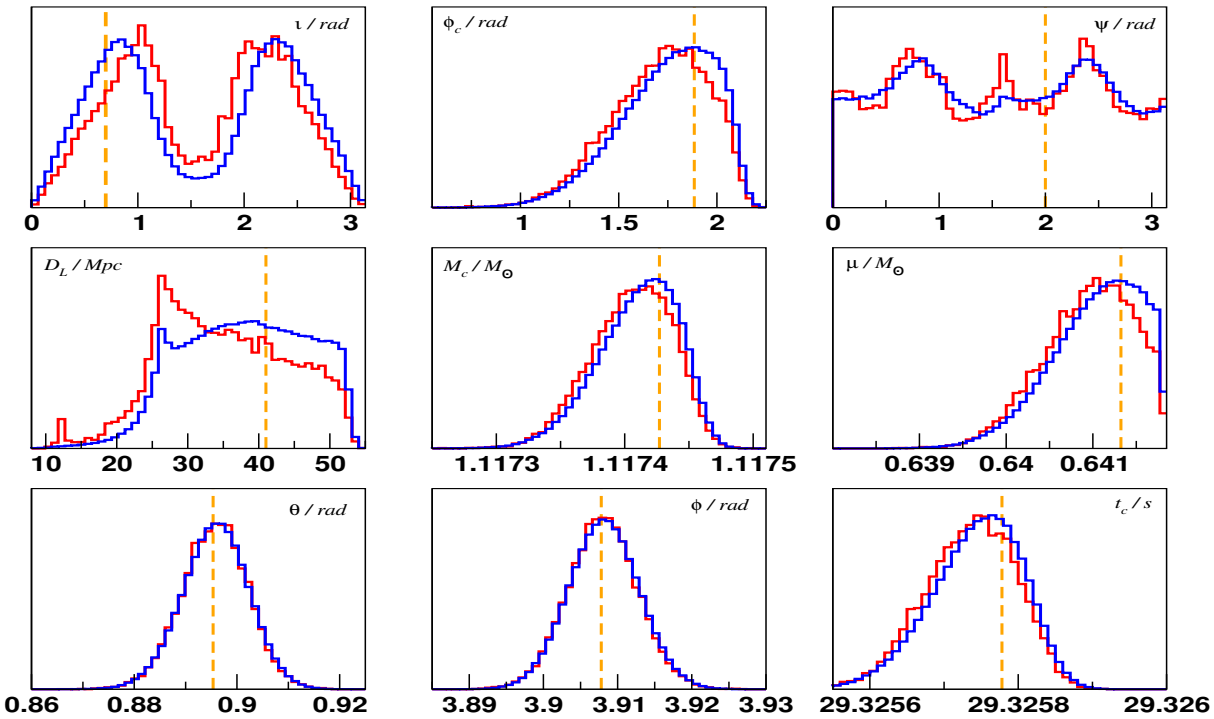

Figure C.1: Marginalised posterior distribution of the nine parameters for BNS2 using a 10^6 iteration DEMC (red) and 10^6 trajectory HMC (blue).

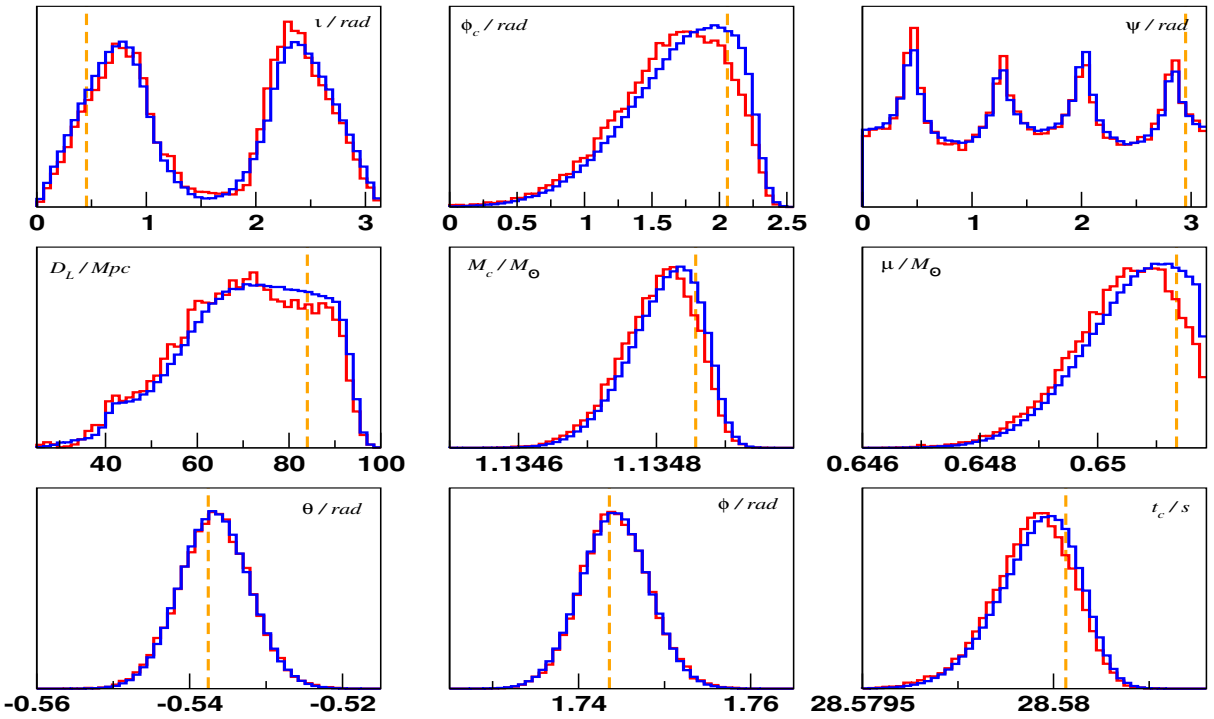

Figure C.2: Marginalised posterior distribution of the nine parameters for BNS3 using a 10^6 iteration DEMC (red) and 10^6 trajectory HMC (blue).

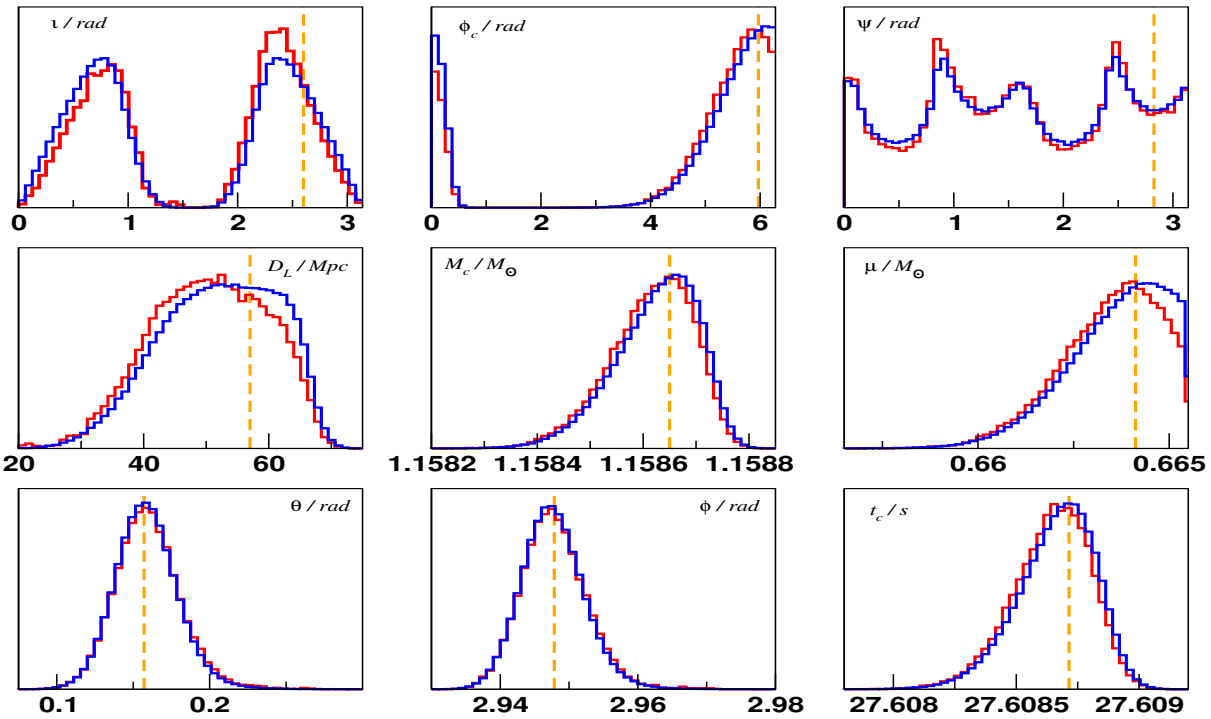

Figure C.3: Marginalised posterior distribution of the nine parameters for BNS4 using a 10^6 iteration DEMC (red) and 10^6 trajectory HMC (blue).

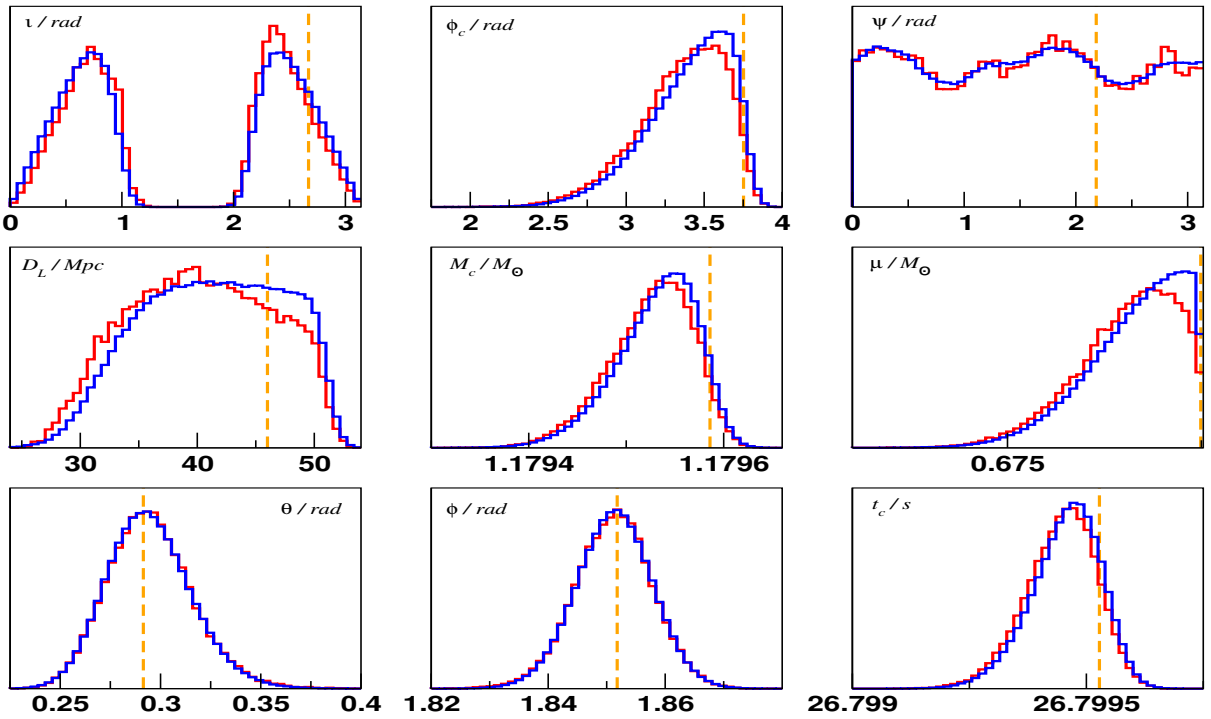

Figure C.4: Marginalised posterior distribution of the nine parameters for BNS6 using a 10^6 iteration DEMC (red) and 10^6 trajectory HMC (blue).

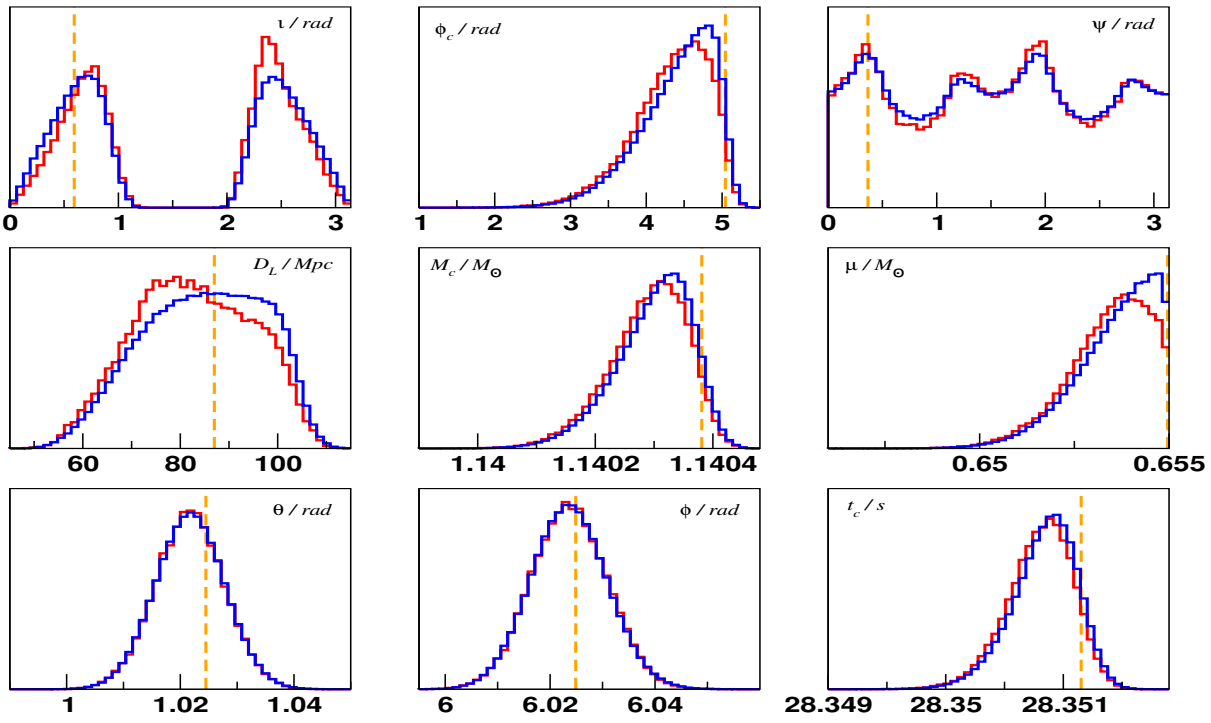

Figure C.5: Marginalised posterior distribution of the nine parameters for BNS7 using a 10^6 iteration DEMC (red) and 10^6 trajectory HMC (blue).

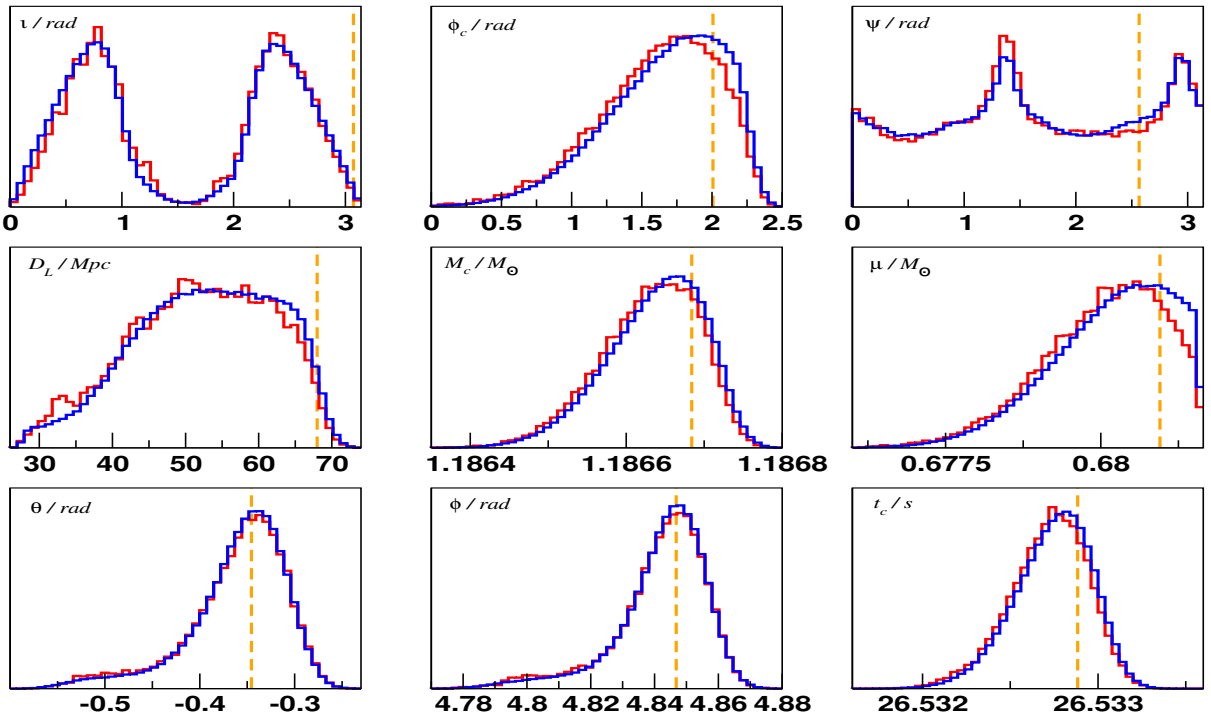

Figure C.6: Marginalised posterior distribution of the nine parameters for BNS8 using a 10^6 iteration DEMC (red) and 10^6 trajectory HMC (blue).

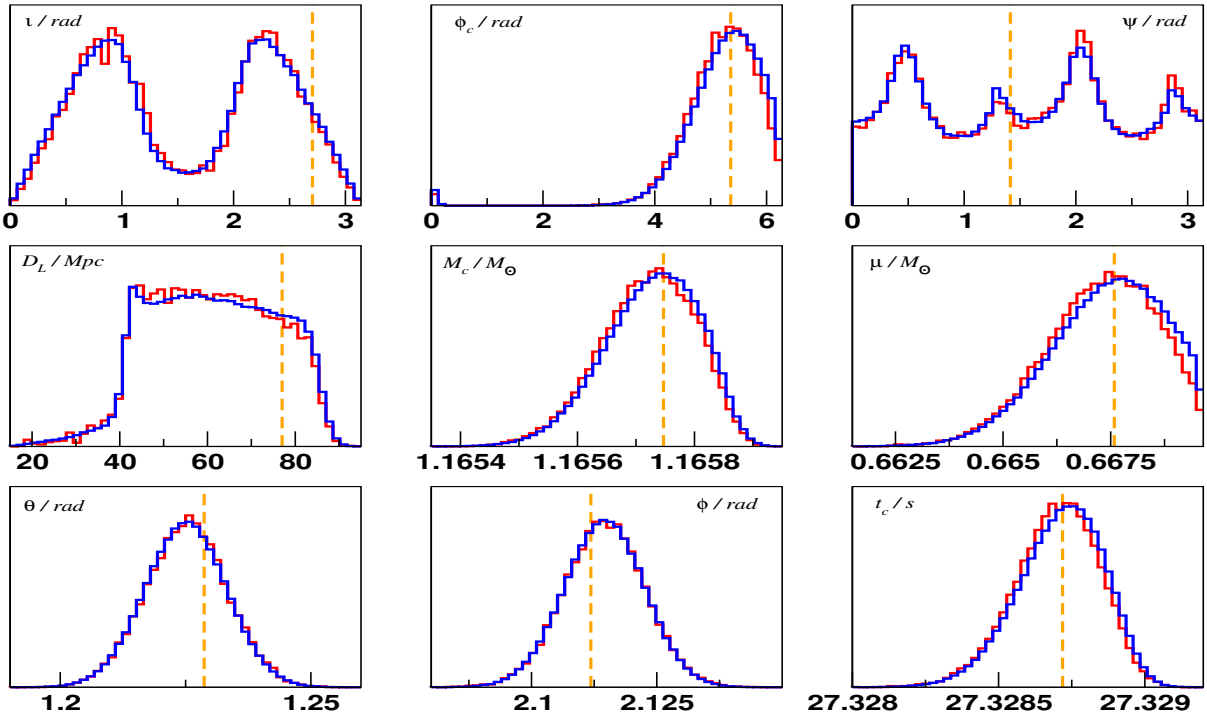

Figure C.7: Marginalised posterior distribution of the nine parameters for BNS9 using a 10^6 iteration DEMC (red) and 10^6 trajectory HMC (blue).

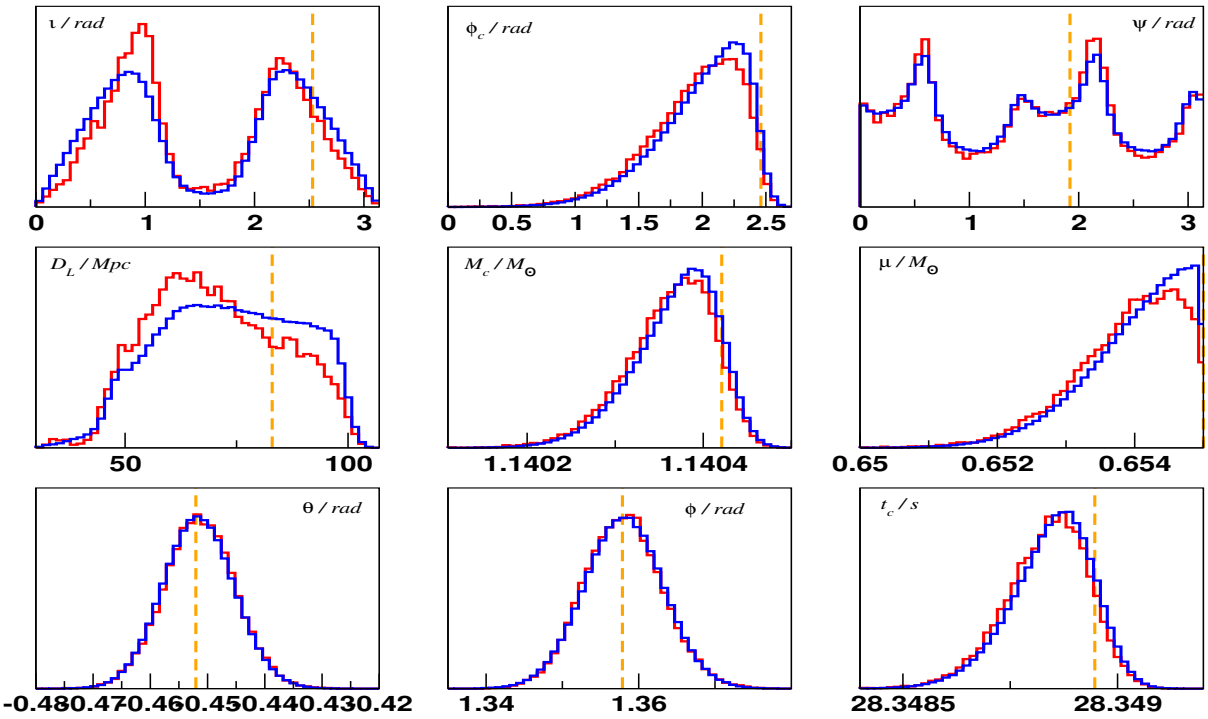

Figure C.8: Marginalised posterior distribution of the nine parameters for BNS10 using a 10^6 iteration DEMC (red) and 10^6 trajectory HMC (blue).

Appendix D

Tables of recovered values for GB search with eLISA

Here we present the injected and recovered values from the search pipeline developed for the identification of galactic binaries with eLISA for both data sets. For each each binary, the top row represents the injected values, and the bottom row the recovered values.

Binary	ι (rad)	φ_0 / rad	A ($\times 10^{-22}$)	ψ / rad	f_0 / mHz	θ / rad	ϕ / rad	ρ
1	1.921	5.401	8.592	2.272	1.2868381	1.607	0.861	28.574
	1.925	5.425	8.513	2.278	1.2868384	1.636	0.860	28.477
2	2.037	0.407	6.234	2.130	1.4108256	2.109	1.005	27.049
	1.928	0.073	6.887	2.095	1.4108281	2.014	1.011	27.300
3	2.614	5.399	7.001	1.997	1.0496413	0.657	4.379	25.834
	2.258	3.236	9.175	1.611	1.0496403	0.670	4.302	25.807
4	0.973	3.079	3.562	2.159	1.4499807	1.822	2.603	21.913
	0.653	3.163	3.134	1.562	1.4499779	1.919	2.549	22.433
5	0.794	1.146	7.543	2.447	0.9077304	1.899	3.754	18.679
	1.027	0.661	9.381	2.498	0.9077328	2.005	3.712	18.998
6	0.169	1.721	4.138	2.258	1.0125854	2.527	3.814	17.991
	1.229	2.567	9.616	1.386	1.0125906	2.546	3.472	18.413
7	0.707	5.536	7.505	0.378	0.8900178	2.181	2.625	17.933
	0.736	3.146	7.238	1.572	0.8900200	2.120	2.575	18.049
8	1.374	5.872	6.656	1.307	1.1128036	2.440	2.393	14.569
	1.469	3.157	8.624	2.892	1.1128001	2.445	2.528	15.000
9	2.265	5.416	4.265	2.420	1.0739092	1.861	2.984	14.004
	1.894	4.639	6.736	2.248	1.0739122	1.966	3.106	14.902
10	1.334	5.966	6.307	0.062	1.1657974	0.910	4.236	11.869
	1.200	5.440	4.746	0.123	1.1658040	0.681	4.120	12.613
11	1.292	6.011	8.820	2.480	0.9523456	1.860	5.740	12.057
	1.102	6.610	6.147	2.324	0.9523423	2.015	5.588	12.619
12	1.505	1.234	9.139	2.217	0.8539045	1.698	3.036	11.698
	1.433	4.555	9.850	0.661	0.8539042	1.762	3.109	11.809
13	1.310	1.508	2.749	0.964	1.4916823	1.821	3.124	11.179
	1.382	2.009	3.192	1.078	1.4916762	1.835	3.163	11.591
14	1.037	0.901	2.971	0.590	1.2312381	0.796	0.794	11.129
	1.076	3.150	3.480	1.598	1.2312356	0.785	0.722	11.516
15	1.633	2.941	7.499	1.073	1.0349592	2.173	4.157	10.819
	1.639	6.147	7.950	2.585	1.0349597	2.198	4.107	11.042
16	1.234	1.502	3.227	2.230	1.2654466	1.224	0.927	11.207
	1.290	1.607	4.010	2.000	1.2654471	1.264	0.955	11.755
17	1.885	0.241	6.627	1.720	0.8813608	2.248	6.083	10.498
	1.805	2.752	9.734	0.053	0.8813639	2.244	6.218	10.917
18	1.642	2.812	4.951	1.350	1.3962156	2.417	6.095	10.585
	1.389	3.152	3.461	0.663	1.3961841	1.188	1.141	10.881

Table D.1: True values (upper line) and recovered median values (lower line) for data set 1.

Binary	i (rad)	φ_0 / rad	A ($\times 10^{-22}$)	ψ / rad	f_0 / mHz	θ / rad	ϕ / rad	ρ
1	2.724	2.486	9.190	0.071	1.0137261	2.486	3.997	33.372
	2.497	3.161	10.258	1.565	1.0137248	2.467	3.987	33.398
2	0.657	5.360	9.485	0.157	0.9914607	1.007	4.245	31.033
	0.634	3.157	9.512	1.569	0.9914620	0.970	4.190	31.117
3	0.490	0.466	8.480	2.625	1.0149462	2.510	3.019	29.325
	0.759	3.130	9.926	1.599	1.0149459	2.490	3.027	29.347
4	0.654	5.595	8.554	0.232	1.0107870	1.179	4.783	29.059
	0.656	3.140	8.878	1.571	1.0107844	1.234	4.760	29.134
5	2.710	5.245	6.090	1.250	0.9907225	1.653	4.248	26.531
	2.499	3.159	7.889	1.568	0.9907217	1.688	4.258	26.581
6	0.087	3.574	6.578	1.917	0.9879189	1.517	1.154	25.948
	0.761	3.098	8.463	1.604	0.9879187	1.502	1.136	25.968
7	0.642	0.205	6.039	1.474	1.0129264	1.694	2.440	22.489
	0.685	3.125	6.448	1.560	1.0129254	1.804	2.430	22.519
8	2.435	1.804	5.927	1.392	0.9962224	1.614	5.307	21.139
	2.363	3.160	6.690	1.579	0.9962234	1.575	5.335	21.217
9	0.416	1.879	5.615	2.911	1.0132963	1.057	0.906	21.055
	0.692	3.127	6.224	1.600	1.0132978	1.079	0.948	20.871
10	0.573	0.381	5.691	1.367	0.9969600	1.705	5.894	18.579
	0.993	0.325	7.371	1.546	0.9969575	1.684	5.814	18.698
11	1.109	6.107	8.611	2.887	0.9932594	0.991	3.671	18.766
	1.128	6.293	9.225	2.788	0.9932599	0.988	3.661	18.778
12	2.918	3.411	5.474	1.230	0.9936598	2.416	3.223	18.298
	2.291	3.209	6.835	1.601	0.9936605	2.372	3.157	18.344
13	0.587	4.908	5.728	1.846	1.0010903	2.781	4.516	17.407
	0.866	3.097	6.929	1.525	1.0010910	2.839	4.481	17.517
14	2.909	5.736	4.362	1.583	1.0105097	1.915	4.873	17.207
	2.414	3.059	5.489	1.617	1.0105107	1.989	4.873	17.292
15	2.355	2.960	6.865	0.269	0.9997605	2.801	5.002	16.237
	2.090	3.164	8.168	1.543	0.9997629	2.842	5.003	16.388
16	0.763	5.151	5.300	3.043	1.0006721	1.475	4.290	15.903
	1.110	4.872	7.285	3.084	1.0006747	1.566	4.321	16.135
17	1.382	4.964	8.471	1.291	1.0087702	1.115	0.025	15.684
	1.334	4.871	9.959	1.292	1.0087716	1.062	0.017	15.791
18	2.413	4.630	4.752	0.491	0.9867635	1.230	4.405	15.853
	2.157	4.796	6.555	0.566	0.9867638	1.168	4.404	15.931
19	2.226	0.092	5.368	2.827	1.0063174	2.011	0.955	15.081
	2.384	3.134	4.746	1.570	1.0063165	2.058	0.949	15.104
20	0.714	1.599	5.244	0.996	0.9849761	1.785	0.078	15.473
	0.749	3.038	5.301	1.501	0.9849765	1.681	0.068	15.604
21	2.519	5.564	4.728	0.527	0.9966749	1.188	6.056	14.957
	2.179	3.109	5.907	1.568	0.9966725	1.215	6.029	14.838
22	1.400	3.590	9.501	2.058	1.0037884	1.387	1.419	15.368
	1.354	0.594	9.750	0.408	1.0037880	1.242	1.459	15.643
23	2.336	1.589	5.933	1.244	0.9944868	2.164	2.792	14.590
	2.314	3.114	5.099	1.569	0.9944899	2.131	2.710	14.644
24	1.068	0.755	6.286	0.107	0.9874341	1.749	2.011	14.236
	0.822	3.113	4.832	1.572	0.9874327	1.726	2.039	14.274
25	1.484	4.598	7.791	1.088	0.9987247	1.759	3.263	13.653
	1.505	1.332	9.130	2.646	0.9987257	1.660	3.227	13.795
26	0.598	0.256	3.732	0.489	1.0143321	1.709	4.371	13.398
	0.859	3.167	4.538	1.584	1.0143343	1.768	4.295	13.587
27	2.598	5.467	3.224	0.452	1.0092818	0.807	5.125	12.370
	2.172	3.135	5.185	1.577	1.0092834	0.678	5.351	12.774
28	1.354	4.636	7.251	1.196	0.9898903	1.722	0.786	12.877
	1.236	5.007	6.642	1.146	0.9898880	1.816	0.754	13.043
29	1.648	0.571	7.034	0.244	0.9983514	0.642	2.039	12.022
	1.578	3.455	7.034	1.850	0.9983527	0.631	2.197	12.055
30	2.713	4.060	3.253	0.771	1.0045681	2.501	5.855	10.966
	2.222	3.121	4.273	1.584	1.0045687	2.517	5.928	10.968

Table D.2: True values (upper line) and recovered median values (lower line) for data set 2.

Bibliography

[1] A. Einstein, "Näherungsweise Integration der Feldgleichungen der Gravitation," *Sitzungsberichte K. Preuss. Akad. Wiss.*, vol. 1, p. 688, 1916.

[2] A. Einstein, "Näherungsweise Integration der Feldgleichungen der Gravitation," *Sitzungsberichte K. Preuss. Akad. Wiss.*, vol. 1, p. 154, 1918.

[3] R. A. Hulse and J. H. Taylor, "Discovery of a pulsar in a binary system," *Astrophysical Journal* , vol. 195, pp. L51–L53, Jan. 1975.

[4] J. H. Taylor and J. M. Weisberg, "A new test of general relativity - gravitational radiation and the binary pulsar psr 1913+16," vol. 253, pp. 908–920, 01 1982.

[5] F. Acernese *et al.*, "Advanced Virgo: a second-generation interferometric gravitational wave detector," *Class. Quant. Grav.*, vol. 32, no. 2, p. 024001, 2015.

[6] J. Aasi *et al.*, "Advanced ligo," *Classical and Quantum Gravity*, vol. 32, no. 7, p. 074001, 2015.

[7] `http://gwcenter.icrr.u-tokyo.ac.jp/en/researcher/parameter`.

[8] `https://www.elisascience.org/files/publications/LISA_L3_20170120.pdf`.

[9] B. P. Abbott *et al.*, "Observation of Gravitational Waves from a Binary Black Hole Merger," *Phys. Rev. Lett.*, vol. 116, no. 6, p. 061102, 2016.

[10] B. P. Abbott *et al.*, "GW151226: Observation of Gravitational Waves from a 22-Solar-Mass Binary Black Hole Coalescence," *Phys. Rev. Lett.*, vol. 116, no. 24, p. 241103, 2016.

[11] B. P. Abbott *et al.*, "GW170104: Observation of a 50-Solar-Mass Binary Black Hole Coalescence at Redshift 0.2," *Phys. Rev. Lett.*, vol. 118, no. 22, p. 221101, 2017.

[12] B. P. Abbott *et al.*, "GW170814: A Three-Detector Observation of Gravitational Waves from a Binary Black Hole Coalescence," *Phys. Rev. Lett.*, vol. 119, no. 14, p. 141101, 2017.

[13] B. P. Abbott *et al.*, "Astrophysical Implications of the Binary Black-Hole Merger GW150914," *Astrophys. J.*, vol. 818, no. 2, p. L22, 2016.

[14] B. P. Abbott *et al.*, "Supplement: The Rate of Binary Black Hole Mergers Inferred from Advanced LIGO Observations Surrounding GW150914," *Astrophys. J. Suppl.*, vol. 227, no. 2, p. 14, 2016.

[15] B. P. Abbott *et al.*, "Tests of general relativity with GW150914," *Phys. Rev. Lett.*, vol. 116, no. 22, p. 221101, 2016.

[16] S. Caroll, *Spacetime and Geometry: An Introduction to General Relativity*. Addison Wesley, 2003.

[17] B. P. Abbott *et al.*, "GW150914: The Advanced LIGO Detectors in the Era of First Discoveries," *Phys. Rev. Lett.*, vol. 116, no. 13, p. 131103, 2016.

[18] R. W. P. Drever, "Gravitational Wave Detectors using Laser Interferometers and Optical Cavities: Ideas, Principals and Prospects," pp. 503–514, 1981. [NATO Sci. Ser. B94,503(1983)].

[19] D. G. Blair, ed., *The Detection of gravitational waves*. 1991.

[20] J. Mizuno, K. A. Strain, P. G. Nelson, J. M. Chen, R. Schilling, A. Ruediger, W. Winkler, and K. Danzmann, "Resonant sideband extraction: A New configuration for interferometric gravitational wave detectors," *Phys. Lett.*, vol. A175, pp. 273–276, 1993.

[21] J. Harms, "Terrestrial gravity fluctuations," *Living Reviews in Relativity*, vol. 18, p. 3, Dec 2015.

[22] R. L. Forward, "Wideband laser-interferometer graviational-radiation experiment," *Phys. Rev. D*, vol. 17, pp. 379–390, Jan 1978.

[23] B. P. Abbott *et al.*, "Prospects for Observing and Localizing Gravitational-Wave Transients with Advanced LIGO and Advanced Virgo," 2013. [Living Rev. Rel.19,1(2016)].

[24] B. S. Sathyaprakash and B. F. Schutz, "Physics, astrophysics and cosmology with gravitational waves," *Living Reviews in Relativity*, vol. 12, p. 2, Mar 2009.

[25] S. Fairhurst, "Triangulation of gravitational wave sources with a network of detectors," *New J. Phys.*, vol. 11, p. 123006, 2009. [Erratum: New J. Phys.13,069602(2011)].

[26] S. Fairhurst, "Source localization with an advanced gravitational wave detector network," *Class. Quant. Grav.*, vol. 28, p. 105021, 2011.

[27] P. Ajith and S. Bose, "Estimating the parameters of nonspinning binary black holes using ground-based gravitational-wave detectors: Statistical errors," *Phys. Rev. D*, vol. 79, p. 084032, Apr 2009.

[28] `https://www.ligo.caltech.edu/image/ligo20160211c`.

[29] S. Hild *et al.*, "Sensitivity Studies for Third-Generation Gravitational Wave Observatories," *Class. Quant. Grav.*, vol. 28, p. 094013, 2011.

[30] B. P. Abbott *et al.*, "Exploring the Sensitivity of Next Generation Gravitational Wave Detectors," *Class. Quant. Grav.*, vol. 34, no. 4, p. 044001, 2017.

[31] M. Armano *et al.*, "Sub-femto-g free fall for space-based gravitational wave observatories: Lisa pathfinder results," *Phys. Rev. Lett.*, vol. 116, p. 231101, Jun 2016.

[32] P. Amaro-Seoane *et al.*, "Low-frequency gravitational-wave science with eLISA/NGO," *Class. Quant. Grav.*, vol. 29, p. 124016, 2012.

[33] G. P. Kuiper, "The Empirical Mass-Luminosity Relation.," *Astrophys. J.*, vol. 88, p. 472, Nov. 1938.

[34] N. Duric, *Advanced astrophysics*. Cambridge University Press, 2004.

[35] `https://www.eso.org/public/images/eso0728c/`.

[36] K.-P. Schroder and R. C. Smith, "Distant future of the Sun and Earth revisited," *Mon. Not. Roy. Astron. Soc.*, vol. 386, p. 155, 2008.

[37] M. Schwarzschild and R. Härm, "Red Giants of Population II. II.," *Astrophys. J.*, vol. 136, p. 158, July 1962.

[38] E. M. Levesque, P. Massey, K. A. G. Olsen, B. Plez, E. Josselin, A. Maeder, and G. Meynet, "The Effective temperature scale of Galactic red supergiants: Cool, but not as cool as we thought," *Astrophys. J.*, vol. 628, pp. 973–985, 2005.

[39] S. Woosley and T. Janka, "The physics of core-collapse supernovae," *Nature Phys.*, vol. 1, p. 147, 2005.

[40] S. O. Kepler, A. D. Romero, I. Pelisoli, and G. Ourique, "White Dwarf Stars," *ArXiv e-prints*, Feb. 2017.

[41] W. Anderson, "Über die grenzdichte der materie und der energie," *Zeitschrift für Physik*, vol. 56, pp. 851–856, Nov 1929.

[42] J. M. Lattimer, "The nuclear equation of state and neutron star masses," *Ann. Rev. Nucl. Part. Sci.*, vol. 62, pp. 485–515, 2012.

[43] `https://en.wikipedia.org/wiki/Neutron_star`.

[44] J. R. Oppenheimer and G. M. Volkoff, "On massive neutron cores," *Phys. Rev.*, vol. 55, pp. 374–381, Feb 1939.

[45] R. C. Tolman, "Static solutions of Einstein's field equations for spheres of fluid," *Phys. Rev.*, vol. 55, pp. 364–373, 1939.

[46] I. Bombaci, "The maximum mass of a neutron star.," *A&A* , vol. 305, p. 871, Jan. 1996.

[47] J. Antoniadis *et al.*, "A Massive Pulsar in a Compact Relativistic Binary," *Science*, vol. 340, p. 6131, 2013.

[48] T. B. Littenberg, B. Farr, S. Coughlin, V. Kalogera, and D. E. Holz, "Neutron stars versus black holes: probing the mass gap with LIGO/Virgo," *Astrophys. J.*, vol. 807, no. 2, p. L24, 2015.

[49] J. McClintock and R. Remillard, "Black hole binaries," 2003.

[50] S. A. Hughes, "Trust but verify: The Case for astrophysical black holes," *eConf*, vol. C0507252, p. L006, 2005.

[51] `https://fr.wikipedia.org/wiki/Ergosphere`.

[52] K. A. Postnov and L. R. Yungelson, "The Evolution of Compact Binary Star Systems," *Living Rev. Rel.*, vol. 17, p. 3, 2014.

[53] c. Wolf and G. Rayet *Comptes rendus de l'Acadmie des sciences,*, vol. 65, p. 292, 1867.

[54] K. S. Thorne and A. N. Zytkow, "Stars with degenerate neutron cores. I - Structure of equilibrium models," *Astrophys. J.*, vol. 212, pp. 832–858, Mar. 1977.

[55] C. W. Helstrom, *Statistical Theory of Signal Detection.* Pergamon Press, London, 1968.

[56] C. Cutler and E. E. Flanagan, "Gravitational waves from merging compact binaries: How accurately can one extract the binary's parameters from the inspiral wave form?," *Phys. Rev.*, vol. D49, pp. 2658–2697, 1994.

[57] C. J. Moore, R. H. Cole, and C. P. L. Berry, "Gravitational-wave sensitivity curves," *Class. Quant. Grav.*, vol. 32, no. 1, p. 015014, 2015.

[58] L. S. Finn, "Detection, measurement and gravitational radiation," *Phys. Rev.*, vol. D46, pp. 5236–5249, 1992.

[59] B. F. Schutz, "Introduction to the analysis of low frequency gravitational wave data," 1997.

[60] E. K. Porter, "Computational resources to filter gravitational wave data with P approximant templates," *Class. Quant. Grav.*, vol. 19, pp. 4343–4360, 2002.

[61] B. J. Owen, "Search templates for gravitational waves from inspiraling binaries: Choice of template spacing," *Phys. Rev. D*, vol. 53, pp. 6749–6761, Jun 1996.

[62] N. J. Cornish and E. K. Porter, "Detecting galactic binaries with LISA," *Class. Quant. Grav.*, vol. 22, pp. S927–S934, 2005.

[63] M. Vallisneri, "Use and abuse of the Fisher information matrix in the assessment of gravitational-wave parameter-estimation prospects," *Phys. Rev.*, vol. D77, p. 042001, 2008.

[64] E. K. Porter and N. J. Cornish, "Fisher versus Bayes: A comparison of parameter estimation techniques for massive black hole binaries to high redshifts with eLISA," *Phys. Rev.*, vol. D91, no. 10, p. 104001, 2015.

[65] J. von Neumann, "Various techniques used in connection with random digits," in *Monte Carlo Method* (A. Householder, G. Forsythe, and H. Germond, eds.), pp. 36–38, Washington, D.C.: U.S. Government Printing Office: National Bureau of Standards Applied Mathematics Series, 12, 1951.

[66] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of State Calculations by Fast Computing Machines," *J. Chem. Phys.*, vol. 21, p. 1087, June 1953.

[67] W. K. Hastings, "Monte Carlo Sampling Methods Using Markov Chains and Their Applications," *Biometrika*, vol. 57, pp. 97–109, 1970.

[68] J. Veitch *et al.*, "Parameter estimation for compact binaries with ground-based gravitational-wave observations using the LALInference software library," *Phys. Rev.*, vol. D91, no. 4, p. 042003, 2015.

[69] C. L. Rodriguez, B. Farr, V. Raymond, W. M. Farr, T. B. Littenberg, D. Fazi, and V. Kalogera, "Basic Parameter Estimation of Binary Neutron Star Systems by the Advanced LIGO/Virgo Network," *Astrophys. J.*, vol. 784, p. 119, 2014.

[70] C. P. L. Berry *et al.*, "Parameter estimation for binary neutron-star coalescences with realistic noise during the Advanced LIGO era," *Astrophys. J.*, vol. 804, no. 2, p. 114, 2015.

[71] E. K. Porter, "An Overview of LISA Data Analysis Algorithms," 2009.

[72] N. J. Cornish and E. K. Porter, "The Search for supermassive black hole binaries with LISA," *Class. Quant. Grav.*, vol. 24, pp. 5729–5755, 2007.

[73] N. J. Cornish and J. Crowder, "LISA data analysis using MCMC methods," *Phys. Rev.*, vol. D72, p. 043005, 2005.

[74] B. P. Abbott *et al.*, "Properties of the Binary Black Hole Merger GW150914," *Phys. Rev. Lett.*, vol. 116, no. 24, p. 241102, 2016.

[75] M. van der Sluys, V. Raymond, I. Mandel, C. Rover, N. Christensen, V. Kalogera, R. Meyer, and A. Vecchio, "Parameter estimation of spinning binary inspirals using Markov-chain Monte Carlo," *Class. Quant. Grav.*, vol. 25, p. 184011, 2008.

[76] R. Storn and K. Price *Journal of Global Optimization*, vol. 11, p. 341, 1997.

[77] C. ter Braak, "A markov chain monte carlo version of the genetic algorithm differential evolution: easy bayesian computing for real parameter spaces," *Stat. Comp.*, vol. 16, no. 3, pp. 239–249, 2006.

[78] C. ter Braak and J. Vrugt, "Differential evolution markov chain with snooker updater and fewer chains," *Stat. Comp.*, vol. 18, no. 4, pp. 435–446, 2008.

[79] A. Hajian, "Efficient Cosmological Parameter Estimation with Hamiltonian Monte Carlo," *Phys. Rev.*, vol. D75, p. 083525, 2007.

[80] L. Baiotti and L. Rezzolla, "Binary neutron star mergers: a review of Einsteins richest laboratory," *Rept. Prog. Phys.*, vol. 80, no. 9, p. 096901, 2017.

[81] E. Berti, V. Cardoso, and A. O. Starinets, "Quasinormal modes of black holes and black branes," *Class. Quant. Grav.*, vol. 26, p. 163001, 2009.

[82] T. Damour, P. Jaranowski, and G. Schaefer, "Equivalence between the ADM-Hamiltonian and the harmonic coordinates approaches to the third postNewtonian dynamics of compact binaries," *Phys. Rev.*, vol. D63, p. 044021, 2001. [Erratum: Phys. Rev.D66,029901(2002)].

[83] T. Damour, P. Jaranowski, and G. Schaefer, "Dimensional regularization of the gravitational interaction of point masses," *Phys. Lett.*, vol. B513, pp. 147–155, 2001.

[84] L. Blanchet, T. Damour, and G. Esposito-Farese, "Dimensional regularization of the third post-Newtonian dynamics of point particles in harmonic coordinates," *Phys. Rev.*, vol. D69, p. 124007, 2004.

[85] V. C. de Andrade, L. Blanchet, and G. Faye, "Third postNewtonian dynamics of compact binaries: Noetherian conserved quantities and equivalence between the harmonic coordinate and ADM Hamiltonian formalisms," *Class. Quant. Grav.*, vol. 18, pp. 753–778, 2001.

[86] L. Blanchet and B. R. Iyer, "Third postNewtonian dynamics of compact binaries: Equations of motion in the center-of-mass frame," *Class. Quant. Grav.*, vol. 20, p. 755, 2003.

[87] Y. Itoh and T. Futamase, "New derivation of a third postNewtonian equation of motion for relativistic compact binaries without ambiguity," *Phys. Rev.*, vol. D68, p. 121501, 2003.

[88] L. Blanchet, T. Damour, G. Esposito-Farese, and B. R. Iyer, "Gravitational radiation from inspiralling compact binaries completed at the third post-Newtonian order," *Phys. Rev. Lett.*, vol. 93, p. 091101, 2004.

[89] L. Blanchet, T. Damour, G. Esposito-Farese, and B. R. Iyer, "Dimensional regularization of the third post-Newtonian gravitational wave generation from two point masses," *Phys. Rev.*, vol. D71, p. 124004, 2005.

[90] L. Blanchet, B. R. Iyer, and B. Joguet, "Gravitational waves from inspiralling compact binaries: Energy flux to third postNewtonian order," *Phys. Rev.*, vol. D65, p. 064005, 2002. [Erratum: Phys. Rev.D71,129903(2005)].

[91] L. Blanchet, G. Faye, B. R. Iyer, and B. Joguet, "Gravitational wave inspiral of compact binary systems to 7/2 postNewtonian order," *Phys. Rev.*, vol. D65, p. 061501, 2002. [Erratum: Phys. Rev.D71,129902(2005)].

[92] L. Blanchet and B. R. Iyer, "Hadamard regularization of the third post-Newtonian gravitational wave generation of two point masses," *Phys. Rev.*, vol. D71, p. 024004, 2005.

[93] C. Cutler, T. A. Apostolatos, L. Bildsten, L. S. Finn, E. E. Flanagan, D. Kennefick, D. M. Markovic, A. Ori, E. Poisson, G. J. Sussman, and K. S. Thorne, "The last three minutes: Issues in gravitational-wave measurements of coalescing compact binaries," *Phys. Rev. Lett.*, vol. 70, pp. 2984–2987, May 1993.

[94] E. Poisson, "Gravitational radiation from a particle in circular orbit around a black hole. 6. Accuracy of the postNewtonian expansion," *Phys. Rev.*, vol. D52, pp. 5719–5723, 1995. [Addendum: Phys. Rev.D55,7980(1997)].

[95] A. Buonanno, B. Iyer, E. Ochsner, Y. Pan, and B. S. Sathyaprakash, "Comparison of post-Newtonian templates for compact binary inspiral signals in gravitational-wave detectors," *Phys. Rev.*, vol. D80, p. 084043, 2009.

[96] T. Damour, B. R. Iyer, and B. S. Sathyaprakash, "Frequency domain P approximant filters for time truncated inspiral gravitational wave signals from compact binaries," *Phys. Rev.*, vol. D62, p. 084036, 2000.

[97] B. Allen, W. G. Anderson, P. R. Brady, D. A. Brown, and J. D. E. Creighton, "FINDCHIRP: An Algorithm for detection of gravitational waves from inspiraling compact binaries," *Phys. Rev.*, vol. D85, p. 122006, 2012.

[98] W. G. Anderson, P. R. Brady, J. D. E. Creighton, and E. E. Flanagan, "An Excess power statistic for detection of burst sources of gravitational radiation," *Phys. Rev.*, vol. D63, p. 042003, 2001.

[99] `https://web.archive.org/web/20120401083859/http://earth-info.nga.mil/GandG/wgs84/index.html`.

[100] B. P. Abbott *et al.*, "Upper Limits on the Rates of Binary Neutron Star and Neutron Starblack Hole Mergers From Advanced Ligos First Observing run," *Astrophys. J.*, vol. 832, no. 2, p. L21, 2016.

[101] J. Abadie *et al.*, "Search for Gravitational Waves from Low Mass Compact Binary Coalescence in LIGO's Sixth Science Run and Virgo's Science Runs 2 and 3," *Phys. Rev.*, vol. D85, p. 082002, 2012.

[102] B. P. Abbott *et al.*, "Binary Black Hole Mergers in the first Advanced LIGO Observing Run," *Phys. Rev.*, vol. X6, no. 4, p. 041015, 2016.

[103] L. Bosi and E. K. Porter, "Data Analysis Challenges for the Einstein Telescope," *Gen. Rel. Grav.*, vol. 43, pp. 519–535, 2011.

[104] L. P. Singer *et al.*, "The First Two Years of Electromagnetic Follow-Up with Advanced LIGO and Virgo," *Astrophys. J.*, vol. 795, no. 2, p. 105, 2014.

[105] P. C. Peters and J. Mathews, "Gravitational radiation from point masses in a keplerian orbit," *Phys. Rev.*, vol. 131, pp. 435–440, Jul 1963.

[106] B. Allen, W. G. Anderson, P. R. Brady, D. A. Brown, and J. D. E. Creighton, "Findchirp: An algorithm for detection of gravitational waves from inspiraling compact binaries," *Phys. Rev. D*, vol. 85, p. 122006, Jun 2012.

[107] N. J. Cornish and E. K. Porter, "Catching supermassive black hole binaries without a net," *Phys. Rev.*, vol. D75, p. 021301, 2007.

[108] C. Cutler, "Angular resolution of the LISA gravitational wave detector," *Phys. Rev.*, vol. D57, pp. 7089–7102, 1998.

[109] S. Duan, A. D. Kennedy, B. Pendleton, and D. Roweth, "Hybrid Monte Carlo," *Phys. Lett. B.*, vol. 195, p. 216, 1987.

[110] R. Neal, *Bayesian learning for neural networks*. PhD thesis, University of Toronto, 1995.

[111] D. J. C. MacKay, *Information theory, inference and learning algorithms*. Cambridge University, 2003.

[112] R. M. Neal, "MCMC using Hamiltonian dynamics," *ArXiv e-prints*, June 2012.

[113] M. Girolami, B. Calderhead, and S. A. Chin, "Riemannian Manifold Hamiltonian Monte Carlo," *ArXiv e-prints*, July 2009.

[114] E. K. Porter and J. Carr, "A Hamiltonian Monte Carlo method for Bayesian Inference of Supermassive Black Hole Binaries," *Class. Quant. Grav.*, vol. 31, p. 145004, 2014.

[115] M. D. Hoffman and A. Gelman, "The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo," *ArXiv e-prints*, Nov. 2011.

[116] `http://www.apc.univ-paris7.fr/FACe/cluster-arago`.

[117] D. S. Broomhead and D. Lowe, "Multivariable functional interpolation and adaptive networks," *Complex Systems*, vol. 2, 1988.

[118] V. Skala, "A practical use of radial basis functions interpolation and approximation," *Revista ivestigacion operacional*, vol. 37, p. 137, 2016.

[119] V. Skala, "Rbf interpolation with csrbf of large data sets," *Procedia Computer Science*, vol. 108, pp. 2433 – 2437, 2017. International Conference on Computational Science, ICCS 2017, 12-14 June 2017, Zurich, Switzerland.

[120] `https://www.gnu.org/software/gsl/`.

[121] L. P. Singer and L. R. Price, "Rapid Bayesian position reconstruction for gravitational-wave transients," *Phys. Rev.*, vol. D93, no. 2, p. 024013, 2016.

[122] G. Nelemans, L. R. Yungelson, and S. F. Portegies Zwart, "The gravitational wave signal from the galactic disk population of binaries containing two compact objects," *Astron. Astrophys.*, vol. 375, pp. 890–898, 2001.

[123] A. J. Ruiter, K. Belczynski, M. Benacquista, S. L. Larson, and G. Williams, "The LISA Gravitational Wave Foreground: A Study of Double White Dwarfs," *Astrophys. J.*, vol. 717, pp. 1006–1021, 2010.

[124] N. J. Cornish and J. Crowder, "LISA data analysis using MCMC methods," *Phys. Rev.*, vol. D72, p. 043005, 2005.

[125] P. Amaro-Seoane, S. Aoudia, S. Babak, P. Bintruy, E. Berti, A. Boh, C. Caprini, M. Colpi, N. J. Cornish, K. Danzmann, and others, "eLISA: Astrophysics and cosmology in the millihertz regime," *arXiv preprint arXiv:1201.3621*, 2012.

[126] N. J. Cornish and S. L. Larson, "LISA data analysis: Source identification and subtraction," *Phys. Rev.*, vol. D67, p. 103001, 2003.

[127] A. Blaut, S. Babak, and A. Krlak, "Mock LISA data challenge for the Galactic white dwarf binaries," *arXiv preprint arXiv:0911.3020*, 2009.

[128] J. T. Whelan, R. Prix, and D. Khurana, "Searching for Galactic White Dwarf Binaries in Mock LISA Data using an F-Statistic Template Bank," *Class. Quant. Grav.*, vol. 27, p. 055010, 2010.

[129] N. J. Cornish and E. K. Porter, "Detecting galactic binaries with LISA," *Class. Quant. Grav.*, vol. 22, pp. S927–S934, 2005.

[130] J. Crowder, N. J. Cornish, and L. Reddinger, "Darwin meets Einstein: LISA data analysis using genetic algorithms," *Phys. Rev.*, vol. D73, p. 063011, 2006.

[131] S. D. Mohanty and R. K. Nayak, "Tomographic approach to resolving the distribution of LISA Galactic binaries," *Phys. Rev.*, vol. D73, p. 083006, 2006.

[132] M. Trias, A. Vecchio, and J. Veitch, "Studying stellar binary systems with the Laser Interferometer Space Antenna using Delayed Rejection Markov chain Monte Carlo methods," *Class. Quant. Grav.*, vol. 26, p. 204024, 2009.

[133] J. Crowder and N. Cornish, "A Solution to the Galactic Foreground Problem for LISA," *Phys. Rev.*, vol. D75, p. 043008, 2007.

[134] N. J. Cornish and E. K. Porter, "The Search for supermassive black hole binaries with LISA," *Class. Quant. Grav.*, vol. 24, pp. 5729–5755, 2007.

[135] N. J. Cornish and E. K. Porter, "Searching for Massive Black Hole Binaries in the first Mock LISA Data Challenge," *Class. Quant. Grav.*, vol. 24, pp. S501–S512, 2007.

[136] A. Vecchio and E. D. L. Wickham, "The Effect of the LISA response function on observations of monochromatic sources," *Phys. Rev.*, vol. D70, p. 082002, 2004.

[137] A. Stroeer, J. Gair, and A. Vecchio, "Automatic Bayesian inference for LISA data analysis strategies," *AIP Conf. Proc.*, vol. 873, pp. 444–451, 2006. [,444(2006)].

[138] J. Veitch and A. Vecchio, "A Bayesian approach to the follow-up of candidate gravitational wave signals," *Phys. Rev.*, vol. D78, p. 022001, 2008.

[139] N. J. Cornish and T. B. Littenberg, "Tests of Bayesian Model Selection Techniques for Gravitational Wave Astronomy," *Phys. Rev.*, vol. D76, p. 083006, 2007.

[140] T. B. Littenberg and N. J. Cornish, "A Bayesian Approach to the Detection Problem in Gravitational Wave Astronomy," *Phys. Rev.*, vol. D80, p. 063007, 2009.

[141] S. Babak and others, "Report on the second Mock LISA Data Challenge," *Classical Quantum Gravity*, 2008.

[142] J. Kennedy and R. Eberhart *IEEE International Conference on Neural Networks Proceedings*, vol. 4, p. 1942, 1995.

[143] Y. Shi and R. Eberhart *The 1998 IEEE International Conference on Evolutionary Computation Proceedings*, p. 69, 1998.

[144] N. J. Cornish and L. J. Rubbo, "The LISA response function," *Phys. Rev.*, vol. D67, p. 022001, 2003. [Erratum: Phys. Rev.D67,029905(2003)].

[145] A. Krolak, M. Tinto, and M. Vallisneri, "Optimal filtering of the LISA data," *Phys. Rev.*, vol. D70, p. 022003, 2004. [Erratum: Phys. Rev.D76,069901(2007)].

[146] A. Petiteau, S. Yu, and S. Babak, "The Search for spinning black hole binaries using a genetic algorithm," *Class. Quant. Grav.*, vol. 26, p. 204011, 2009.

[147] M. G. H. Omran, A. P. Engelbrecht, and A. Salman *European Journal of Operational Research*, 2008.

[148] J. R. Gair and E. K. Porter, "Cosmic Swarms: A Search for Supermassive Black Holes in the LISA data stream with a Hybrid Evolutionary Algorithm," *Class. Quant. Grav.*, vol. 26, p. 225004, 2009.

[149] S. R. Taylor, J. R. Gair, and L. Lentati, "Using Swarm Intelligence To Accelerate Pulsar Timing Analysis," 2012.

[150] Y. Wang, S. D. Mohanty, and F. A. Jenet, "A coherent method for the detection and estimation of continuous gravitational wave signals using a pulsar timing array," *Astrophys. J.*, vol. 795, no. 1, p. 96, 2014.

[151] Y. Wang and S. D. Mohanty, "Particle Swarm Optimization and gravitational wave data analysis: Performance on a binary inspiral testbed," *Phys. Rev.*, vol. D81, p. 063002, 2010.

[152] J. Prasad and T. Souradeep, "Cosmological parameter estimation using Particle Swarm Optimization (PSO)," *Phys. Rev.*, vol. D85, no. 12, p. 123008, 2012. [Erratum: Phys. Rev.D90,no.10,109903(2014)].

[153] `http://www.particleswarm.info/Standard_PSO_2006.c`.

[154] C. Huwyler, E. K. Porter, and P. Jetzer, "Supermassive Black Hole Tests of General Relativity with eLISA," *Phys. Rev.*, vol. D91, no. 2, p. 024037, 2015.

[155] M. Kilic, W. R. Brown, and J. J. Hermes, "Ultra-Compact Binaries: eLISA Verification Sources," *ASP Conf. Ser.*, vol. 467, pp. 47–58, 2013.

[156] G. H. A. Roelofs, A. Rau, T. R. Marsh, D. Steeghs, P. J. Groot, and G. Nelemans, "Spectroscopic Evidence for a 5.4-Minute Orbital Period in HM Cancri," *Astrophys. J.*, vol. 711, p. L138, 2010.

[157] E. K. Porter, "Effects of different eLISA-like configurations on massive black hole parameter estimation," *Phys. Rev.*, vol. D92, no. 6, p. 064001, 2015.