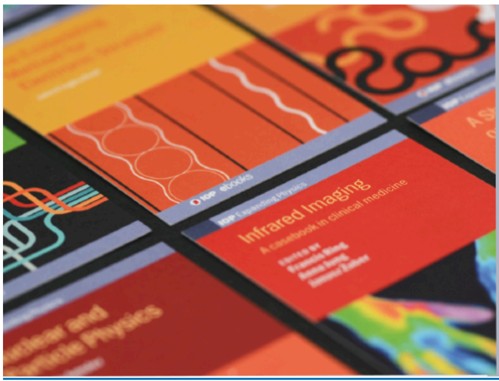
Classical and Quantum Gravity

PAPER

Initial data and eccentricity reduction toolkit for binary black hole numerical relativity waveforms

To cite this article: Sarah Habib *et al* 2021 *Class. Quantum Grav.* **38** 125007



IOP Publishing Classical and Quantum Gravity

Class. Quantum Grav. **38** (2021) 125007 (9pp) https://doi.org/10.1088/1361-6382/abe691

Initial data and eccentricity reduction toolkit for binary black hole numerical relativity waveforms

Sarah Habib[1,2,*], Antoni Ramos-Buades[3], E A Huerta[1,4,5,6,7,*], Sascha Husa[8], Roland Haas[1] and Zachariah Etienne[9,10]

[1] National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801, United States of America
[2] TAPIR, Walter Burke Institute for Theoretical Physics, California Institute of Technology, Pasadena, California 91125, United States of America
[3] Max Planck Institute for Gravitational Physics (Albert Einstein Institute), Am Mühlenberg 1, D-14476 Potsdam, Germany
[4] Department of Physics, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801, United States of America
[5] Department of Astronomy, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801, United States of America
[6] Illinois Center for Advanced Studies of the Universe, University of Illinois at Urbana-Champaign, Urbana, Illinois, 61801, United States of America
[7] University of Chicago, Chicago, Illinois 60637, United States of America
[8] Departament de Física, Universitat de les Illes Balears, IAC3—IEEC, Crta. Valldemossa km 7.5, E-07122 Palma, Spain
[9] Department of Physics and Astronomy, West Virginia University, Morgantown, West Virginia 26506, United States of America
[10] Center for Gravitational Waves and Cosmology, West Virginia University, Chestnut Ridge Research Building, Morgantown, West Virginia 26505, United States of America

E-mail: shabib@caltech.edu and elihu@anl.gov

Received 25 November 2020, revised 27 January 2021
Accepted for publication 15 February 2021
Published 13 May 2021

Abstract

The production of numerical relativity waveforms that describe quasi-circular binary black hole mergers requires high-quality initial data, and an algorithm to iteratively reduce residual eccentricity. To date, these tools remain closed source, or in commercial software that prevents their use in high performance computing platforms. To address these limitations, and to ensure that the broader numerical relativity community has access to these tools, herein we provide all the required elements to produce high-quality numerical relativity simulations in supercomputer platforms, namely: open source parameter files to

[*] Authors to whom any correspondence should be addressed.

numerically simulate spinning black hole binaries with asymmetric mass-ratios; open source Python tools to produce high-quality initial data for numerical relativity simulations of spinning black hole binaries on quasi-circular orbits; and open source Python tools for eccentricity reduction, both as stand-alone software and also deployed in the Einstein Toolkit's software infrastructure. This open source toolkit fills in a void in the literature at a time when numerical relativity has an ever increasing role in the study and interpretation of gravitational wave sources. As part of our community building efforts, and to streamline and accelerate the use of these resources, we provide tutorials that describe, step by step, how to obtain and use these open source numerical relativity tools.

Keywords: numerical relativity, initial data, black holes, eccentricity reduction

(Some figures may appear in colour only in the online journal)

1. Introduction

Numerical relativity [1–4] plays a central role in contemporary gravitational wave astrophysics [5–8]. The use of numerical relativity waveforms has been essential to develop approximate waveform models that are extensively used for gravitational wave detection and parameter estimation [9–13]. The construction of numerical relativity waveform catalogs [14–19] has enabled in-depth analyses of the astrophysical properties of gravitational wave sources [20–23].

As gravitational wave astrophysics continues to probe the gravitational wave spectrum [24–28], numerical relativity will be essential to enable and interpret new discoveries, enlighten our understanding of the physics of these sources, and provide constraints that may further establish general relativity or favor alternatives theories of gravity [29–31].

Advancing our understanding of gravitational wave sources depends critically on the production of high quality numerical relativity waveforms that, in the case of binary black hole mergers, span an 8D parameter space that includes mass-ratio, two 3D vectors that define the individual spin of the binary components, and orbital eccentricity, $(q, \mathbf{s}_1, \mathbf{s}_2, e)$, respectively. It is then apparent that despite the existence of thousands of numerical relativity waveforms, we need to be creative about how to combine them to densely sample this high dimensional signal manifold [32, 33]. It is also clear that we need to continue producing numerical relativity waveforms to describe sources whose parameters are not accurately captured by existing approximate waveform models or available numerical relativity waveforms.

In order to empower the broader numerical relativity community to participate in the construction of numerical relativity waveform catalogs, we introduce open source Python libraries that have been tested and deployed within the Einstein Toolkit [34] to streamline and accelerate these activities. This approach builds upon our previous software development that consisted of open source Python libraries to post-process numerical relativity data to extract the waveform strain at future null infinity [35]. These combined tools provide the required end-to-end software infrastructure to utilize the Einstein Toolkit for the construction of high-quality numerical relativity waveform catalogs.

This manuscript is organized as follows. Section 2 describes our approach to construct high-quality initial data, and to post-process the data products of numerical relativity simulations to remove residual eccentricity. We put these tools at work in section 3, where we show that we can produce nearly circular initial data, and that our method for eccentricity reduction produces waveforms with eccentricities of order $\mathcal{O} \sim 10^{-4}$ after just one iteration. We summarize this work, and outline future research directions in section 4. We present a tutorial that describes how to obtain and use these libraries in appendix A.

2. Methods

In this section we describe the approach followed to produce high-quality initial data for binary black hole simulations. Thereafter, we briefly introduce the method used for eccentricity reduction.

2.1. Initial data production

The first guesses for the tangential, p_ϕ, and radial, p_r, components of the momenta for the black hole binary system are generated using techniques presented in [36, 37]. They extract momenta components from Hamilton's equations of motion in post-Newtonian (PN) theory, combined with high-PN-order expressions for the gravitational-wave flux, $\mathrm{d}E_{\mathrm{GW}}/\mathrm{d}t$, and the tidal energy injected into the black holes, $\mathrm{d}M/\mathrm{d}t$.

The Hamiltonian contains orbital [38], spin–orbit [38–40], spin–spin [38, 41, 42], and spin–spin–spin [43] terms up to and including 3.5 PN order.

The high-PN-order expressions for $\mathrm{d}E_{\mathrm{GW}}/\mathrm{d}t$ incorporate nonspinning and precessing-spin terms [44, 45], and are adjusted to account for the tidal energy injected into the black holes $\mathrm{d}M/\mathrm{d}t$ [46].

The above expressions were implemented in the open-source, Python-based `NRPyPN` software, which is part of `NRPy`+ ('Python-based code generation for numerical relativity... and beyond!') [47]. A tutorial for using the software is given in appendix A.1 below. In short, the expression for tangential momentum $p_\phi(r)$ up to and including 3.5 PN order is taken from [36] and validated up to 3 PN order against [37], and up to 3.5 PN order against the original `Mathematica` notebooks used by [36].

Meanwhile, the expression for radial momentum p_r up to and including 3.5 PN order is derived in `NRPyPN` as follows. First, Hamilton's equations of motion imply that

$$\frac{\mathrm{d}r}{\mathrm{d}t} = \frac{\partial H}{\partial p_r}. \tag{1}$$

Next we Taylor expand $\partial H/\partial p_r$ in powers of p_r, about $p_r = 0$, to obtain (to first order in p_r):

$$p_r \approx \left(\frac{\mathrm{d}r}{\mathrm{d}t} - \left.\frac{\partial H}{\partial p_r}\right|_{p_r=0} \right) \left(\left.\frac{\partial^2 H}{\partial p_r^2}\right|_{p_r=0} \right)^{-1}, \tag{2}$$

where

$$\frac{\mathrm{d}r}{\mathrm{d}t} = \left(\frac{\mathrm{d}E_{\mathrm{GW}}}{\mathrm{d}t} + \frac{\mathrm{d}M}{\mathrm{d}t} \right) \left[\frac{\mathrm{d}H_{\mathrm{circ}}}{\mathrm{d}r} \right]^{-1}, \tag{3}$$

and

$$\frac{\mathrm{d}H_{\mathrm{circ}}(r, p_\phi(r))}{\mathrm{d}r} = \frac{\partial H(p_r = 0)}{\partial r} + \frac{\partial H(p_r = 0)}{\partial p_\phi} \frac{\partial p_\phi}{\partial r}, \tag{4}$$

are given explicitly in terms of binary input parameters and $M\Omega$ (as given to 3.5 PN order by [36]).

2.2. Eccentricity reduction

The algorithm we describe in this section was introduced in [36], and was originally developed as a `Mathematica` notebook. As part of this work, we have re-written this eccentricity reduction method using `Python` libraries, optimized it, and tailored it to conduct automated, large-scale, numerical relativity campaigns on high performance computing platforms.

This eccentricity reduction method is applied to remove eccentricity from a numerical relativity simulation whose initial data were produced with the method described in the previous section. Once the numerical simulation has progressed enough, typically between 500M to 600M of evolution, we process the relevant data files, as described in appendix A, to compute correction factors, λ_ϕ, λ_r, of the initial components of the momenta p_ϕ^0 and p_r^0.

To compute λ_ϕ, λ_r we assume that oscillations induced as a result of eccentricity in the orbital frequency, Ω, take the form

$$\mathcal{R}_\Omega = A + B\cos(\Omega_r t + \Psi), \tag{5}$$

where A, B and Ψ are coefficients to be determined, and Ω_r is the frequency of the radial oscillations. Using the 1 PN order quasi-Keplerian parametrization [48], we can obtain closed form expressions for these correction factors

$$\lambda_\phi = 1 + \left[\frac{B}{4\Omega_0} - \gamma\frac{B(3\eta + 1)}{8r_0\Omega_0}\right]\cos\Psi, \tag{6}$$

$$\lambda_r = 1 + \frac{B\eta}{2r_0^{1/2}\Omega_0\,|p_r^0|}\left[1 + \gamma\frac{1}{r_0}\right]\sin\Psi, \tag{7}$$

where $\eta = m_1m_2/(m_1 + m_2)$ is the symmetric mass ratio, (m_1, m_2) represent the masses of the binary components, r_0 is the initial orbital separation, and Ω_0 is the quasi-circular initial orbital frequency calculated at 3.5 PN order [36]. In appendix A.1 we describe how to use simulation data and analytical approximations to compute the correction factors λ_ϕ and λ_r.

3. Results

In this section we combine the tools described above for initial data production and eccentricity reduction. We selected two binary black hole systems whose properties are described in table 1. These include the separation between the punctures, $D[M]$; the 3D spin vectors of the black holes, $\mathbf{s}_i = (\chi_i^x, \chi_i^y, \chi_i^z)$, with $i = \{1, 2\}$; and the mass-ratio of the binary system, q. Iteration zero corresponds to the simulation run with the initial data for the momenta, (p_r, p_ϕ), provided by our toolkit. Iteration 1 utilizes our eccentricity correction software to estimate values for the momenta that best describe a nearly quasi-circular binary black hole system. Notice that these systems span two different mass-ratios, $q \in \{1, 3\}$, and several spinning, non-precessing configurations.

The results presented in table 1 show that for all the binary systems under consideration, our toolkit produces systems whose initial eccentricities are $e_0 \sim 10^{-3}$. Furthermore, these eccentricity values are reduced to $e_0 \sim 10^{-4}$ after just one iteration, clearly indicating that our eccentricity reduction algorithm does remove residual eccentricity by $2 - 3X$ after just one iteration. It is worth highlighting that initial data codes used to simulate quasi-circular binary black holes typically produce systems with residual eccentricity $e_0 \sim 10^{-2}$ for iteration zero. However, our initial data produces simulations with eccentricities $e_0 \sim 10^{-3}$ from the very first

Table 1. Summary of the astrophysical and orbital parameters of three binary black hole systems used to test our open source toolkit for the production of initial data and eccentricity reduction. Notice that in all cases the initial eccentricity is of order $e_0 \sim \mathcal{O}\left(10^{-3}\right)$ for the zeroth iteration, and it is reduced to $e_0 \sim \mathcal{O}\left(10^{-4}\right)$ after just one iteration.

$D[M]$	$(\chi_1^x, \chi_1^y, \chi_1^z)$	$(\chi_2^x, \chi_2^y, \chi_2^z)$	q	Iter #	$p_r[\times 10^{-4}]$	$p_\phi[\times 10^{-2}]$	$e_0[\times 10^{-3}]$
11.0	(0.0, 0.0, −0.4)	(0.0, 0.0, −0.5)	1.0	**0**	−8.60	9.293	2.43
S_q_1				**1**	−7.70	9.284	0.74
9.0	(0.0, 0.0, 0.4)	(0.0, 0.0, −0.5)	3.0	**0**	−7.50	7.652	1.70
S_q_3				**1**	−6.60	7.650	0.71

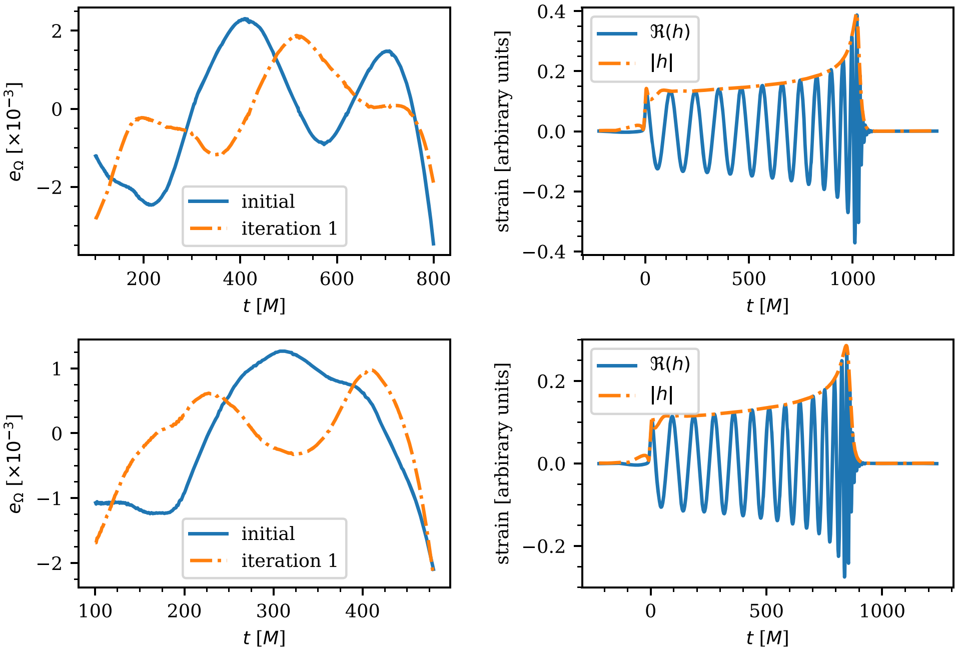


Figure 1. Top panels: results of the eccentricity reduction procedure for simulation **S_q_1**. Top-left panel: eccentricity estimator results for the zeroth and first iterations. Top-right panel: gravitational wave signal, extracted at future null infinity, after eccentricity reduction. Bottom panels: as top panels, but for simulation **S_q_3**.

iteration. In other words, these ready-to-use tools produce high-quality numerical relativity waveforms after a minimal number of iterations.

Figure 1 presents two types of results. The left panels present results for the eccentricity estimator e_Ω of the orbital frequency defined in equation (3.13) of [36]. These results show, as discussed previously, that even the zeroth iteration is already nearly circular. The right panels present waveforms of the first iteration extracted at future null infinity [35].

These results indicate that the open source tools presented in this article, along with the tutorials and configuration files released with this work, will provide the required building blocks to enable a broader cross section of the numerical relativity community to participate in the production of numerical relativity waveform catalogs.

4. Conclusions

Numerical relativity simulations [1–3] of binary black hole mergers were produced a decade before the first gravitational wave detection of these astrophysical events was realized by the advanced LIGO detectors [24]. Over the last decade, numerical relativity software stacks have matured to the point of automating and streamlining the production of large-scale numerical relativity catalogs [14–19]. Nonetheless, the available number of numerical relativity waveforms is not sufficient to densely cover the high dimensional signal manifold spanned by these astrophysical events.

Furthermore, essential tools to produce initial data and to automate eccentricity reduction continue to be kept as closed source software or licensed software. Neither of these solutions is adequate if we aim to enable a larger cross section of the numerical relativity community to participate in the production of numerical relativity waveforms to accurately infer the astrophysical properties of compact binary sources. This need will become a pressing issue as advanced gravitational wave detectors gradually reach design sensitivity, and the number of detections reaches the expected number of one event for every 15 min of searched data.

The deployment of these tools as stand-alone software and within the Einstein Toolkit builds upon our community building efforts to produce and release scientific software that streamlines the production of high quality initial data, automates eccentricity reduction, and facilitates post-processing of numerical relativity data products to extract numerical relativity waveforms at future null infinity [35]. These user-friendly tools will allow new users to use and further develop open source numerical relativity software, using the `Einstein Toolkit` as the driver for such activities.

Acknowledgments

EAH gratefully acknowledges National Science Foundation (NSF) awards OAC-1931561 and OAC-1934757. RH gratefully acknowledges NSF awards OAC-1550514, OAC-2004879, and ACI-1238993. This research is part of the Blue Waters sustained-petascale computing project, which is supported by the National Science Foundation (awards OCI-0725070 and ACI-1238993), the State of Illinois, and as of December, 2019, the National Geospatial-Intelligence Agency. Blue Waters is a joint effort of the University of Illinois at Urbana-Champaign and its National Center for Supercomputing Applications. We acknowledge support from the NCSA and the Students Pushing INnovation (SPIN) undergraduate internship Program at NCSA. We thank the NCSA Gravity Group for useful feedback. NSF-1659702 and XSEDE TG-PHY160053 grants are gratefully acknowledged.

Data availability statement

The data that support the findings of this study are available upon reasonable request from the authors.

Appendix A. Step by step tutorial

The tools introduced below may be used as stand-alone scientific software or within the software infrastructure of the Einstein Toolkit.

A.1. Initial data production

NRPyPN is part of the open-source NRPy+ ('Python-based code generation for numerical relativity… and beyond!') [47], and provides the zeroth estimate for low-eccentricity initial data in this paper. To obtain this estimate from NRPyPN, first clone the NRPy+ github repository:

```
git clone https://github.com/zachetienne/nrpytutorial.git
```

Then (assuming that Python 2 or 3 is installed with pip), install SymPy [49] and Jupyter:

```
pip install -U sympy jupyter
```

Next, from within the nrpytutorial/NRPyPN/directory, run

```
jupyter notebook NRPyPN.ipynb
```

A Jupyter notebook will open, in which the binary black hole initial data parameters for initial separation, spins, and mass ratio can be specified in the code cell at the bottom of the notebook. When the code cell is run (shift + enter), the radial p_r and tangential momenta p_ϕ to 3.5 PN order (largely following [36] but fully documented in the linked Jupyter notebooks) will be output. These momenta can be directly inserted into a Bowen-York binary black hole initial data solver (the TwoPunctures thorn was used in this work). For example, for a binary orbiting in the xy-plane with black holes initially located on the x-axis at $x = \pm a$ (with the center of mass at the origin, $x = y = z = 0$), $\pm|p_r|$ will correspond to the x-component of momentum p_x for the puncture at $x = \mp a$, respectively. One may also choose $\pm|p_\phi|$ to correspond to the y-component of momentum p_y for the puncture at $x = \pm a$ or $\mp a$, depending on whether a clockwise or a counterclockwise orbit is desired.

A.2. Eccentricity reduction

Our eccentricity code is an available as a Python3 module on GitHub [50]

```
git clone https://github.com/ncsagravity/eccred
```

Then, assuming Python 3 is installed, run:

```
python >>> import EccRed >>> EccRed.ComputeCorrections
('output_glob', MinTime = X, MaxTime = Y)
```

Where *output_glob* is a shell pattern (glob) that matches all directories containing output files, and MinTime and MaxTime are time bounds. For best results, MinTime should be shortly after any 'junk' radiation has passed from the vicinity of the black holes and any initial gauge transition has settled, and MaxTime should be close to when the time plunge occurs.

Four correction values as well as the estimated eccentricity will be returned from EccRed.ComputeCorrections. In order, they are λ_r, λ_ϕ computed using two different methods (from PN expansion and from an eccentricity estimator respectively), and δR, the correction factors to radial and tangential momentum components and (additive) correction to initial orbital separation respectively. These corrections can then be applied to the respective initial values.

The code expects two sets of files in the output directories: (i) a file *TwoPunctures.bbh* as produced by the TwoPunctures thorn that describes the parameters of the initial black holes, and (ii) a set of puncture location files *puncturetracker-pt_loc…asc* as produced by the PunctureTracker thorn. Columns *pt_loc_x[0], pt_loc_x[1]* etc, are expected to contain the location of the original plus and minus punctures. This matches the setup in [51].

In the event that EccRed.ComputeCorrections throws a runtime error, a likely solution is to adjust MinTime or MaxTime to better characterize the time domain of inspiral.

A.3. Automated eccentricity reduction

To automate the process of eccentricity reduction using `Simulation Factory` [52] the `Python` module can be called as a command line script

```
./EccRed.py --tmin X --tmax X --input-parfile
'input_parameter_file'\--output-parfile 'output_parameter_file'
'output_glob'
```

Which automatically applies the correction factors to `TwoPunctures`' radial and tangential momentum parameters found in *input_parameter_file* and produces a new parameter file in *output_parameter_file*.

The eccentricity reduction code provides a fragment of code in `RunScript.part` that can be inserted into `Simulation Factory`'s run script files to automate the process of extending a simulation until sufficient data has been produced, estimating eccentricity, computing correction factors, applying them to the parameter file and submitting a new round of eccentricity reduction.

The fragment contains placeholders *@ECC_TARGET@* and *@ECC_TIME@* for the estimated eccentricity at which to stop the iteration and the time for which to simulate before applying the correction algorithm:

```
sim create --define ECC_TARGET 'ECC_TARGET' --define
ECC_TIME 'ECC_TIME'...
```

Which starts the automated process.

ORCID iDs

Sarah Habib https://orcid.org/0000-0002-4725-4978
Antoni Ramos-Buades https://orcid.org/0000-0002-6874-7421
E A Huerta https://orcid.org/0000-0002-9682-3604
Sascha Husa https://orcid.org/0000-0002-0445-1971
Roland Haas https://orcid.org/0000-0003-1424-6178
Zachariah Etienne https://orcid.org/0000-0002-6838-9185

References

[1] Pretorius F 2005 *Phys. Rev. Lett.* **95** 121101
[2] Campanelli M, Lousto C, Marronetti P and Zlochower Y 2006 *Phys. Rev. Lett.* **96** 111101
[3] Baker J G, Centrella J, Choi D I, Koppitz M and van Meter J 2006 *Phys. Rev. Lett.* **96** 111102
[4] Cardoso V, Gualtieri L, Herdeiro C and Sperhake U 2015 *Living Rev. Relativ.* **18** 1
[5] Sperhake U 2015 *Class. Quantum Grav.* **32** 124011
[6] Centrella J, Baker J G, Kelly B J and van Meter J R 2010 *Rev. Mod. Phys.* **82** 3069
[7] Baumgarte T W and Shapiro S L 2010 *Numerical Relativity: Solving Einstein's Equations on the Computer* (Cambridge: Cambridge University Press)
[8] Alcubierre M 2008 *Introduction to 3 + 1 Numerical Relativity* (Oxford: Oxford University Press)
[9] Hannam M, Schmidt P, Bohé A, Haegel L, Husa S, Ohme F, Pratten G and Pürrer M 2014 *Phys. Rev. Lett.* **113** 151101
[10] Bohé A *et al* 2017 *Phys. Rev.* D **95** 044028
[11] Khan S, Husa S, Hannam M, Ohme F, Pürrer M, Jiménez Forteza X and Bohé A 2016 *Phys. Rev.* D **93** 044007
[12] Blackman J, Field S E, Scheel M A, Galley C R, Ott C D, Boyle M, Kidder L E, Pfeiffer H P and SzilÃ¡gyi B 2017 *Phys. Rev.* D **96** 024058
[13] Husa S, Khan S, Hannam M, Pürrer M, Ohme F, Jiménez Forteza X and Bohé A 2016 *Phys. Rev.* D **93** 044006

[14] Mroue A H *et al* 2013 *Phys. Rev. Lett.* **111** 241104
[15] Healy J, Lousto C O, Zlochower Y and Campanelli M 2017 *Class. Quantum Grav.* **34** 224001
[16] Boyle M *et al* 2019 *Class. Quant. Grav.* **36** 195006
[17] Healy J, Lousto C O, Lange J, O'Shaughnessy R, Zlochower Y and Campanelli M 2019 *Phys. Rev.* D **100** 024021
[18] Huerta E *et al* 2019 *Phys. Rev.* D **100** 064003
[19] Jani K, Healy J, Clark J A, London L, Laguna P and Shoemaker D 2016 *Class. Quantum Grav.* **33** 204001
[20] Kumar P *et al* 2019 *Phys. Rev.* D **99** 124005
[21] Abbott B *et al* (LIGO Scientific, Virgo) 2016 *Phys. Rev.* D **94** 064035
[22] Lange J *et al* 2017 *Phys. Rev.* D **96** 104041
[23] Lovelace G *et al* 2016 Modeling the source of GW150914 with targeted numerical-relativity simulations *Class. Quantum Grav.* **33** 244002
[24] Abbott B *et al* (LIGO Scientific, Virgo) 2019 *Phys. Rev.* X **9** 031040
[25] Abbott B *et al* (LIGO Scientific, Virgo) 2017 *Phys. Rev. Lett.* **119** 161101
[26] Abbott B *et al* (LIGO Scientific, Virgo) 2020 *Astrophys. J. Lett.* **892** L3
[27] Abbott R *et al* (LIGO Scientific, Virgo) 2020 arXiv:2004.08342
[28] Abbott R *et al* 2020 arXiv:2010.14527
[29] Okounkova M, Stein L C, Moxon J, Scheel M A and Teukolsky S A 2020 *Phys. Rev.* D **101** 104016
[30] Yunes N, Yagi K and Pretorius F 2016 *Phys. Rev.* D **94** 084002
[31] Nair R, Perkins S, Silva H O and Yunes N 2019 *Phys. Rev. Lett.* **123** 191101
[32] Varma V, Field S E, Scheel M A, Blackman J, Kidder L E and Pfeiffer H P 2019 *Phys. Rev.* D **99** 064045
[33] Rifat N E, Field S E, Khanna G and Varma V 2020 *Phys. Rev.* D **101** 081502
[34] Löffler F *et al* 2012 *Class. Quantum Grav.* **29** 115001
[35] Johnson D, Huerta E A and Haas R 2018 *Class. Quantum Grav.* **35** 027002
[36] Ramos-Buades A, Husa S and Pratten G 2019 *Phys. Rev.* D **99** 023003
[37] Healy J, Lousto C O, Nakano H and Zlochower Y 2017 *Class. Quantum Grav.* **34** 145011
[38] Buonanno A, Chen Y and Damour T 2006 *Phys. Rev.* D **74** 104005
[39] Damour T, Jaranowski P and Schaefer G 2008 *Phys. Rev.* D **77** 064032
[40] Hartung J and Steinhoff J 2011 *Ann. Phys., Lpz.* **523** 783–90
[41] Steinhoff J, Hergt S and Schaefer G 2008 *Phys. Rev.* D **77** 081501
[42] Steinhoff J, Hergt S and Schaefer G 2008 *Phys. Rev.* D **78** 101503
[43] Levi M and Steinhoff J 2015 *J. High Energy Phys.* JHEP06(2015)059
[44] Blanchet L 2014 *Living Rev. Relativ.* **17** 2
[45] Ossokine S, Boyle M, Kidder L E, Pfeiffer H P, Scheel M A and Szilágyi B 2015 *Phys. Rev.* D **92** 104028
[46] Brown D, Fairhurst S, Krishnan B, Mercer R, Kopparapu R, Santamaria L and Whelan J 2007 arXiv:0709.0093
[47] Ruchlin I, Etienne Z B and Baumgarte T W 2018 *Phys. Rev.* D **97** 064036
[48] Damour T and Deruelle N 1985 *Ann. Inst. Henri Poincaré Phys. Théor.* **43** 107–32
[49] Meurer A *et al* 2017 Sympy: symbolic computing in python *Peer J. Comput. Sci.* **3** e103
[50] Habib S, Huerta E, Haas R and Etienne Z B 2020 Automated framework to produce low eccentricity numerical binary black hole simulations https://github.com/ncsagravity/eccred
[51] Wardell B, Hinder I and Bentivegna E 2016 Simulation of GW150914 binary black hole merger using the Einstein Toolkit https://doi.org/10.5281/zenodo.155394
[52] SimFactory http://simfactory.org/