Parameter estimation on gravitational-wave signals from compact binary inspirals with arbitrary spin using Markov-chain Monte Carlo

Marc van der Sluys

(Dated: March 28, 2013)

We present a parameter-estimation code which uses a Markov-chain Monte Carlo (MCMC) technique to study the source parameters of gravitational-wave signals from the inspirals of stellar-mass compact binaries detected with ground-based gravitational-wave detectors such as LIGO and Virgo, including the spins of the binary members.

We discuss a number of experimental aspects of the MCMC algorithm that may allow us to sample the parameter space in an efficient way and assess their effect. We conclude that there seems to be no need to use adaptive update proposals, that correlated update proposals should only be used in the burn-in phase, that parallel tempering has a clearly positive effect and that sinusoidal temperatures may be used in parallel tempering after the burn-in.

In addition, we show the analyses of a few example inspiral signals of double black holes to indicate the typical accuracy that can be expected from GW observations. In particular, we discuss the effect that the presence of spin in the black holes has on the analyses, both for the case where spins are allowed for in the MCMC and for the case where the black holes are assumed to be non-spinning.

I. INTRODUCTION

Inspirals of stellar-mass compact binaries induced by gravitational radiation are among the most promising gravitational-wave sources for ground-based laser interferometers, such as LIGO [1–3] and Virgo [4, 5]. If such a binary contains a black hole (BH), theoretical studies suggest it can be spinning moderately [6]. To date, more-or-less accurate values for the dimensionless spins ($a_{\rm spin}$) of about 15 black holes in X-ray binaries (XRBs) have been determined observationally [see *e.g.* 7–10, and references therein]. Fender et al. [11] compile a list of "trustworthy" spin values for fifteen BHs in XRBs, of which four have "low" spin $a_{\rm spin} \lesssim 0.4$ and the remaining 11 have "high" spin $a_{\rm spin} \gtrsim 0.6$ (see their Table 1 and Figure 1, and their references to the observational papers). Recent measurements of the BH-spin in Cygnus X-1 suggest that it may be close to maximum rotation, shifting the low vs. high spin balance for these 15 binaries to 3 – 12 [10, 12, 13]. However, it is not yet clear what the observational biases are. Observed long-term (one to three orders of magnitude longer than the orbital period variations in the X-ray luminosity of black-hole XRBs may be due to the precession of the inner accretion disc [*e.g.* 14, 15]. For a compact binary inspiral, a misalignment of a spinning black hole with the orbital angular momentum will cause the binary orbit to precess in a similar way, introducing phase and amplitude modulations in the gravitational-wave signal. This effect should be taken into account in the data analysis. The accuracy with which the binary parameters can be estimated from the gravitational-wave observation is of significant astrophysical interest.

We developed a code called SPINspiral [16] which implements a Markov-chain Monte Carlo (MCMC) technique [17] to compute the posterior probability-density functions (PDFs) of the source parameters, including the spins of the binary components. This code is a modification of an earlier parameter-estimation code for analysis on binaries without spin [18, 19]. In addition to including post-Newtonian gravitational waveforms with one [20] or two [21] spinning objects, we have also implemented a number of improvements designed to make the parameter-space exploration more efficient, such as parallel tempering.

First results obtained with this code are described in a number of earlier papers, discussing the accuracies of parameter estimation on the inspiral signal from a binary consisting of a black hole and a neutron star (NS) and the influence of the spin of the BH on that accuracy [22], the degeneracies in sky localisation for a detection with only two interferometers, and the influence of spin on that degeneracy [23], the analysis of hardware injections into LIGO data [24], and the effect of using LIGO data instead of Gaussian noise [25]. The code was also used in the NINJA project [26, 27]. A brief introduction to our MCMC implementation was published in [28], and in the current paper, we describe this implementation more fully, and discuss the complications that are introduced when the effect of spin is present in binary-inspiral signals, and the reason why allowing for spin in the parameter estimation is of pre-eminent importance.

SPINspiral was included in the automated LIGO-Virgo compact-binary coalescence (CBC) *follow-up* analysis pipeline [29] to estimate the physical parameters from the interferometer data after the CBC *detection* pipeline [30] has identified the event in the data stream and concluded that it is indeed a binary inspiral signal. The code is part of the LSC Algorithm Library suite [LALSuite; 31].

This paper is organised as follows. In section II we describe the way our code handles data, our choice of waveforms and the basic signal characteristics we use. Section III deals with the implementation of the MCMC algorithm in the code and related details, such as the choice of parameters, prior distributions, how the chains are started, implemented update proposals, and parallel tempering. We describe the diagnostics that allow us to verify the quality of an analysis, and hence the validity of the results, in section IV. The statistical analysis of the Markov chains and the representation of the posterior PDF are discussed in section V. In section VI we present the results of example MCMC analyses for a range of different signals, ob-

tained with SPINspiral. In section VII we discuss the strong points and weaknesses of our code, summarise our conclusions and describe planned future improvements.

II. DATA AND SIGNAL

A. Data handling

In order to obtain a data set, the code can either read detector data directly, or we can simulate a gravitational-wave signal embedded in noise by *injecting* a waveform with parameters of our choice into a stretch of Gaussian noise (*i.e.*, adding the signal to the noise). The resulting data set is filtered, downsampled, Fourier transformed, and subsequently examined by the MCMC analysis. A different stretch of data, where no signal is supposed to be present, is used to estimate the power spectral density (PSD). More details of the data handling can be found in Röver [19].

Test analyses show that the results of parameter estimation on detector data are qualitatively similar to the results using Gaussian noise, but can result in somewhat larger uncertainties when the data are less well-behaved, especially for low SNR [25].

1. Data reading

Noise data are read from files in the LIGO/Virgo Frame format [32] in the time domain. This format allows us to read detector data which contains some 'signal', *e.g.* an interesting trigger or an either known or blind hardware injection [33], which we would like to analyse, or presumed 'clean' detector data or Gaussian noise at the designed sensitivity level for the detectors [34] to do a software injection of a signal of our choice. The Frame format also allows different groups to exchange data, which allowed us to participate in *e.g.* the NINJA project [26, 27].

Detector data are read in at a sampling rate of 16 kHz or 20 kHz for the LIGO and Virgo detectors respectively.

2. Software injections of waveforms

Software injections are a useful method to test our MCMC code, since a specific signal can be selected and because the signal parameters are known, so that it can easily be verified whether the correct parameter values were found. If a software injection is to be performed, this is done directly after the data is read in, at the original sampling frequency and before any filtering or downsampling are done. Thus, we obtain a data set that is a good representation of a stretch of detector data containing a signal.

3. Filtering and downsampling

After a data set with signal is obtained, either by reading the detector data in directly from a Frame file, or by the software injection of a waveform, the data set is first filtered and then downsampled.

The data set is low-pass filtered using a finite-impulse-response (FIR) filter. We typically find that using 129 coefficients is sufficient and use a *Remez exchange* [*e.g.* 35] algorithm [36] to compute these coefficients. The frequency ranges of the pass, transition and stop band are determined as follows.

The lower cut-off frequency f_{low} is effectively determined by the bandwidth of the detector used. We typically use $f_{\mathrm{low}} = 40\,\mathrm{Hz}$ for Initial LIGO/Virgo. We want to filter out all frequencies higher than the Nyquist frequency f_{Nq}, which is determined by the sampling frequency that we want to use for the MCMC analysis f_{MCMC} (see below). For that purpose, we use a transition band to smoothly connect the low-frequency pass band to the high-frequency stop band, and $f_{\mathrm{Nq}} = \frac{1}{2} f_{\mathrm{MCMC}}$ is the maximum frequency for the boundary between the transition band and the stop band. The width of the transition band is arbitrary, and should both be narrow enough to attenuate as little of the signal as possible, and broad enough to provide a smooth result. We find that a width of $0.025\, f_{\mathrm{MCMC}}$ provides a good and stable compromise. This then defines the maximum frequency for the boundary between the pass band and transition band as $0.475\, f_{\mathrm{MCMC}}$. The high cut-off frequency f_{high} for the templates that we use should not be higher than this frequency ($f_{\mathrm{high}} < 0.475\, f_{\mathrm{MCMC}}$), but may be manually set to a value lower than that. Thus, for a typical analysis with a sampling frequency $f_{\mathrm{MCMC}} = 4\,\mathrm{kHz}$, the pass band would cover 40.0 – 1946 Hz, the transition band 1.9 – 2.0 kHz and everything above 2 kHz would be completely attenuated. In such a case we would use $f_{\mathrm{low}} = 40\,\mathrm{Hz}$ and $f_{\mathrm{high}} \leq 1900\,\mathrm{Hz}$ as frequency cutoffs for the waveform template.

After filtering, the data are usually downsampled to reduce the data set and hence the time needed for the analysis. We typically use a sampling frequency for the analysis of $f_{\mathrm{MCMC}} = 4\,\mathrm{kHz}$, but we can downsample to 1 kHz or less, if the frequency of the innermost stable orbit (ISCO; a function of mass and spin [37]) of the presumed source allows it. Hence the factor by which we downsample the data, usually a power of two, is determined by the masses and spins of the binary components, where higher mass and/or lower spin allows for a lower upper cut-off frequency, hence more downsampling and a faster analysis. In the case of a real detection, the detection trigger would provide us with a good estimate of the masses and hence of the downsample factor that should be used. The downsampling itself is achieved simply by thinning the data, using the weighted mean of a number of data points to provide the value of the thinned data point.

4. Windowing and Fourier transformation

In order to analyse the data in the frequency domain, they need to be Fourier transformed. To reduce artificial contributions from outside the frequency band we are interested in, such as bleeding, the data are windowed. At the moment, the data stretch in the time domain that is supposed to contain the signal ($t_{\rm low}$ – $t_{\rm high}$) is marked manually, but this can easily be automated from the supposed time of coalescence, the chirp mass, as provided by *e.g.* a detection trigger, and $f_{\rm low}$. Because of the rounding off that will be discussed below, the non-spinning, Newtonian approximation is sufficiently accurate (typically to $\approx 1\%$) to estimate the time between the moment a source with a chirp mass $\mathcal{M}$ enters the detector band at $f_{\rm low}$ and the moment of coalescence:

$$t_{\rm coal} = 5(8\pi)^{-8/3} \left(\frac{f_{\rm low}}{\rm Hz}\right)^{-8/3} \left(\frac{\mathcal{M}}{M_\odot}\right)^{-5/3} \text{seconds}. \tag{1}$$

In order to window the data, we use a Tukey window. For the i-th element of a discrete data set consisting of N points ($0 \leq i < N$), the windowing factor is given by:

$$\mathcal{W}(i) = \frac{1-\cos\left(\frac{2\pi}{\alpha}\frac{i}{N}\right)}{2}, \quad 0 \leq i \leq \frac{\alpha N}{2} \tag{2}$$

$$= 1, \qquad \frac{\alpha N}{2} \leq i \leq N - \frac{\alpha N}{2} \tag{3}$$

$$= \frac{1-\cos\left(\frac{2\pi}{\alpha}\frac{i}{N-i}\right)}{2}, \quad N - \frac{\alpha N}{2} \leq i < N \tag{4}$$

(*e.g.* [38]). This window has a flat pass band bordered by two cosine-shaped transition bands at high and low frequency. The parameter $0 \leq \alpha \leq 1$ determines the fraction of the window that is sinusoidal. The time range of the pass band is set by $t_{\rm low}$ and $t_{\rm high}$ and has a width $\Delta t = t_{\rm high} - t_{\rm low}$. We find that $\alpha = 0.15$ is a good compromise: the attenuation is sufficient, while the width of each wing of the Tukey window is set to $\approx 0.09\ \Delta t$, hence the increase in the size of the data set is only 18%. This optimal value was found experimentally by increasing α starting from a low value, computing the SNR, and selecting the value for α where the SNR stabilised. For LIGO data alone, $\alpha \approx 0.05$ was sufficient, but when Virgo data are used $\alpha \approx 0.15$ is needed. The boundaries of the data stretch $t_{\rm low}$ and $t_{\rm high}$ are then updated to enclose the transition regions of the Tukey window.

The beginning and end of the data stretch in the time domain are rounded off to the nearest second before $t_{\rm low}$ and after $t_{\rm high}$. This rounding off ensures that the whole signal is included.

The windowed data set is then Fourier transformed using the Fast Fourier Transform (FFT) algorithm from the `fftw3` library [39]. Finally, the Fourier-transformed data set is normalised by dividing by the sampling rate to provide the frequency-domain data set that can be analysed by the MCMC.

5. PSD generation

When reading in detector noise, the noise estimates are based on data taken close to the time of the signal, without including the signal itself. A stretch of 256 seconds of data, divided in 31 overlapping segments of 16 s each, is used to estimate the one-sided power-spectral density (PSD) of the noise $S_{\rm n}(f)$.

The data for the PSD estimation are filtered, downsampled, windowed and Fourier transformed in a similar way as the data that are to be analysed by the MCMC (see Sects. II A 3 and II A 4). The downsampling of the PSD happens in log(PSD), by taking the weighted mean of the two adjacent log(PSD) values for each new (downsampled) frequency bin. The noise PSD is used to compute the overlap of two waveforms, which is needed to calculate the likelihood and SNR (see Sect. II C).

B. Waveforms

1. Apostolatos, 1.5-pN, single-spin waveform

In order to quickly test our methods, we use a simplified waveform that takes into account post-Newtonian (pN) expansions up to the 1.5-pN order in phase and is restricted to the Newtonian order in amplitude. The waveform includes the simple-precession prescription [20]. This choice of waveform template allows us to investigate the first-order effects of spin (spin-orbit coupling), as long as either only one binary member has spin, or the mass ratio is sufficiently far from unity. Example waveforms for different values of spin are shown in Fig. 1. In comparison to higher-pN or double-spin waveforms, this waveform can be computed analytically and has lower dimensionality, so that the computational cost per iteration is lower and the number of iterations needed for convergence is smaller.

The assumption of only one spinning component can be valid in cases where only one of the binary members is spinning, but also in the case of a large mass ratio (*e.g.* a $10\,M_\odot$ BH and a $1.4\,M_\odot$ NS), in which case the spin of the less massive object can be neglected.

Each waveform template is computed in the time domain, and then windowed and Fourier transformed. The calculation of the likelihood, which measures how well a model waveform matches the data, is carried out in the frequency domain.

2. SpinTaylor, 3.5-pN, double-spin waveform

Although the 1.5-pN, simple-precession waveform is useful to investigate the principal effects of spin on parameter estimation, a more realistic waveform is needed to properly analyse detected signals. The waveform we use for this is the SpinTaylor waveform, which provides waveforms up to the 3.5-pN order in phase and at the Newtonian order in amplitude. We use the implementation in the LSC Algorithm Library [LAL; 31], which closely follows [21]. In addition to the higher pN order, the SpinTaylor waveform allows for spins on both of the binary members and takes into account effects like spin-spin coupling and Thomas

FIG. 1: Last second of the tapered detector signal (strain) from a $10.0 + 1.4\,M_\odot$ BH-NS binary inspiral with a non-spinning NS and a BH with $a_{\rm spin1} = 0.0$ (top panel), $a_{\rm spin1} = 0.5$ (middle panel) and $a_{\rm spin1} = 1.0$ (bottom panel). The angle between the orbital and spin angular-momentum vectors ($\theta_{\rm spin1}$) is 20°. The vertical scale is the same in the three panels, the three SNRs scale as $1.00 : 1.10 : 1.20$. The changes in morphology and SNR between the signals depend on the projection of the wave onto a detector. The waveform type used is that using simple precession, by [20] as described in our Sect.II B 1.

FIG. 2: Last second of the tapered detector signal (strain) from a $11.0+7.0\,M_\odot$ BH-BH binary inspiral. The spin magnitudes are ($a_{\rm spin1}$,$a_{\rm spin2}$) = (0.0,0.0), (0.9,0.0), (0.0,0.7) and (0.9,0.7), the angles between orbit and spin are ($\theta_{\rm spin1}$,$\theta_{\rm spin2}$) = (0°,0°), (70°,0°), (0°,50°) and (70°,50°) for the four panels, respectively. The four SNRs scale as $1.00 : 1.14 : 1.09 : 1.18$. The changes in morphology and SNR between the signals depend on the projection of the wave onto a detector. The deviations from a simple chirp in the last panel are dominated by the spin from the more massive, faster spinning binary component. These deviations are less dramatic than in Fig.1 in part because the mass ratio lies closer to unity in this case. We used the SpinTaylor waveform described in Sect.II B 2.

precession. Hence, the SpinTaylor templates allow us to compute waveforms that are quite close to those which nature provides, as has been shown by comparison to numerical-relativity waveforms [*e.g.* 40–45]. Using the higher pN-order of the SpinTaylor waveform as an MCMC template, instead of the 1.5-pN order waveform, ensures smaller biases in the parameter estimation, due to the smaller mismatch between the template and waveforms found in nature.

In addition to that, the SpinTaylor waveform also contains more detailed information about the source, enabling us to estimate the source parameters more accurately. In an early study, we showed that this improvement in accuracy can be on the order of a factor of two for the individual masses [23].

Finally, as we will demonstrate in Section VI,

parameter-estimation results can yield strong systematic errors when the model that is used is does not correctly or fully describe the source of the signal. Hence, if one conducts parameter estimation on a signal from a binary with a spinning member, but using a waveform template that does not take into account the spin, or if the signal contains effects of two spins whereas only one is fitted for, the results (especially chirp mass) cannot be trusted. Hence, a more general waveform typically makes a better template for parameter estimation, and SpinTaylor is clearly preferable over more simplified waveforms for the final analysis.

However, this precision comes at a cost. The waveform templates can no longer be computed analytically, and involve numerical integration of differential equations. Compared to the Apostolatos et al. waveform, the CPU cost of a single SpinTaylor template is roughly 2.5 times as high. Hence, much work still needs to be done to improve the turnover timescale for analyses with this waveform (Sect. VII).

3. GeneratePPN, 3.5-pN, no-spin waveform

We also incorporated a non-spinning waveform from LAL to analyse the effects of spins on parameter estimation, GeneratePPN [46]. It is the fastest template, and has the lowest dimensionality (9 parameters), allowing for the shortest MCMC runs. As for the SpinTaylor waveform, GeneratePPN goes up to 3.5 pN order in phase and is Newtonian in amplitude.

4. PhenSpin, 3.5-pN, phenomenological double-spin waveform

For high masses ($35\,M_\odot$ and higher) the ringdown of the waveform is expected to be in the LIGO/Virgo band and contribute non-negligibly to the SNR. In order to analyse those signals, we use the phenomenological waveform PhenSpin [47]. It consists of a SpinTaylor inspiral phase to which is added a phenomenological merger-and-ringdown part tuned to Georgia Tech numerical waveforms.

5. Frequency cut-offs and tapering

The time-domain waveform is computed in principle between the low and high cut-off frequencies $f_{\rm low}$ and $f_{\rm high}$ as defined in Sect. II A 3. However, we also verify that the frequency increases monotonously as a function of time. When this condition is no longer fulfilled, *i.e.* when the gravitational-wave frequency decreases between two subsequent time points, we judge the remainder of the waveform as unphysical and treat this as having reached $f_{\rm high}$ by cutting off the waveform.

While the abrupt start of our waveform template at $f_{\rm low}$ has relatively little effect — since it takes place in the low-frequency wing of the detector noise curve, where the noise level is high — the end of the waveform template typically occurs at higher frequencies, where the detectors are more sensitive. The abrupt cut-off of the waveform at the high-frequency end of the template would result in a discontinuous amplitude and would thus cause spurious effects in the Fourier transform. To suppress these effects, we taper off the time-domain waveform at the high-frequency end, by multiplying the amplitude with a factor

$$f_{\rm tap}(v_{\rm orb}, v_{\rm orb0}) = \frac{1 - \tanh\left[100\left(v_{\rm orb}^2 - v_{\rm orb0}^2\right)\right]}{2}, \quad (5)$$

where $v_{\rm orb}$ is the Keplerian orbital velocity, and $v_{\rm orb0}$ is that quantity at one binary orbital period before the waveform is cut off. The factor 100 forces the taper to take place over approximately 1–2 gravitational-wave cycles.

6. Scaling injection signal-to-noise ratio

The signal-to-noise ratio (SNR, Eqs. 14,15) of an injected signal depends, apart from the binary parameters, on the detector orientation and the exact noise realisation. While changing injection parameters like the masses or spins between two injections will have similar effects (change in amplitude, frequency evolution and "shape" of the waveform) in different interferometers, the effect of a change in sky position or binary orientation differs from interferometer to interferometer in a detector network, due to the different projections of the signal onto the detector (known as the "antenna pattern").

Since the accuracy of parameter estimation scales roughly with 1/SNR [*e.g.* 48], we need to eliminate this influence when determining the effect of a change in setup or a different source. In order to obtain the desired SNR $\rho_{\rm tot,0}$ we first inject the signal into the noise for a typical distance (*e.g.* $d_{\rm L,0} = 20\,$Mpc) and compute the SNR $\rho_{\rm tot}$ of that injection using Eq. 15. The waveform is then re-injected in the same stretch of noise but with the new distance

$$d_{\rm L} = d_{\rm L,0} \cdot \frac{\rho_{\rm tot}}{\rho_{\rm tot,0}}. \quad (6)$$

Scaling the SNR by changing the distance scales the amplitude only, and hence preserves the frequency evolution and projection of the waveform onto the different detectors.

C. Signal characteristics

1. Overlap

The overlap between the two signals $a(f)$ and $b(f)$ in the frequency domain is defined as:

$$\langle a|b\rangle \equiv 4\,{\rm Re}\left(\int_0^\infty \frac{a(f)b^*(f)}{S_{\rm n}(|f|)}df\right), \quad (7)$$

where Re() is a function that takes the real part of an expression, the asterisk denotes a complex conjugate and $S_{\rm n}(|f|)$ is the one-sided power-spectral density.

2. Likelihood

We follow a Bayesian approach to infer the posterior probability-density functions (PDFs), also simply called *posteriors*, of the 12–15 parameters that describe our waveform. The posterior of a parameter vector $\vec{\lambda}$ describing a fixed model M and given an observed data set d follows from Bayes' theorem:

$$p(\vec{\lambda}|d,M) = \frac{p(\vec{\lambda}|M)\,p(d|\vec{\lambda},M)}{p(d|M)} \;\propto\; p(\vec{\lambda})\,L(d|\vec{\lambda}), \quad (8)$$

where $p(\vec{\lambda})$ is the *prior* distribution of the parameters, and $L(d|\vec{\lambda})$ is the *likelihood* function. We calculate the likelihood for a model waveform $m(\vec{\lambda})$ with parameters $\vec{\lambda}$ and data set d as measured by a detector i in the usual way, using the notation from Eq. 7 for the overlap:

$$L_i(d|\vec{\lambda}) \propto \exp\left(-\frac{1}{2}\langle d-m(\vec{\lambda})|d-m(\vec{\lambda})\rangle\right) \quad (9)$$

Since we will be considering the *ratio* of likelihoods, we do not need to take into account the normalisation factor, and it is sufficient to compute the proportionality in Eq. 9.

The practical implementation of Eq. 9 is done by defining the *relative* likelihood as

$$\log\left(L_i(d|\vec{\lambda})\right) = \langle d|m(\vec{\lambda})\rangle - \frac{1}{2}\langle m(\vec{\lambda})|m(\vec{\lambda})\rangle, \quad (10)$$

where log is the natural logarithm and the brackets denote the overlap of two signals as defined in Eq. 7. Equation 10 defines the likelihood of the data for a null-model signal (*i.e.*, $m(\vec{\lambda}_{\rm null}) = 0$) as $\log(L_{{\rm null},i}) \equiv \log\left(L_i(d|\vec{\lambda_{\rm null}})\right) = 0$, or $L_{{\rm null},i} = 1$.

For a typical analysis we use the data from two or (preferably) three non-colocated interferometers (the two 4-km LIGO detectors at Hanford and Livingston, and the 3-km Virgo detector near Pisa). The noise of these interferometers is independent, in which case the total likelihood for a coherent network of N detectors can be found by multiplying the individual likelihoods:

$$L_{\rm tot}(d|\vec{\lambda}) = \prod_{i=1}^{N} L_i(d|\vec{\lambda}). \quad (11)$$

The expression for the PDF generalises to

$$p(\vec{\lambda}|d) \propto p(\vec{\lambda}) L_{\rm tot}(d|\vec{\lambda}), \quad (12)$$

while the null-likelihood remains

$$L_{\rm null} = \prod_{i=1}^{N} L_{{\rm null},i} = 1. \quad (13)$$

3. Signal-to-noise ratio

The signal-to-noise ratio (SNR or ρ) of an injected model signal m with a parameter set $\vec{\lambda}$ as detected by a single detector i can be computed as follows:

$$\rho_i(\vec{\lambda}) = \sqrt{4\int_{f_{\rm low}}^{f_{\rm high}} \frac{\left|\tilde{m}(\vec{\lambda},f)\right|^2}{S_{\rm n}(f)} df}. \quad (14)$$

In this expression, $\tilde{m}(\vec{\lambda},f)$ is the frequency-domain model waveform and $S_{\rm n}(f)$ is the noise PSD. For numerical calculations the integral is replaced with a sum, df with the width of the frequency bins Δf, and the sum is computed over the frequency bins between the frequency cut-offs $f_{\rm low}$ and $f_{\rm high}$. The total SNR for a network of N detectors is given by

$$\rho_{\rm tot}(\vec{\lambda}) = \sqrt{\sum_{i=1}^{N}\left(\rho_i(\vec{\lambda})\right)^2}. \quad (15)$$

For a given MCMC analysis, the *expected* network SNR can be obtained from the maximum-likelihood point found in the Markov chains from

$$\rho_{\rm tot} = \sqrt{2\,\log\left(\frac{L_{\rm max}}{L_{\rm null}}\right)}, \quad (16)$$

which, because of Eq. 13, reduces to

$$\rho_{\rm tot} \sim \sqrt{2\,\log\left(L_{\rm max}\right)}. \quad (17)$$

Henceforth, we will refer to the expected SNR as simply "SNR".

III. IMPLEMENTATION OF MCMC

The code SPINspiral is based on a Markov-chain Monte Carlo code that was developed for the analysis of binary-inspiral signals where no spin is present [18, 19, 49, 50]. In this section, we describe the features of our implementation, some of which are taken from this earlier code, and some of which were introduced in the present code for use on inspirals with one or two spinning objects.

A. Choice of MCMC parameters

The waveform for an inspiral with non-spinning objects is described by nine parameters: the chirp mass $\mathcal{M} \equiv \frac{(M_1M_2)^{3/5}}{(M_1+M_2)^{1/5}}$, symmetric mass ratio $\eta \equiv \frac{M_1M_2}{(M_1+M_2)^2}$, the luminosity distance $d_{\rm L}$ and sky position α (right ascension) and δ (declination), the time and orbital phase at the moment of coalescence $t_{\rm c}$, $\varphi_{\rm c}$, and two angles that define the orientation of the binary ι (inclination) and ψ (polarisation angle).

Adding spin for each binary member increases the dimensionality of the parameter space with three, to twelve for an inspiral with a single spin and fifteen for the case where both spins are fitted. The additional parameters are the dimensionless spin magnitude $a_{\rm spin1,2} \equiv S_{1,2}/M_{1,2}^2$, the angle between spin and orbital angular momentum at the moment of coalescence $\theta_{\rm spin1,2}$ and the precession phase at coalescence $\varphi_{\rm spin1,2}$.

B. Prior distributions

We use prior distributions that are uniform in $\log(d_{\rm L})$ or $d_{\rm L}^3$, $\cos(\theta_{\rm spin1,2})$, $\cos(\iota)$, $\sin(\delta)$, (the sine is used for parameters defined in the domain $[-\frac{\pi}{2}, \frac{\pi}{2}]$, the cosine for those $\in [0, \pi]$) and in the linear scales of the remaining parameters.

The prior ranges for the MCMC parameters that we use depend in part on the value that is provided by the detection trigger, which we will denote by the subscript *tr*, in particular the chirp mass $\mathcal{M}_{\rm tr}$, the time of coalescence $t_{\rm c,tr}$ and the effective luminosity distance $d_{\rm L,eff,tr}$ (for software injections, we will use the injection values instead). Especially for $\mathcal{M}$ and $t_{\rm c}$ it is important not to choose too wide prior ranges in order to constrain the parameter space that needs to be sampled. The default values we use are: $\mathcal{M} \in [0.5\,\mathcal{M}_{\rm tr}, 2.0\,\mathcal{M}_{\rm tr}]$; $\eta \in [0.0, 0.25]$; $t_{\rm c} \in [t_{\rm c,tr} - 50\,{\rm ms}, t_{\rm c,tr} + 50\,{\rm ms}]$; $d_{\rm L} \in [10^{-3}\,{\rm Mpc}, 1.5\,d_{\rm L,eff,tr}]$; $a_{\rm spin1,2} \in [0, 1]$; $\cos(\theta_{\rm spin1,2}) \in [-1, 1]$; $\varphi_{\rm spin1,2} \in [0, 2\pi]$; $\alpha \in [0, 2\pi]$; $\sin(\delta) \in [-1, 1]$; $\cos(\iota) \in [-1, 1]$; $\psi \in [0, \pi]$ and $\varphi_{\rm c} \in [0, 2\pi]$. When we have triggers from multiple interferometers, they usually yield (somewhat) different parameter values. We determine $\mathcal{M}_{\rm tr}$ and $t_{\rm c,tr}$ by taking the mean of the available values, and $d_{\rm L,eff,tr}$ by taking the maximum of the different values.

We use these broad, uniform priors to keep our study general. However, when additional information becomes available (for instance the time and sky location of a gamma-ray burst from an electromagnetic observation), we can restrict our priors accordingly.

C. Starting values for the Markov chains

The purpose of a parameter-estimation analysis is to first find the true source parameters of a GW signal, and then to describe their probability-density functions (PDFs). Although some of the source parameters which will be provided to the MCMC code from the detection trigger will be quite accurate, notably the chirp mass and time of coalescence, information for the other parameters is usually less-well known or even completely unknown. As a consequence, it is impossible to start the Markov chains from "good" initial values, since, even if two or three of the 9–15 parameters have the correct values, this bears little or no significance in the multi-dimensional parameter space in which the chains sample, and these parameters may well drift away from their correct values in the initial part of the analysis.

For each analysis, we usually start multiple, independent chains, which helps to determine convergence (Sect. IV A), quantify sampling quality (Sect. IV B) and select Markov-chain data for postprocessing (Sect. V). In order to do this, it is important that the chains are started from different locations in parameter space. Hence, rather than only giving the separate Markov chains different random-number seeds, we also start them somewhat offset from the trigger values if such information is available, or completely randomly from the prior range if it is not. This ensures that the chains will start exploring the parameter space at a lower likelihood before they find the true modes of the PDFs, and allows us to verify the agreement of the chains on the parameter values and especially on the maximum likelihood value they find.

We draw the starting values for our Markov chains as follows (again using the subscript *tr* to denote values obtained from a search trigger). For the chirp mass we pick a value randomly from a Gaussian distribution with a width of $0.25\,M_\odot$ around $\mathcal{M}_{\rm tr}$ and for the time of coalescence we do the same from a Gaussian distribution of 100 ms width around $t_{\rm c,tr}$.[63] The starting values for each of the other parameters are drawn randomly from a uniform distribution that spans the entire prior range of that parameter (see Sect. III B). Before we accept an initial parameter set, we require that $L > L_{\rm null}$, where $L_{\rm null}$ is the likelihood for the case of a null-signal model, *i.e.* $\tilde{m}(\vec{\lambda}, f) = 0$ in Eq. 9. Because of the way we implemented the likelihood calculation (Eq. 10), this requirement reduces to $L > 1$. As long as this condition is not met, we redraw the initial parameter set. This ensures an efficient start of the chain and hardly ever requires more than a few hundred cheap (no parallel tempering) draws.

When testing our MCMC code by injecting a signal ourselves and simulating an actual parameter-estimation analysis, we carry out a semi-blind analysis and treat the injection values as trigger values for the chirp mass and time of coalescence. Hence, the chains are started from parameter values that are offset from the injection values, rather than from the trigger values for these two parameters, and randomly from the prior for all other parameters, as we do in the case of a real analysis. In this way, we model the information that will be available after a detection trigger has been created by the LIGO-Virgo data-analysis pipeline.

D. Update proposals

The Markov chain is created as follows. If in the current iteration i, the chain has the location in parameter space (set of waveform parameters, or state) $\vec{\lambda}_i$, we propose a random jump $\Delta\vec{\lambda}_i$ to the new location $\vec{\lambda}_{i+1} = \vec{\lambda}_i + \Delta\vec{\lambda}_i$. Since the jump proposal is random, the next state of the chain should depend *only* on the current state, thus giving the chain its Markovian property. We will discuss a number of points below where we deviate from the Markovian rules, and remind the reader here that the Markovian property is a sufficient, but not a necessary condition to obtain an ergodic chain.

We then compute the likelihood for the new state as given by Eq. 12 and determine whether to accept it using a Metropolis algorithm [17, 51]. Thus, we compare the acceptance probability, which is given by the ratios of the *posterior* (the left-hand side in Eq. 18), to a random number r drawn from a uniform distribution between 0 and 1:

$$\frac{p(\vec{\lambda}_{i+1})}{p(\vec{\lambda}_i)} \frac{L(d|\vec{\lambda}_{i+1})}{L(d|\vec{\lambda}_i)} > r. \tag{18}$$

The implicit assumption when using a Metropolis sampler is that the probability of jumping from state $\vec{\lambda}_i$ to state $\vec{\lambda}_{i+1}$ is equally likely as the reverse jump. The jump to state $\vec{\lambda}_{i+1}$ is accepted if Eq. 18 is fulfilled. Otherwise the jump is rejected, the chain keeps the old parameter set $\vec{\lambda}_{i+1} = \vec{\lambda}_i$ and a new iteration is started by drawing a new random jump proposal $\Delta\vec{\lambda}_{i+1}$ to a different state $\vec{\lambda}_{i+2}$. Equation 18 shows that a new state is always accepted when it increases the value of the posterior, and that a larger decrease in this value means a smaller probability of acceptance.

1. Adaptation

We use an *adaptive* scheme for the proposed jump size [52]. The size of the jump proposal for the parameter λ^j (the j-th element of the parameter vector $\vec{\lambda}$) is drawn from a Gaussian distribution with width $\sigma^j_{\rm jump}$. Thus, these widths form a vector $\vec{\sigma}_{\rm jump}$ with the same length as $\vec{\lambda}$. The adaptation of the jump size consists of increasing $\sigma^j_{\rm jump}$ by multiplying it with a factor $f_{\rm acc}$ when a proposed jump in the parameter λ^j is accepted and decreasing $\sigma^j_{\rm jump}$ by multiplying it with a factor $f_{\rm rej}$ when a proposal is rejected. In a typical analysis, we use $f_{\rm rej} = 0.5$ and $f_{\rm acc} = f_{\rm rej}^{\left(\frac{\alpha_{\rm acc}-1}{\alpha_{\rm acc}}\right)}$, where $0 < \alpha_{\rm acc} < 1$ is the target acceptance ratio. So far, we have used $\alpha_{\rm acc} = 0.25$, which (for $f_{\rm rej} = 0.5$) results in $f_{\rm acc} = 8.0$ so that one acceptance and three rejections return the original value for $\sigma^j_{\rm jump}$ and indeed (on average) 25% of the proposals are accepted (as designed with $\alpha_{\rm acc} = 0.25$). We are planning to explore different (especially larger) values for $\alpha_{\rm acc}$.

While this scheme makes the chains non-Markovian in principle, $\vec{\sigma}_{\rm jump}$ quickly reaches an equilibrium value (when averaged over several times $1/\alpha_{\rm acc}$ iterations) and the adaptation should not affect the ergodicity of the chains.

2. Uncorrelated proposals

The default method for choosing a jump proposal is to draw the jump size independently in the different dimensions of the parameter space. This implies that adaptation is done per parameter as well. We make these updates in two categories. The first category contains single-parameter updates, where the likelihood is calculated after proposing a jump in one parameter only, thus deciding whether to accept the jump for each parameter separately. The second category involves proposing a jump in all parameters at once before calculating the likelihood only once (a 'block update'). This is typically done in 10% of the uncorrelated update proposals. For both categories of uncorrelated update proposals the same vector $\vec{\sigma}_{\rm jump}$ is used.

3. Correlated update proposals

There can exist strong correlations between parameters, in which case uncorrelated jump proposals can be very inefficient.[64] We implemented a method to calculate the correlations between the parameters of a block of $N_{\rm corr}$ iterations (typically $N_{\rm corr} \approx 10^3 - 10^4$). We then draw the subsequent $N_{\rm corr}$ jump proposals from a multivariate normal distribution that is given by the Cholesky decomposition of this covariance matrix.

By using the covariance matrix computed from past iterations in order to propose updates, we violate the Markovian property. Typically, however, the matrix is only updated at the beginning of the Markov chain and it is more or less constant during the remainder of the iterations, so that the ergodicity of the chain is not affected.

The correlated update proposals are always block updates of all nine, twelve or fifteen parameters at once, hence there is a separate $\sigma_{\rm jump,\,corr}$ for these updates. In a typical MCMC analysis, we do 70–90% of the update proposals in a correlated way, hence 10–30% in an uncorrelated way.

a. Updating the correlation matrix We recompute the covariance matrix and its Cholesky decomposition at the end of each block of $N_{\rm corr}$ iterations, and decide whether to use the new matrix or not by checking how the diagonal elements of the matrix have changed. This is necessary because the new matrix may not always be a good representation of the true local covariance matrix, for example because the chains happened not to explore the local parameter space very broadly or very well during those $N_{\rm corr}$ iterations. If that is the case, using the new matrix can lead to sampling of an even smaller part of the parameter space, and the chains are prone to get stuck after a few matrix updates. This problem does not arise if we accept the new matrix only if it does not lead to (much) smaller jump sizes. We find that accepting the new matrix only when $\sim 50\%$ of the diagonal elements have *decreased* in value provides the desired result.

E. Parallel tempering

The problem that arises when using MCMC for parameter estimation, and especially to find the (unknown) modes of the PDFs, is that the chains should typically be broad enough to sample the whole allowed prior parameter ranges, while also being able to probe the narrow region of maximum likelihood in a detailed way. These two demands are almost mutually exclusive, but the technique known as *parallel tempering* offers a solution [e.g. 19, 53, 54].

Parallel tempering consists of several ($N_{\rm pch}$) parallel chains that each have a different *'temperature'*. In addition to the default Markov chain with $T = 1$, parallel chains of higher temperature are computed. Hotter chains are more likely to accept a jump that decreases the likelihood, by adjusting Eq. 18 to accept

FIG. 3: An example of parallel tempering for a number of chains of different temperature during the burn-in (see Sect. IV A) of an MCMC analysis. *Top panel:* log Posterior value as a function of iteration number; *Middle panel:* value of the chirp mass vs. iteration number for the same analysis. The different symbols indicate the different temperature chains: red plusses: $T = 1$, green circles: $T = 2.5$, blue crosses: $T = 6.3$, and cyan squares: $T = 15.9$. The vertical dashed line in both panels indicates where the $T = 1$ chain has burnt in, the horizontal dashed line shows the values of the injected signal. The high-temperature chains explore a wide range of parameter values at low Posterior values, while the $T = 1$ chain starts sampling the region of interest in detail after the burn-in. *Bottom panel:* PDFs for the four temperature chains from the first two panels, but generated using Markov-chain data from before and after the burn-in. The narrow peak comes from the $T = 1$ chain, the broad, flat distributions come from the two highest-temperature chains.

a jump when:

$$\frac{p(\vec{\lambda}_{i+1})}{p(\vec{\lambda}_i)}\left(\frac{L(d|\vec{\lambda}_{i+1})}{L(d|\vec{\lambda}_i)}\right)^{\frac{1}{T}} > r, \tag{19}$$

where $T \geq 1$ is the temperature of the chain. (Eq. 19 can be viewed as the definition of the "temperature" T.) The property of more frequently accepting jumps that lower the likelihood allows a hot chain to move around in parameter space more widely than a cooler chain, thus allowing it to discover different modes. Hence, a combination of hot and cool chains can probe both wide parameter ranges and the narrow region(s) of maximum likelihood. In order to do so, the chains must be able to exchange information. This is done by swapping the parameter sets between two parallel chains with $T_m < T_n$ whenever:

$$\left(\frac{L_n}{L_m}\right)^{\frac{1}{T_m}-\frac{1}{T_n}} > r. \tag{20}$$

Since the likelihood that is needed to determine whether to swap the parameter sets was already computed, this decision comes almost for free, and we make it for every pair of chains at every iteration. We start by comparing the lowest-temperature chain with $T_{m=1} = 1$ to all the chains with $T_{n>m}$, and then do the same for $m = 2, \ldots, N_{\rm pch} - 1$[65].

1. Setting up a temperature ladder

The temperature ladder is determined by setting the lower temperature to $T = 1$. This is the only chain that is saved and used to create the PDFs. One also has to choose a maximum temperature $T_{\rm max}$, which is typically the lowest temperature that allows the chain to scatter over the whole allowed parameter ranges quickly. Because a signal with a higher SNR results in more contrast in the parameter space, we need to increase the value of $T_{\rm max}$ when analysing a higher-SNR signal. The last quantity to choose is the number of parallel chains $N_{\rm pch}$ in the temperature ladder. This will be a compromise between high computation speed (low $N_{\rm pch}$) and high swap efficiency for the chains by having small differences between adjacent temperatures (high $N_{\rm pch}$). The temperatures are then chosen equidistantly in $\log(T)$ [19]. Our typical setup is $N_{\rm pch} \approx 7$ and $T_{\rm max} \approx 30 - 50$ for SNRs between 10 and 20. The first line of Table I gives an example of such a temperature ladder.

2. Sinusoidal temperatures

The obvious drawback of parallel tempering is that one has to calculate a handful of chains, instead of just one, increasing the computational power needed by the same factor. In order to reduce the number of chains in the temperature ladder, we experimented with the implementation of *sinusoidal* temperatures for all chains with $T > 1$. In order to do this, we set up our temperature ladder as before, but now sinusoidally oscillate the temperature of each chain $m \neq 1$ with an amplitude ΔT_m. We find that the swapping of chains is efficient when we choose $\Delta T_m = T_m - T_{m-1}$ for each chain $m > 1$, so that the minimum temperature of each chain is equal to the mean temperature of the next cooler chain. Furthermore, we make sure that temperatures of adjacent chains are in antiphase, optimising the overlap at the extrema (maxima where the next higher-temperature chain has minima, and vice versa). In this setup, we can use $N_{\rm pch} \approx 4 - 5$ and $T_{\rm max} \approx 15 - 30$ for SNRs between 10 and 20, thus reducing the computational cost of the MCMC analysis with parallel tempering. An example of such a setup is shown in Table I. We suggest that the period of the temperature variation should not be too close to $N_{\rm corr}$ (see Sect. III D 3) and that a too short period may endanger the ergodicity of the chain. Hence we choose a period for the sinusoidal temperatures that is $\sim 5 \cdot N_{\rm corr}$.

TABLE I: Example setup for sinusoidally varying temperature chains with $N_{\rm pch} = 5$ and $T_{\rm max} = 30$.

m	**1**	**2**	**3**	**4**	**5**
$\mathbf{T_{0,m}}$	1.0	2.3	5.5	12.8	30.0
$\mathbf{\Delta T_m}$	0.0	1.3	3.2	7.3	17.2
$\mathbf{T}_{\rm min,m}$	1.0	1.0	2.3	5.5	12.8
$\mathbf{T}_{\rm max,m}$	1.0	3.6	8.7	20.1	47.2

F. Thinning of the Markov chains

While sampling the parameter space by creating jump proposals and determining whether the proposals should be accepted or not as described above, we may decide to not store every point in the parameter space that was ever visited. Instead, we typically save every n-th state of the Markov chain, where a good value of n is estimated before the analysis begins and the value remains fixed throughout an MCMC run. Lower values are preferred for more complicated analyses, and typical values for n that give good results are 100, 50 or 25 for analyses with a network consisting one, two or three detectors, respectively. This *thinning* of the chains decreases the correlations between subsequent states and reduces the amount of disc space needed to store the data. In principle, more thinning can be done as postprocessing.

IV. VERIFICATION OF MCMC RESULTS

In Section III C, we presented our use of multiple (typically 5–10), independent Markov chains for the analysis of a particular data set, each of which is started from different initial parameters (either offset from trigger values, or drawn completely randomly), and hence a different (and typically low) value for the likelihood. Here, we describe how we use these individual chains to determine the quality of the results of such an analysis. In Sect. IV A, we explain how we determine whether each of the chains has converged and at which iteration this happened, *i.e.* when the *burn-*

in phase ended. In addition, this tells us whether any of the chains did not converge, for instance when it got stuck at some local maximum in likelihood in a particular location of the parameter space, and we should ignore these chains. In the case of our semi-blind analysis, it is easy to recognise such chains, since we can see whether the chains have found the likelihood (and parameter values) of the injected signal. In the case of a real (and completely blind) analysis, consistency between the results of the independent Markov chains can (nearly) ascertain that they have found the highest likelihood present. Finally, this analysis indicates which fraction of the independent chains may have converged, and hence whether we can trust the results in the first place. In Section IV B, we describe how multiple chains allow us to determine whether the chains are sampling properly and efficiently, by comparing the variances between chains with those within each chain using the *mixing* parameter $\hat{R}$. This method verifies the correspondence between the *PDFs* of the different chains rather than their (maximum) likelihoods.

A. Determining burn-in length and convergence from the maximum likelihood

In order to test the convergence of the chains in an automated way, we note that the Markov chains may find a maximum value for the likelihood $\log L_{\max}$ which exceeds the likelihood of the true parameter set $\log L_{\rm true}$. The expectation value of the difference between the two is given by

$$E_{\Delta L} \equiv E[\log L_{\max} - \log L_{\rm true}] = \frac{N_{\rm par}}{2}, \qquad (21)$$

where $N_{\rm par}$ is the number of parameters that is fitted for in the MCMC analysis [19]. For the signals we consider, $E_{\Delta L}$ lies roughly between 4 and 8.

In the results of a given analysis, we find the single maximum likelihood in all the independent chains $\log L_{\rm max,all}$ and then for each individual chain in turn, we find the first iteration where the the likelihood exceeds the threshold value $\log L_{\rm thr} \equiv \log L_{\rm max,all} - E_{\Delta L}$ and call this iteration $\ell_{\rm burn} + 1$. Thus, we have determined the length of the burn-in phase $\ell_{\rm burn}$, which is different for each chain, and we will use the data from each chain starting from iteration $\ell_{\rm burn} + 1$ to do the statistical analysis of the chains. An example for a set of five converging Markov chains can be found in Figure 4.

If the convergence of the chains is bad, we will usually find that only a small number of chains ever reach $\log L > \log L_{\rm thr}$ and hence will not contribute any iterations to the analysis. If the number of 'contributing chains' is 50% or smaller, we will in principle reject the whole analysis as *not converged* and hence *inconclusive*. A converged fraction of 70–80% or more is usually a strong indication (though not a watertight proof!) that the results of the converged chains are valid. Hence, for a typical MCMC simulation using 5 or 10 independent chains, we demand that at least 3 or 6 of them converge, respectively (note that one chain always 'converges'), and would prefer at least 4 or 8 converging chains, in order to trust the results. We would like to stress that this agreement between maximum likelihood values is a necessary, but not a sufficient condition; if the chains do not agree, we have to discard the results, but the method cannot prove that a set of Markov chains converged.

B. Determining convergence from the sampling quality

1. *Potential scale-reduction* (mixing) *factor* $\hat{R}$

An important question in Markov-chain Monte Carlo techniques is that of determining how well the chains have sampled the parameter space, also known as the *mixing* of the chains. In order to do this we compute the *potential scale-reduction factor* (PSRF) $\hat{R}$, as described by Gelman and Rubin [55], but without the correction for the degrees of freedom $df/(df-2)$, as this factor turns out to be both incorrect and minor [56]. When we have m chains with (at least) n iterations after the burn-in each, we can for each parameter λ^j compute the mean of the i-th individual chain and the mean of all chains combined

$$\begin{aligned} {\mu_{\rm ch}}_i^{\,j} &= \frac{1}{n}\sum_{k=1}^{n} \lambda_i^j(k) \\ {\mu_{\rm all}}^{j} &= \frac{1}{m}\sum_{i=1}^{m} {\mu_{\rm ch}}_i^{\,j}. \end{aligned}$$

We then compute the variances of the data points for each chain and the variance within the m values of ${\mu_{\rm ch}}_i^{\,j}$

$$\begin{aligned} {\sigma^2_{\rm ch}}^{\,j} &= \frac{1}{m(n-1)}\sum_{i=1}^{m}\sum_{k=1}^{n}\left(\lambda_i^j(k) - {\mu_{\rm ch}}_i^{\,j}\right)^2, \\ {\sigma^2_{\mu_{\rm ch}}}^{\,j} &= \frac{1}{m-1}\sum_{i=1}^{m}\left({\mu_{\rm ch}}_i^{\,j} - {\mu_{\rm all}}^{j}\right)^2. \end{aligned}$$

Finally, we compute the PSRF

$$\hat{R}^j = \frac{n-1}{n} + \frac{{\sigma^2_{\mu_{\rm ch}}}^{\,j}}{{\sigma^2_{\rm ch}}^{\,j}}\left(\frac{m+1}{m}\right), \qquad (22)$$

which expresses how much further one would expect the PDFs to expand before the sampling of the chains is perfect. Hence in practice $\hat{R}^j > 1$ and the closer it is to unity, the better is the sampling. We find that when strong correlations are present, it is harder (i.e. it requires much longer chains) to get $\hat{R}^j$ close to unity. Usually, we are satisfied with $\hat{R}^j \lesssim 1.2$ for most parameters. We also find that when distributions are multimodal, such as the sky position when using data from three non-colocated interferometers, $\hat{R}^j$ values may be somewhat higher while the mixing in each mode looks reasonably good, though the chains make many more jumps within a mode than between modes. In order to quantify the sampling of an entire analysis with one number, we use $\hat{R}$, defined as the median

FIG. 4: Example of convergence for five different Markov chains (indicated by the different symbols/colours) analysing the same data set, but starting from different offset starting values. *Top panel:* logarithm of the posterior as a function of iteration number. The horizontal dotted lines at low posterior indicate the starting values of the chains. The two horizontal dash-dotted lines at high posterior indicate the highest likelihood value found in all chains ($L_{\max}$; top line and solid star) and $L_{\max} - N_{\rm par}/2$ (bottom line). Each of the chains is assumed to have burnt in when they first cross the lower dash-dotted line, and the iteration at which this happens is indicated by a vertical dashed line. *Bottom panel:* value of the chirp mass as a function of iteration number for the same five chains. The symbols of the different chains are the same as in the top panel, the vertical dashed lines again indicate the point where the burn-in ends. The horizontal dotted lines show the starting values for each chain, the horizontal dash-dotted line indicates the true value of the chirp mass of the analysed signal.

of the values $\hat{R}^j, j = 1, \ldots, N_{\rm par}$ for all parameters. Examples of chains that are mixing well and chains that are mixing badly, as well as their $\hat{R}$ values, are shown in Fig. 5.

The disadvantage of $\hat{R}$ is that we cannot define a hard limit below which $\hat{R}$ must lie so that we will trust the results. As with other indicators, this parameter merely helps us to judge the quality of a set of Markov chains and the validity of their implications.

However, if the value of $\hat{R}$ is reasonably small, we can multiply the width of the PDF (e.g. the standard deviation or some probability range) with $\hat{R}$ to estimate the true accuracy, according to its design.[66]

2. *Autocorrelation length*

Another diagnostic that may be used to measure how well the Markov chains are sampling is the autocorrelation. We compute the *normalised* value for this diagnostic for parameter λ_p and an interval $m < N$ as:

$$\rho_m = \frac{1}{(N-m)\sigma_p^2} \sum_{i=1}^{N-m} (\lambda_p(i) - \mu_p) \cdot (\lambda_p(i+m) - \mu_p)\,, \tag{23}$$

where N is the number of data points in the chain, μ_p is the mean value and σ_p^2 the standard deviation of the parameter λ_p. The autocorrelation is determined separately for each independent Markov chain and for each parameter $\lambda_p \in \vec{\lambda}$.

As a diagnostic, we are particularly interested in the autocorrelation *length* $\ell_{\rm ac}$, which we define as the value of m where $\rho_m < 0$ for the first time. In order to characterise the typical autocorrelation length for the whole set of Markov chains (*i.e.*, all parameters of all independent chains) we first compute the median of the autocorrelation lengths for all parameters $\lambda_p \in \vec{\lambda}$ of each chain, which gives us the typical length for that chain. Subsequently, we compute the median of these (typically 5–10) median values to produce the typical autocorrelation length for the analysis, $\ell_{\rm ac}$, expressed as a number of iterations in our Markov chains.

The autocorrelation length is useful to verify the validity and efficiency of a parameter-estimation analysis. First, in order to have meaningful results, we need Markov chains that are many autocorrelation lengths long, *i.e.* $\ell_{\rm ac} \ll N$. Secondly, when using correlated update proposals (see Sect. III D 3), we need to ascertain that $N_{\rm corr} < \ell_{\rm ac}$. And thirdly, when testing different methods of sampling, a smaller value for $\ell_{\rm ac}$ indicates more efficient sampling, or better *mixing*.

FIG. 5: Examples of mixing for two different MCMC analyses. The panels show the Markov chains (parameter value vs. iteration number) for a selection of six parameters; different colours represent different independent chains running on the same data set. The horizontal dotted lines show the starting values of the different chains, the dashed vertical lines show the iteration number at which each chain converges. The values of $\hat{R}$ for each parameter are shown above the panels (as R-hat). *Left:* Analysis with relatively good mixing (median $\hat{R}$ value in all parameters ≈ 1.10). *Right:* Analysis with relatively poor mixing (median $\hat{R}$ value in all parameters ≈ 1.86). See Sect. IV B 1 for more details.

V. REPRESENTATION AND INTERPRETATION OF THE RESULTS

After a parameter-estimation analysis has completed, or while the analysis is ongoing and we consider a "snapshot" of the intermediate output up to that point, we need to convert the output from our MCMC code, *i.e.* the Markov chains, to a form that can be easily interpreted. For this we use a code called AnalyseMCMC [57]. We first determine the burn-in length, *i.e.*, $\ell_{\rm burn}$ for each Markov chain, as described in Sect. IV A. This provides us with a criterion to judge whether the Markov chains have *converged*, and hence whether the whole analysis should be accepted, or not. Secondly, for the remainder of our analysis, we take into account the part of each Markov chain after $\ell_{\rm burn}$. The data from the converged parts of the chains are then combined and treated as a single data set. This data set is the raw result of our analysis; it contains the full $N_{\rm par}$-dimensional posterior probability-density function (posterior PDF).

Since it is difficult to visualise data in a parameter space that has many dimensions, we typically produce one- and two-dimensional *marginalised* posterior PDFs, which allow for a quick interpretation. The two-dimensional PDFs are useful for combinations of parameters that have strong correlations, *e.g.* the two individual masses, the sky position and the binary orientation.

In order to quantify our results, we typically want to express a "best value" and an uncertainty in or accuracy of that value in the marginalised PDFs for each of the parameters. One choice of "best value" and accuracy would be the median (or mean) and the standard deviation. Alternatively, we can use *probability intervals* of the 1D PDFs to express both the "best value" for each parameter (the centre of the interval) and the accuracy (the width of the interval). This is described in Sect. V A.

For 2D PDFs, in order to express the accuracy of *e.g.* the sky position, we compute *probability areas*, as discussed in Sect. V B. In addition, we determine the cross-correlation matrix of the results (Sect. V C). This is particularly useful to find for which parameters the 1D accuracy determinations *overestimate* the amount of uncertainty. For example, Fig. 6 displays a 2D marginalised PDF of the two individual component masses of a binary. When 1D accuracy estimates are combined, the total uncertainty in these two parameters would seem to cover a large rectangle in this

FIG. 6: A two-dimensional (2D) posterior PDF for the component masses of a binary (the signal analysed is NS-0 described in Sect. VI B and Table III). The three different shades/colours show the 1σ, 2σ and 3σ probability areas as indicated at the top of the figure and determined using a *greedy-binning* algorithm (see Sect. V B). The dash-dotted (black) lines and star indicate the true parameter values of the signal ($11\,M_\odot$ and $7\,M_\odot$), the dashed (red) lines show the medians from the one-dimensional (1D) distributions. The (red) arrows and dotted lines show the 1D 2σ (95%) probability intervals. Judging from these intervals, the uncertainty in this plane would amount to $4.1\,M_\odot^2$, but the 2D probability area indicates that the actual uncertainty is $0.42\,M_\odot^2$, nearly an order of magnitude smaller. This difference is extreme, because the two component masses are strongly correlated, with a normalised correlation of $\sigma_{M_1,M_2} = -0.99$ (see Sect. V C).

figure, whereas in reality the PDF is restricted to a relatively narrow, sickle-shaped area.

Finally, the *Bayes factor* expresses how well the assumed model matches the data set. This is a useful parameter to compare analyses that use different models, for example a model that allows for spinning objects, a model that does not and a model that allows for noise only, and determine which model describes the data set better, and to what extent. We describe the computation of the Bayes factor from MCMC output in Sect. V D.

A. 1D marginalised PDFs and probability intervals

Marginalising the full $N_{\rm par}$-dimensional PDF over all parameters $\vec{\lambda}$ but one (λ_p) in order to obtain the 1D marginalised PDF, is trivial for Markov-chain Monte Carlo output. In fact, for each iteration, our code stores the values of the parameters at which the chain resides at that iteration, and we can simply consider only the output for one of these parameters.

1. Best value and accuracy

In order to compare results from different analyses, it is convenient to express the outcome of the parameter estimation in just a few numbers — in particular a "best value" and uncertainty in that value for each parameter. It should be noted here that by describing a 1D marginalised PDF by only two numbers, some of the details of the PDF are inevitably lost (just as details of the full n-dimensional PDF are lost by expressing it as n 1D marginalised PDFs). However, while keeping this in mind, reducing the results to only a few numbers gives us a better quantitative "feel" of the results.

There are two choices of parameters to express the "best value" and accuracy for each parameter that we regularly use. One choice is the the median of the 1D marginalised distribution for the "best value" and the standard deviation for the accuracy. Alternatively, we can use the centre of the 1D probability interval for each parameter as a measure of the "best value" for that parameter and the width of that interval as the accuracy with which the parameter was estimated. The expressions for both choices are determined by sorting the N data points of λ_p to increasing value.

The median $\mu(\lambda_p)$ is determined in the usual way, by picking the value of the $(\frac{N+1}{2})$-th number in this sorted list ($\lambda_p(\frac{N+1}{2})$) if N is odd, or the arithmetic mean of $\lambda_p(\frac{N}{2})$ and $\lambda_p(\frac{N}{2}+1)$ if N is even. The median is a good statistic for the "best estimate" for the value of λ_p and is less affected by distant outliers than is the mean.

In order to express the uncertainty in the parameter estimation, we determine a probability range for each parameter λ_p. As a default, we consider the 95.4%-probability interval (or "2σ" interval), defined as the narrowest range that contains 95.4% of our data points. This range can be obtained easily from the sorted list of data points, by considering the distance in coordinate space $\lambda_p(N_2) - \lambda_p(N_1)$ between the points $N_1 = 1$ and $N_2 = 0.954 \cdot N$, $N_1 = 2$ and $N_2 = 0.954 \cdot N + 1$, etc., and finding the pair of N_1, N_2 that give the smallest difference. The probability interval thus obtained can be used to express the uncertainty around the "best value" determined by the median, or can even produce both by quoting the central value of the interval $(\lambda_p(N_2) - \lambda_p(N_1))/2$ as the "best value". For periodic parameters, such as the phases or right ascension, we test all possibilities wrapping around the value 2π (π for the polarisation angle ψ), and the central value of the probability interval is used to centre the 1D and 2D histograms (see below).

2. Histograms of 1D PDFs

A marginalised 1D PDF is constructed by binning the data points of the parameter λ_p and creating a histogram. Such a histogram has the value of the parameter λ_p on its horizontal axis (see Fig. 7 for an example) and, since the MCMC samples according to the posterior, a value that is proportional to the pos-

FIG. 7: Example results from the MCMC analysis of a simulated signal from an inspiralling BH-BH binary with masses of 11 $M_\odot$ and 7 $M_\odot$, and spins of 0.9 and 0.7 respectively (signal HS-2, described in Sect. VI B). The grey-shaded histograms represent the 1D marginalised PDFs for each of the 15 parameters that describe the source binary. Vertical dash-dotted (black) lines indicate the parameter values of the injected signal, vertical dashed (red) lines show the median in each parameter ("best parameter value") and the two vertical dotted (red) lines indicate the 95% ("2σ") probability range ("uncertainty in the parameter value"), the value of which is indicated at the top of each panel.

terior probability density on the vertical axis. The number of bins can be chosen manually, but we find that in general good results are obtained when determining the number of bins $N_{\rm bin}$ from the number of data points $N \gtrsim 100$ using

$$N_{\rm bin} = 10 \times {}^{10}\log N. \tag{24}$$

Typically, we use around 50 bins for our 1D histograms.

Although the most important quantitative results can be derived without creating such a histogram, it provides important information at a glance, such as whether a distribution is multimodal or not, whether the distribution 'rails' against the prior boundaries and the general quality of the analysis (chains that get 'stuck' at a certain parameter value produce a narrow peak, independent Markov chains that do not *mix* well (see Sect. IV B) produce many separate peaks). In many cases, we apply minimal smoothing to the 1D histograms before we present them, sufficient to smear out small-scale features in the PDFs, but not so much as to change the qualitative interpretation of the PDFs. This smoothing does not affect our quantitative results either, since they are determined independently.

B. 2D marginalised PDFs and probability areas

Some combinations of parameters are heavily correlated, for example the chirp mass and spin magnitude (see Sect. V C), or are useful to present together, such as the right ascension and declination of a source in a sky map. For such cases, we compute two-dimensional marginalised posterior PDFs in order to determine the "best value" or accuracy of parameter estimation, or to visualise the results in a figure. The main difference between the one- and two-dimensional analyses is that in the latter case, we cannot create sorted lists of data points and we need to bin our data first. Hence, the quantitative results obtained for the 2D case depend on the way the bins are constructed, although experience shows that this dependence is generally weak for reasonable choices of the number of bins used. Nevertheless, we usually refrain from determining "best values" from 2D results, as 1D medians are much more robust and reliable.

As in the 1D case, marginalisation of the full $N_{\rm par}$-dimensional PDF over all but two parameters is trivial and can be done by considering only the MCMC output for the two selected parameters. The number of bins in each direction can be determined manually, or automatically from the number of data points, using the prescription from the 1D case in Eq. 24. When

producing a rectangular figure, we typically increase the number of bins on the longer axis in order to keep the 2D bins, or pixels, square. Hence, we typically use roughly 50×50 pixels, although the precise number may depend on the modality of or the strength of correlations in the distribution.

After the data have been binned, we use a so-called *greedy algorithm* to define the probability areas, the 2D equivalent of the probability intervals defined in Sect. V A. Using, as an example, the 95.4% probability area, we select the bin that contains the largest number of data points, then the bin with the second-largest number of data points, and so forth. After each of these subsequent pixels are selected, we add up their data points and compute the fraction of the total number of data points N that is contained in all selected bins so far. Once this fraction reaches $0.954 \cdot N$, we stop selecting bins. By adding up the area that all these pixels cover, we find the probability area we were looking for: the smallest area that contains a certain fraction of our data points.

Experience shows that the greedy algorithm generally provides nice coherent areas, even though these areas may not be connected (see Fig. 8). In fact, this is exactly what we need — for the general case of sky localisation with three non-colocated interferometers, for example, we expect two areas in the sky where the source may be. The only slight disadvantage of this method lies in quantifying the probability areas, which depends on the choice of binning. In fact, the area found generally decreases slowly as the pixel size decreases, *i.e.*, as the number of pixels increases. Consider, for example, a sky map with N data points. At one extreme, we could cover the sky with only one pixel, which would evidently need to be selected in order to cover the majority of our data points, and the probability area found would be that of the whole sky or 4π steradians. At the other extreme, we could cover the sky with an infinite number of (infinitesimally small) pixels. In that case, the total area found would be N times an infinitesimally small area, or zero steradians.

C. Cross-correlations between the parameters

We compute the *normalised* cross-correlations between parameters λ_p and λ_q as:

$$\sigma_{p,q} = \frac{1}{N-1} \sum_{i=1}^{N} \frac{\lambda_p(i) - \mu_p}{\sigma_p} \cdot \frac{\lambda_q(i) - \mu_q}{\sigma_q}, \qquad (25)$$

where i is the Markov-chain iteration number, N is the total number of iterations and μ_p and σ_p are the mean and variance of the parameter λ_p, respectively. In this definition, $-1 \leq \sigma_{p,q} \leq 1$, where $|\sigma_{p,q}| \approx 1$ indicates a strong (anti)correlation, and $\sigma_{p,q} \approx 0$ indicates a weak correlation between the two parameters.

The correlations provide important information for the interpretation of the one-dimensional accuracies for strongly correlated parameters (see the introduction of Sect. V and Figures 6 and 9) and give us a qualitative idea of some of the behaviour in the sampling efficiency of our Markov chains. In some cases, these correlations are predictable and can help us to design specialised update proposals for certain combinations of parameters. Examples of parameters that often have strong correlations are the chirp mass with the mass ratio, spin magnitude and spin tilt, the distance with the inclination and the time of coalescence or distance with either of the two sky coordinates. In many cases, the strength of the correlations depends on the exact details of the signal.

D. Bayes factor

Although Markov-chain Monte Carlo is primarily concerned with estimating posterior PDFs, unlike other Bayesian inference techniques that are designed to compute evidences, such as nested sampling [*e.g.* 58, 59], the output from SPINSPIRAL can also be used for evidence computation.

When we consider our definition of Bayes' theorem in Eq. 8 and marginalise over the whole parameter set $\vec{\lambda}$ for a *specific* model M_m, we obtain

$$p(M_m|d) = \frac{p(M_m)\, p(d|M_m)}{p(d)} \qquad (26)$$

and we can compare two models M_m and M_n by computing the *odds ratio*:

$$\begin{aligned} O_{m,n} &= \frac{p(M_m|d)}{p(M_n|d)} = \frac{p(M_m)\, p(d|M_m)}{p(M_n)\, p(d|M_n)} \\ &= \frac{p(M_m)}{p(M_n)} B_{m,n}, \end{aligned} \qquad (27)$$

where

$$B_{m,n} = \frac{p(d|M_m)}{p(d|M_n)} \qquad (28)$$

is the *Bayes factor* of the two models, given by the ratio of the *evidences* for each model.

We follow Raymond et al. [25] [and hence 60] in computing an approximation for the evidence

$$p(d|M_m) \approx N\, V_\mathrm{t} \left[\sum_{i=1}^{N} \frac{1}{p(\vec{\lambda}_i|M_m)\, p(d|\vec{\lambda}_i, M_m)} \right]^{-1}, \qquad (29)$$

which is the harmonic mean of the posterior values sampled by the MCMC, multiplied with N, the number of data points in our Markov chains, and V_t, an approximation of the volume of the whole parameter space given by

$$V_\mathrm{t} \approx \sum_{i=1}^{N} V_{\vec{\lambda}_i} = \sum_{i=1}^{N} \frac{\alpha_p}{p(\vec{\lambda}_i|M_m)\, p(d|\vec{\lambda}_i, M_m)}. \qquad (30)$$

The factor α_p is a proportionality constant.

This allows us to compute the Bayes factor between models M_m and M_n using Eq.28 as

$$\begin{aligned} B_{m,n} &= \frac{N_m}{N_n} \frac{\sum_{k=1}^{N_n} p(\vec{\lambda}_i|M_n)\, p(d|\vec{\lambda}_i, M_n)}{\sum_{k=1}^{N_m} p(\vec{\lambda}_i|M_m)\, p(d|\vec{\lambda}_i, M_m)} \\ &= \frac{N_m}{N_n} \frac{\sum_{k=1}^{N_n} L_n(\vec{\lambda}_k)}{\sum_{k=1}^{N_m} L_m(\vec{\lambda}_k)} \end{aligned} \qquad (31)$$

FIG. 8: Two-dimensional posterior PDF for the sky position of a binary inspiral with two spinning black holes (signal HS-2, described in Sect. VI B). Because the data from three non-colocated detectors were used for the analysis, the PDF shows possible sky positions at the true location of the source (indicated by the symbol) and at a location that is the true position, mirrored in the plane spanned by the three detectors. Because of the extra information contained in the waveform due to the spins, the mirror location has a much lower probability density and is almost ruled out. The PDF is mapped using a Mollweide projection.

FIG. 9: Two-dimensional posterior PDF for the luminosity distance and inclination sky position of a binary inspiral with two spinning black holes (signal HS-2, described in Sect. VI B). Because of the precession induced by the black-hole spins, extra information is available to constrain the inclination. In this example, virtually all probability density is at $\iota < 90^\circ$. The true parameter values of the signal are indicated by the black dash-dotted lines and the symbol. As may be expected, the distance and inclination are strongly correlated, with a normalised correlation of $\sigma_{d_\mathrm{L},\iota} = -0.78$ (see Sect. V C).

where the last step uses the proportionality from Eq. 8. Hence, to compute the Bayes factor or odds ratio from our MCMC output, we need to compute the mean of all the likelihood values from the $T = 1$ Markov chains of each of our MCMC runs using the models M_m and M_n.

Note that the harmonic mean in Eq. 29, and hence the value of the Bayes factor, is very sensitive to data points with a low posterior, a region of the parameter space which an MCMC code is not designed to sample well. Hence, the value of the Bayes factor depends strongly on the quality of our sampling and is typically overestimated somewhat.

VI. EXAMPLE MCMC SIMULATIONS

A. Sampling efficiency

In this section we compare the MCMC sampling efficiency for the parameter-estimation analyses of three different simulated data sets, each containing a different CBC signal, using seven different combinations of settings for the MCMC code. The results of this exercise are shown in Table II.

The left-hand side of the table shows the combination of sampling techniques which are switched on or off for each of the seven different analyses. These features, described in Sections III D and III E, are: adaptive updates (code settings, column 2), correlated update proposals (column 3), parallel tempering (column 4) and sinusoidal temperatures when using parallel tempering (code settings, column 5). The top part of the table shows the effect of gradually switching on features, the bottom part shows the effect of alternatingly switching on all features but one (repeating two lines from the top part).

We measure the efficiency of the sampling for each of the three signals in the next three parts of Table II. We use four indicators to quantify the quality

TABLE II: Sampling efficiencies for the analyses of three different signals, using a variety of code settings (see the text in Sect. VI A).

Code settings:					Results signal 1:				Results signal 2:				Results signal 3:			
Sim. nr.	**adapt. upd.**	**corr. prop.**	**par. temp.**	**sin. temp.**	$\mathbf{N_{ch}}$	$\ell_{\rm burn}$ (10^4)	$\ell_{\rm ac}$ (10^4)	$\hat{\mathbf{R}}$	$\mathbf{N_{ch}}$	$\ell_{\rm burn}$ (10^4)	$\ell_{\rm ac}$ (10^4)	$\hat{\mathbf{R}}$	$\mathbf{N_{ch}}$	$\ell_{\rm burn}$ (10^4)	$\ell_{\rm ac}$ (10^4)	$\hat{\mathbf{R}}$
1	off	off	off	—	5	30	31	2.08	5	19.3	17	1.42	5	9.1	17	1.98
2	**on**	off	off	—	1	—	24	—	5	15.5	18	1.57	5	3.6	13	1.06
3	**on**	**on**	off	—	1	—	16	—	5	2.1	18	2.05	4	2.0	18	1.42
4	**on**	**on**	**on**	off	5	2.1	21	1.49	5	0.9	10	1.03	5	0.7	10	1.10
5	**on**	**on**	**on**	**on**	5	3.3	19	1.19	5	1.9	10	1.05	5	1.1	11	1.05
6	off	**on**	**on**	**on**	5	4.8	16	1.22	5	0.9	8.0	1.07	5	0.8	8.0	1.04
7	**on**	off	**on**	**on**	5	4.9	13	1.61	5	1.1	8.0	1.05	5	2.3	3.0	1.05
(3)	**on**	**on**	off	—	1	—	16	—	5	2.1	18	2.05	4	2.0	18	1.42
(4)	**on**	**on**	**on**	off	5	2.1	21	1.49	5	0.9	10	1.03	5	0.7	10	1.10

of sampling: the number of converged chains ($N_{\rm ch}$; see Sect. IV A), the length of the burn-in ($\ell_{\rm burn}$; see Sect. IV A), the autocorrelation length of the chains ($\ell_{\rm ac}$; Sect. IV B 2) and the potential scale-reduction factor ($\hat{R}$; Sect. IV B 1).

Signal 1 is that of a $10 + 1.4\,M_\odot$ BH-NS system with spin for the BH only ($a_{\rm spin1} = 0.5$), using a 1.5-pN waveform with simple precession (Sect. II B 1), and 7 s of data. **Signal 2** comes from a $11 + 7\,M_\odot$ BH-BH inspiral with spin for the most massive BH ($a_{\rm spin1} = 0.5$), also using a 1.5-pN waveform and simple precession, but analysing 3 s of data. Finally, **signal 3** is generated by the inspiral of a $10 + 1.4\,M_\odot$ BH-NS binary without spins, using a 2.0-pN GeneratePPN waveform (Sect. II B 3) and 7 s of data. For all analyses, we use the data from one detector and Gaussian, simulated noise. Each of the three signals is injected with an SNR of 15 by scaling the source's distance. For each of the different analyses of a given signal, five independent MCMC chains have been computed for 10^6 iterations each, and the starting values for the MCMC chains are identical between the different analyses.

The results in Table II indicate the following:

Adaptive update proposals have a mildly negative effect on the autocorrelation lengths of the chains (compare lines 1 to 2, and 6 to 5 in the Table), while being neutral to the other indicators. The effect of adaptive proposals seems to be more favourable when parallel tempering is not used. This seems to be a reason *not* to use them, especially when using parallel tempering.

Correlated update proposals have a negative effect on the autocorrelation, but mildly positive effects on the burn-in length and the mixing ($\hat{R}$; in the Table, compare line 2 to 3 and 7 to 5). Since the method may be a threat to the ergodicity of the chains, and little sampling efficiency is gained from it, we conclude that it can be used in the burn-in to optimise the convergence of the chains, but should preferably not be used afterwards.[67]

Parallel tempering has a strongly positive effect on the sampling efficiency according to all four indicators (lines 3 and 4 in the Table). The absence of parallel tempering causes fewer chains to converge in some cases, and even causes some analyses to fail to converge at all. However, we have not taken into account that an analysis using parallel tempering needs about five times as much CPU time, since typically five different parallel chains, with a different temperature each, are computed. While it is not straightforward to compensate for this in the comparison (especially if some MCMC chains do not converge at all), we point out that the parallel tempering is a natural point where an MCMC code can be parallelised, by computing the waveforms needed for the jump proposals on different CPUs for chains of different temperatures.[68] In cases where the clock time to obtain a result is more important than CPU time, parallel tempering should be used in a parallelised code.

Sinusoidal temperatures of the $T > 1$ chains in parallel tempering hamper the convergence during the burn-in appreciably, have a neutral effect on the autocorrelation and improve the mixing efficiency noticeably (compare line 4 to 5 in the Table). Hence, they should be used after the burn-in.[69]

In conclusion, there seems to be no need to use adaptive update proposals, correlated update proposals should only be used in the burn-in, parallel tempering has a positive effect, and sinusoidal temperatures should be used after the burn-in.

B. Accuracy of parameter estimation

1. *Description of the signals*

In order to give the reader an impression of the accuracies that can be achieved in extracting astrophysical parameters from GW inspiral signals as measured by LIGO and Virgo, and of the effect of the presence of spins on this accuracy, we here present the analyses of three more simulated example signals. All three signals are computed for the inspiral of a BH-BH system

(for example accuracies for BH-NS inspirals, see van der Sluys et al. [22]), with masses of $M_1 = 11.0\,M_\odot$ and $M_2 = 7.0\,M_\odot$ ($\mathcal{M} \approx 7.6\,M_\odot$, $\eta \approx 0.238$). The differences are in the spin magnitude; the first signal has no spins (NS; $a_{\rm spin1} = 0.0$, $a_{\rm spin2} = 0.0$); the second signal has low spins (LS; $a_{\rm spin1} = 0.2$, $a_{\rm spin2} = 0.1$); and the third signal has high spins (HS; $a_{\rm spin1} = 0.9$, $a_{\rm spin2} = 0.7$). The tilts between the spin and orbital vectors are $\theta_{\rm spin1} = 20°$ and $\theta_{\rm spin2} = 10°$, and the distances of the three sources (53.3 Mpc for NS, 59.8 Mpc for LS and 74.5 Mpc for HS) are chosen such that the total signal SNR is always 15.0. We analyse 4 s of data from the 4-km LIGO detectors in Hanford and Livingston, and the 3-km Virgo detector. The data sets consist of a simulated signal injected into Gaussian noise with the spectra of Initial LIGO and Virgo. Both for the signal injection and for the MCMC analysis, we use the SpinTaylor 3.5-pN double-spin waveform (see Sect. II B 2).

2. Description of the analyses

Each signal is analysed twice — once while allowing for two spins in the MCMC (using 15 parameters; NS-2, LS-2 and HS-2) and once assuming there is no spin in the black holes ($a_{\rm spin1,2} = 0$, using 9 parameters; NS-0, LS-0 and HS-0). Each analysis consisted of nine independent Markov chains when the spins were fitted for and five such chains for the cases where the spin was fixed. In each case, each independent chain used five parallel-tempering chains with a maximum temperature of 40 and sinusoidal temperature variations. The results of the analyses where spin is allowed for can be found in the form of 1D marginalised posterior PDFs in Figure 7 (HS-2). In Figure 11 (HS-0 vs. HS-2) we compare the 1D posterior PDFs for a selection of parameters between the cases where spins are fit for and those where they are not. A selection of 2D posterior PDFs can be found in Figures 6 ($M_1 - M_2$ for NS-0), 8 (sky map for HS-2) and 9 (distance–inclination for HS-2). The accuracies for all six analyses are summarised in Table III, for all binary parameters, except right ascension and declination, which are combined in the 2D position. The accuracies for the individual mass components have been added; these values were computed from the data for the chirp mass $\mathcal{M}$ and mass ratio η.

3. Discussion of the results

a. Masses When looking at the results of the analyses that fit for spins (the first three lines), Table III shows that the uncertainties for the chirp mass *increase* when the spin increases. The mass ratio, η, shows the opposite effect — its *accuracy* increases with increasing spin. The latter effect is slightly stronger when considering the individual masses; they are determined more accurately for the case where spins are stronger. This effect is opposite to that of the case of a BH-NS signal found in an earlier paper, where the determination of the chirp mass becomes *more accurate* for a more rapidly spinning black hole [22, note that 90%-probability ranges are used in that paper]. The BH-NS analyses in that study have a much more accurate chirp-mass determination (2.6% for signals with a non-spinning BH, down to 0.6–0.9% for the signals with spin) and a less accurate determination of the mass ratio (0.13 for non-spinning signals, 0.04–0.08 for signals with spin). The combination of these trends in chirp mass and mass ratio translate into a somewhat better determination of the individual masses for the case of strong spins. The estimations of the component masses for the BH-BH systems are comparable to, and perhaps slightly less accurate than in the case of the BH-NS systems.

While the SNRs of the signals are comparable between that study (SNR=17) and the current one (SNR=15), the main differences, which are probably responsible for the differences in accuracies, are the mass ratios ($q \approx 0.14; \eta \approx 0.108$ vs. $q \approx 0.64; \eta \approx 0.238$ here) and the (related) fact that the binary inspirals in the current study have two significant spins as opposed to only one (for the BH in the BH-NS inspirals). These two facts are related since the steeper mass ratio in the BH-NS system ensures that the NS spin would not affect the results strongly, if at all, even if the signals in van der Sluys et al. [22] had had a non-zero NS spin. On the other hand, the parameter estimation in the earlier study did not allow for a finite spin for the NS, restricting the number of parameters to 12. Allowing for a non-zero NS spin might have increased the uncertainties for the chirp mass in the earlier study, as it does in the current one (compare the top half of Table III to the bottom half).

b. Position and orientation The behaviour of the uncertainties in distance in Table III is similar to that of the individual masses: while they are relatively large for the case with a non-spinning BH and slightly less so for low-spin case, the distance for the signal with high spins is significantly better determined. This is connected to the diminished probability for one of the two possible sky locations of the source; timing alone allows for two solutions in the sky position when observing with three (non-colocated) detectors: the true position and that position, mirrored in the plane defined by the three detectors (see Fig. 8). However, the presence of spin causes precession of the binary orbit, which changes the inclination and polarisation angle of the source during the observation, excluding a significant fraction of the possible phase space in the binary orientation, sky position, arrival time and distance, which are strongly correlated (see Fig. 12). For signals without spin, or with low spin, PDFs are therefore often bimodal in these parameters — a bimodality which can be lifted by the presence of spin and the resulting orbital precession. This is illustrated in Fig. 10, where 1D PDFs for the LS-2 and HS-2 analyses are compared. The secondary mode has almost completely disappeared in the latter signal. Interestingly, while for the sky location and time of arrival the main difference is the disappearance of the second mode, for the distance and inclination the remaining mode also seems to widen. This is probably due to the difference in source distance for the two signals

TABLE III: Parameter-estimation accuracies (95%) for the example analyses described in Sect. VI B 3.

ID	$\mathbf{d_L}$	$\mathcal{M}$	η	$\mathbf{M_1}$	$\mathbf{M_2}$	$\mathbf{d_L}$	$\mathbf{t_c}$	$\mathbf{a_{s1}}$	θ_{s1}	φ_{s1}	$\mathbf{a_{s2}}$	θ_{s2}	φ_{s2}	**Pos.**	ψ	ι	φ_{orb}
	(Mpc)	**(%)**		**(%)**	**(%)**	**(%)**	**(ms)**		(°)	(°)		(°)	(°)	($^{\circ 2}$)	(°)	(°)	(°)
NS-2	53.3	4.1	0.064	51	45	125	32	0.45	145	337	0.91	142	323	43.9	132	76	334
LS-2	59.8	4.7	0.058	51	43	116	32	0.57	141	334	0.89	143	320	57.8	115	51	335
HS-2	74.5	9.9	0.041	41	40	79	5.2	0.86	118	326	0.57	115	291	46.9	165	64	341
NS-0	53.3	2.4	0.047	33	34	134	32	–	–	–	–	–	–	32.6	141	85	333
LS-0	59.8	2.0	0.039	34	34	122	32	–	–	–	–	–	–	40.0	114	60	327
HS-0	74.5	3.7	0.069	54	54	135	32	–	–	–	–	–	–	54.9	158	94	342

FIG. 10: A comparison of one-dimensional marginalised posterior PDFs for the signals with low spins (LS-2; upward hashes, red) and the signal with high spins (HS-2; downward hashes, blue), fitting for both spins, showing selected position and orientation parameters. The PDFs are normalised in area, hence narrower PDFs have higher peaks. The dashed vertical lines indicate the parameter values of the injected signal (two different values for the distance). For most parameters, the LS-2 PDFs show two modes, which are reduced to one for the HS-2 case. See Sect. VI B 3 for a discussion.

(applied in order to normalise the SNR). In the case of the distance, the *relative* uncertainty still drops significantly, while for the inclination, where the relative accuracy has little meaning, the *absolute* accuracy is quoted in Table III, which increases.

c. Spin Finally, the accuracies in the spin parameters show erratic behaviour; while the uncertainty in $a_{\rm spin1}$ grows with the strength of the spin, that in $a_{\rm spin2}$ decreases. This may not be easy to explain, as the spins are correlated to the masses and the effects of the mass-spin system are quite complex and should be considered as a whole. This calls for a more elaborate study than is the scope of the current paper. The angle between spin and orbit is virtually undetermined for zero or small spin, as one would expect. While for larger spins these parameters are more constrained, they still span more than half of their parameter space. This behaviour in spin accuracies is in contrast to the findings for BH-NS inspirals, where the (absolute) accuracy with which the spin magnitude and tilt can be measured *increases* with increasing spin [22]. The reasons for this contrast are probably that, firstly, the BH-NS system is simpler in that only the BH, which is much more massive than the NS, has spin. Secondly, the analysis in that study is simplified, in that it only allows for a single spinning binary component (see the next section for possible consequences).

d. Fitting for spin vs. assuming zero spin When comparing the top half of Table III, where finite spins are allowed in the parameter estimation of the three signals, to the bottom half of that table, where the spins are assumed to be zero, one finds that the accuracy is better for the latter case, for most parameters (notable exceptions are the distance and inclination, due to the fact that the secondary modes in position, orientation and arrival time described above do not completely disappear). For the signals from BHs without spin, fixing the spin parameters to their true values simply leads to more accurate masses and sky position (compare the line NS-2 to NS-0), which can be intuitively understood by the fact that constraining one parameter by using prior information will lead to an increased accuracy for correlated parameters. This effect is of course misleading, as in reality such (strong) prior information will not be available. Hence, the NS-2 result is more realistic.

While for the NS-i models assuming zero spins actually fixes those parameters to their true values, for the signals with spins (LS-i and HS-i) this is no longer true. Instead, they are fixed to the *wrong* values. The consequences are therefore more dramatic. Although Table III shows a much increased accuracy for the chirp mass (though interestingly not for most other parameters, due to the perseverance of the secondary modes described earlier), Fig. 11 shows that while the mode of the chirp-mass PDF is indeed narrower, it is also *offset* significantly from the true value (the dashed vertical line). This indicates clearly that while allowing for spins in the parameter-estimation analysis is *computationally costly* — because the dimensionality of the parameter space is much larger and the parameter space itself is more structured, making it more difficult to efficiently sample it — it is also *necessary* to obtain valid results. One of the basic as-

FIG. 11: A comparison of one-dimensional marginalised posterior PDFs for the signal with high spins, using an analysis which assumes that there are no spins (HS-0; downward hashes, blue), and an analysis which allows for spins (HS-2; upward hashes, red), for selected parameters. The PDFs are normalised in area, hence narrower PDFs have higher peaks. The dashed vertical lines indicate the parameter values of the injected signal. The most remarkable difference between the two sets of results is in the chirp mass, where the PDF for zero spin apparently shows a much more accurate result. However, when comparing to the true parameter value, that PDF is seen to be completely offset from the true value. See Sect. VI B 3 for a discussion.

sumptions in MCMC is that the model used is perfect. Performing parameter estimation with a model that does not fit for spin appears to strongly violate this assumption, resulting in seemingly good results (nicely mixing, converging chains, beautiful PDFs) which are in fact completely biased.

C. Correlations between parameters

Two examples, for the analyses LS-2 and HS-2, of cross-correlation matrices are shown in Fig. 12. The parameter sets have been divided into two subsets by the dotted lines, showing mass and spin parameters on one side and time of arrival, position and orientation parameters on the other. We will consider the four quadrants thus formed separately.

The top-left quadrant shows the mutual correlations between masses and spins, and shows that these correlations are typically stronger when more spin is present in the binary (lower-left triangle). This may explain why the determination of the chirp mass is less accurate when more spin is present; a strong correlation between two parameters allows the values of both parameters to move away from the true values — as long as they do so together, the match between model and data will still be reasonable — thus increasing the uncertainty in both parameters. Also, this may well be one of the reasons why MCMC sampling when analysing a signal with spins present is less efficient than for a signal without spins; the chirp mass is the dominant parameter and in the HS-2 case this parameter is more strongly correlated with all but two of the 14 other parameters when compared to the LS-2 analysis. Interestingly, one of the exceptions to this is the magnitude of spin 2.

The top-right and bottom-left blocks in the matrix show the correlations between mass/spin parameters on the one hand and the position/orientation parameters on the other. The most noticeable difference between the two cases is in the correlations between the mass parameters and the rest, which are significantly stronger for the HS-2 analysis than for LS-2. This may be due to the fact that one of the two modes in the position and orientation parameters disappears for the case with stronger spins (see below).

Finally, the lower-right quadrant shows the correlations amongst the position and orientation parameters (including the arrival time). For these parameters, the (near-)disappearance of the second mode in sky position, arrival time, inclination and distance (see Fig. 10 and paragraph VI B 3 b) probably plays an important role; while the second mode is there, the chains regularly jump between the two locations, and the strong correlations between especially the first five of these parameters in the matrix are likely to be a consequence of that. When the secondary mode (nearly) disappears, in the case of stronger spins, the jumping between the two modes (nearly) seizes, and the correlations become much weaker. The correlation between the orbital phase φ_c and polarisation angle ψ is due to their physical similarity; for a face-on binary without spins, these two parameters are in fact identical and hence fully degenerate [61]. For the case where the binary is not face-on, there may still be a strong correlation, which is expected to diminish when spin is present, due to the precession.

FIG. 12: Cross-correlation matrices for all 15 parameters from the analyses LS-2 (upper-right triangle; blue) and HS-2 (lower-left triangle; red). The values printed are the normalised cross-correlations as defined in Eq. 25. To guide the eye, stronger (anti)correlations have a darker background, while weaker correlations ($\sigma_{p,q} < 0.25$) have an increasingly light font — hence "empty" entries have $\sigma_{p,q} \sim 0.0$. A rough division has been made between mass and spin parameters on the one hand, and position and orientation parameters on the other by the dotted lines. See Sect. VI C for a discussion.

VII. SUMMARY AND DISCUSSION

In this technical paper, we have provided a complete description of a method of parameter estimation (PE) on gravitational-wave (GW) signals, after their detection by the LIGO-Virgo compact-binary-coalescence detection pipeline. Section II details the data filtering and windowing, the PSD generation, the computation of waveform templates, their injection into the noise and the signal characteristics that we use. In Section III we present an implementation of Markov-chain Monte Carlo (MCMC), *i.e.* the way we start and update the chains, including parallel tempering and some experimental features. Section IV deals with one of the major issues in MCMC, that of determining whether the chains have converged and hence whether the results can be trusted. In Section V we describe how the results can be presented, using one- and two-dimensional posterior probability-density functions (PDFs), correlations and Bayes factors.

In Section VI all the above is brought together for two example applications. First, we assess the effect of the experimental MCMC features described in Section III on the sampling efficiency of the MCMC in Section VI A. We find that parallel tempering always improves the sampling efficiency, and that using sinusoidal temperatures there may help after the burn-in phase. Using correlated update proposals is tricky and should not be used after the burn in. Adaptive update proposals seem to harm the MCMC sampling efficiency and should not be used. The second application is provided in Sections VI B and VI C, where we analyse a few example GW data sets containing inspiral signals of double black holes with typical SNRs. We find accuracies of 2–10 percent in the chirp mass, 35–55% in the individual masses, $\sim 100\%$ in the distance and 30–60°[2] in the sky position. While we cannot determine the black hole spins very accurately, our results can distinguish between weak, intermediate and strong spin. We conclude that the presence of spin in the system on the one hand implies additional information in the signal, through the precession of the binary orbit and thus a modulation of the GW signal, while on the other hand it may severely hamper the efficiency of the PE, due to the larger dimensionality of the parameter space, and the presence of stronger correlations between the parameters. Despite the problems the spin effects may cause, we also show in Section VI B that if the effects of spins are *not* taken into account in the PE while present in the binary and GW signal, the results can be strongly biased and may fail to recover even the most important parameter describing the signal, the chirp mass.

SPINspiral was an early implementation of MCMC for LIGO/Virgo parameter estimation, and

the first to allow for spins [22]. The code allowed us to learn to develop the techniques needed to perform PE, in particular in the complex multi-dimensional parameter space that is required to describe signals which include spin and orbital precession effects. While this code was developed, other codes were already mature, such as the MCMC PE code *followupMCMC* for binary inspirals without spin [18, 50], or being developed in parallel, like the *nested sampling code inspNest* for Bayesian model selection [59]. These codes have now been integrated in a Bayesian framework in the software package *LALInference*, which is part of LALSuite [31]. LALInference is currently the preferred software to do Bayesian analysis on gravitational-wave signals detected by LIGO and Virgo [62].

[1] B. C. Barish and R. Weiss, Physics Today **52**, 44 (1999).

[2] B. Abbott, R. Abbott, and al., New Journal of Physics **11**, 073032 (2009).

[3] M. Landry, in *APS April Meeting Abstracts* (2012), p. B8003.

[4] F. Acernese, M. Alshourbagy, and al., Class. Quant. Grav. **25**, 184001 (2008).

[5] T. Accadia, F. Acernese, and al., Class. Quant. Grav. **28**, 114002 (2011).

[6] K. Belczynski, R. E. Taam, E. Rantsiou, and M. van der Sluys, ApJ **682**, 474 (2008).

[7] S. N. Zhang, W. Cui, and W. Chen, ApJL **482**, L155 (1997).

[8] R. Shafee, J. E. McClintock, R. Narayan, S. W. Davis, L. Li, and R. A. Remillard, Astrophys. J. **636**, L113 (2006).

[9] L. Gou, J. E. McClintock, J. Liu, R. Narayan, J. F. Steiner, R. A. Remillard, J. A. Orosz, S. W. Davis, K. Ebisawa, and E. M. Schlegel, Astrophys. J. **701**, 1076 (2009), 0901.0920.

[10] A. C. Fabian, D. R. Wilkins, J. M. Miller, R. C. Reis, C. S. Reynolds, E. M. Cackett, M. A. Nowak, G. G. Pooley, K. Pottschmidt, J. S. Sanders, et al., MNRAS **424**, 217 (2012), 1204.5854.

[11] R. P. Fender, E. Gallo, and D. Russell, MNRAS **406**, 1425 (2010), 1003.5516.

[12] L. Gou, J. E. McClintock, M. J. Reid, J. A. Orosz, J. F. Steiner, R. Narayan, J. Xiang, R. A. Remillard, K. A. Arnaud, and S. W. Davis, Astrophys. J. **742**, 85 (2011), 1106.3690.

[13] R. Duro, T. Dauser, J. Wilms, K. Pottschmidt, M. A. Nowak, S. Fritz, E. Kendziorra, M. G. F. Kirsch, C. S. Reynolds, and R. Staubert, A&A **533**, L3 (2011), 1108.1157.

[14] A. P. Cowley, P. C. Schmidtke, K. Ebisawa, F. Makino, R. A. Remillard, D. Crampton, J. B. Hutchings, S. Kitamoto, and A. Treves, Astrophys. J. **381**, 526 (1991).

[15] A. P. Smale and P. T. Boyd, Astrophys. J. **756**, 146 (2012), 1207.3775.

[16] SPINspiral, URL `http://spinspiral.sf.net`.

[17] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, *Markov chain Monte Carlo in practice* (London/Boca Raton, Fl.: Chapman & Hall/CRC, 1996).

[18] C. Röver, R. Meyer, and N. Christensen, Phys. Rev. D **75**, 062004 (2007).

[19] C. Röver, Ph.D. thesis, The University of Auckland (2007), URL `http://hdl.handle.net/2292/2356`.

[20] T. A. Apostolatos, C. Cutler, G. J. Sussman, and K. S. Thorne, Phys. Rev. D **49**, 6274 (1994).

[21] A. Buonanno, Y. Chen, and M. Vallisneri, Phys. Rev. D **67**, 104025 (2003).

[22] M. V. van der Sluys, C. Röver, A. Stroeer, V. Raymond, I. Mandel, N. Christensen, V. Kalogera, R. Meyer, and A. Vecchio, ApJ **688**, L61 (2008).

[23] V. Raymond, M. V. van der Sluys, I. Mandel, V. Kalogera, C. Röver, and N. Christensen, Classical and Quantum Gravity **26**, 114007 (2009), 0812.4302.

[24] M. van der Sluys, I. Mandel, V. Raymond, V. Kalogera, C. Röver, and N. Christensen, Classical and Quantum Gravity **26**, 204010 (2009), 0905.1323.

[25] V. Raymond, M. V. van der Sluys, I. Mandel, V. Kalogera, C. Röver, and N. Christensen, Classical and Quantum Gravity **27**, 114009 (2010), 0912.3746.

[26] L. Cadonati *et al.*, Class. Quant. Grav. **26**, 114008 (2009), 0905.4227.

[27] B. Aylott *et al.*, Class. Quant. Grav. **26**, 165008 (2009), 0901.4399.

[28] M. Van der Sluys, V. Raymond, I. Mandel, C. Röver, N. Christensen, V. Kalogera, R. Meyer, and A. Vecchio, Class. Quant. Grav. **25**, 184011 (2008), 0805.1689.

[29] Gouaty R for the LIGO Scientific Collaboration, Class. Quant. Grav. **25**, 184006 (2008).

[30] B. P. Abbott, R. Abbott, and al., Phys. Rev. D **79**, 122001 (2009), 0901.0302.

[31] LIGO Scientific Collaboration, *The LSC Algorithm Library suite (LALSuite)*, URL `https://www.lsc-group.phys.uwm.edu/daswg/projects/lalsuite.html`.

[32] *Frame library (Fr)*, URL `http://lappweb.in2p3.fr/virgo/FrameL`.

[33] D. A. Brown and the LIGO Scientific Collaboration, Class. Quant. Grav. **21**, 797 (2004).

[34] D. Sigg, Class. Quant. Grav. **21**, S409 (2004).

[35] W. Fraser, J. ACM **12**, 295 (1965), ISSN 0004-5411.

[36] J. Janovetz, *Parks-McClellan (Remez exchange) program*, URL `http://www.janovetz.com/jake/`.

[37] J. M. Bardeen, W. H. Press, and S. A. Teukolsky, Astrophys. J. **178**, 347 (1972).

[38] F. J. Harris, Proceedings of the IEEE **66**, 51 (1978).

[39] M. Frigo and S. G. Johnson, Proceedings of the IEEE **93**, 216 (2005), URL `http://www.fftw.org/`.

[40] J. G. Baker, J. Centrella, D.-I. Choi, M. Koppitz, and J. van Meter, Phys. Rev. D **73**, 104002 (2006), URL `http://link.aps.org/doi/10.1103/PhysRevD.73.104002`.

[41] A. Buonanno, G. B. Cook, and F. Pretorius, Phys. Rev. D **75**, 124018 (2007), URL `http://link.aps.org/doi/10.1103/PhysRevD.75.124018`.

[42] M. Hannam, S. Husa, B. Brügmann, and A. Gopakumar, Phys. Rev. D **78**, 104007 (2008), URL `http://link.aps.org/doi/10.1103/PhysRevD.78.104007`.

[43] J. G. Baker, J. R. van Meter, S. T. McWilliams, J. Centrella, and B. J. Kelly, Phys. Rev. Lett. **99**, 181101 (2007), URL `http://link.aps.org/doi/10.1103/PhysRevLett.99.181101`.

[44] M. Boyle, D. A. Brown, L. E. Kidder, A. H. Mroué, H. P. Pfeiffer, M. A. Scheel, G. B. Cook, and S. A. Teukolsky, Phys. Rev. D **76**, 124038 (2007), URL `http://link.aps.org/doi/10.1103/`

PhysRevD.76.124038.
[45] M. Campanelli, C. O. Lousto, H. Nakano, and Y. Zlochower, Phys. Rev. D **79**, 084010 (2009), URL http://link.aps.org/doi/10.1103/PhysRevD.79.084010.
[46] B. Allen (2000), URL http://www.lsc-group.phys.uwm.edu/~ballen/grasp-distribution.
[47] R. Sturani, S. Fischetti, L. Cadonati, G. M. Guidi, J. Healy, D. Shoemaker, and A. Viceré, Journal of Physics Conference Series **243**, 012007 (2010).
[48] C. Cutler and É. E. Flanagan, Phys. Rev. D **49**, 2658 (1994).
[49] C. Röver, R. Meyer, and N. Christensen, Class. Quant. Grav. **23**, 4895 (2006).
[50] C. Röver, R. Meyer, G. M. Guidi, A. Viceré, and N. Christensen, Class. Quant. Grav. **24**, 607 (2007b), 0707.3962.
[51] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, The Journal of Chemical Physics **21**, 1087 (1953), URL http://link.aip.org/link/?JCP/21/1087/1.
[52] Y. F. Atchadé and J. S. Rosenthal, Bernoulli **11**, 815 (2005).
[53] K. Hukushima and K. Nemoto, Journal of the Physical Society of Japan **65**, 1604 (1996), arXiv:cond-mat/9512035.
[54] U. Hansmann, Chemical Physics Letters **281**, 140 (1997), arXiv:physics/9710041.
[55] A. Gelman and D. B. Rubin, Stat. Sciences **7**, 457 (1992).
[56] P. Brooks and A. Gelman, Journal of Comp. & Graph. Stat. **7**, 434 (1998).
[57] AnalyseMCMC, URL http://analysemcmc.sf.net.
[58] J. Skilling, Bayesian Anal. **1**, 833 (2006), ISSN 1931-6690.
[59] J. Veitch and A. Vecchio, Classical and Quantum Gravity **25**, 184010 (2008), 0807.4483.
[60] M. A. Newton and A. E. Raftery, Journal of the Royal Statistical Society. Series B (Methodological) **56**, 3 (1994), ISSN 00359246, URL http://www.jstor.org/stable/2346025.
[61] S. Shah, M. van der Sluys, and G. Nelemans, A&A **544**, A153 (2012), 1207.6770.
[62] J. Aasi, J. Abadie, B. P. Abbott, and al., *in preparation* (2013).
[63] As in any other iteration, the proposed starting value must lie within the prior range in order to be accepted
[64] See for example Fig. 6, where large jump proposals will typically only be accepted if they are oriented along the crescent.
[65] Actually, the order is not important.
[66] Even though this would implicitly assume, possibly incorrectly, that the expansion from the PDF found to the true PDF would be symmetric around e.g. the median.
[67] However, this would involve a pre-determined burn-in length, separate from the automatically, *a posteriori*, determined one described in Section IV A. Useful lengths for this burn in can be learnt from experience and their data should always be discarded in the analysis.
[68] Even if a single chain (without parallel tempering) running five times longer would yield a similar result, it would require the same amount of CPU time but five times as much clock time, which may be undesired.
[69] Again, this would be a pre-determined burn in, different from the method described in Sect. IV A.