

High-Accuracy Binary Black Hole Simulations in SpEC

Thesis by
Himanshu Chaudhary

In Partial Fulfillment of the Requirements for the
Degree of
PhD



CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2026
Defended 28-May-2026

© 2026

Himanshu Chaudhary
ORCID: 0000-0002-4101-0534

All rights reserved except where otherwise noted

ACKNOWLEDGEMENTS

I would like to thank my advisors, Saul Teukolsky and Mark Scheel, for all the support and guidance that they provided throughout my PhD. I am extremely grateful for the freedom they gave me to explore unconventional ideas and all the discussions we had over the years. I am also extremely grateful to Rob Owen for being a part of this journey and helping me with all the technicalities and mathematical nuances of Numerical Relativity.

I would also like to thank Nils Vu for his collaboration and various discussions that we had both about research and beyond. Special thanks to his son, Charlie, for being cute and a reason for the whole group to get together and celebrate.

I would also like to thank all the current and past members of the SXS collaboration for writing SpEC, SpECTRE, and various other codes that my research relied on.

Thank you to all my friends, both part of the collaboration and outside, for making my life at Caltech memorable and fun.

Thank you, JoAnn, for taking care of all the administrative stuff and making our stay at Caltech a lot more pleasant.

I would also like to thank my thesis committee members Yanbei Chen and Katerina Chatziioannou for their support of my work.

I would like to thank my family for all they have done for me and all the support they provide. Also, thank you to my partner, Deeksha, for all the joy and happiness that you bring to my life.

ABSTRACT

The next big step in gravitational-wave science is going to be the construction of next-generation detectors like LISA, CE, and ET. These detectors will detect signals with much higher accuracy, which means the numerical relativity waveforms used to build and calibrate various gravitational-wave models will also need to improve. This thesis is about pushing our code, SpEC, as far as possible to see how accurate we can get, and fixing issues that might prevent us from reaching the accuracy required by the next generation of detectors.

In this thesis, I highlight two improvements made to SpEC with this goal in mind. First, I describe work that makes the black hole spin calculations much faster. The spin calculations were becoming very slow near the merger, when the horizons were highly deformed, and they would only get worse as we increased the resolution of our simulations. The new algorithm is much faster and also opens up paths to future improvements if required. Second, I study various sources of errors in SpEC binary black hole simulations and how to reduce them. We find that relatively targeted changes can improve waveform accuracy by around two orders of magnitude without significantly increasing simulation cost. We also identify some more fundamental limitations that should be considered as we do more high-resolution simulations. Finally, I show preliminary results from an ongoing project demonstrating that, for simple systems, the improvements discussed in this thesis can reach the accuracy required by LISA.

Together, these results show that SpEC can be pushed much further than the current catalog simulations. They also provide a clearer path to producing high-accuracy numerical relativity waveforms needed for LISA and other next-generation gravitational-wave detectors.

PUBLISHED CONTENT AND CONTRIBUTIONS

- [1] Himanshu Chaudhary, Robert Owen, Mark A. Scheel, and Saul A. Teukolsky. “Finding black hole spins efficiently during a numerical binary evolution”. To be submitted. 2026.
Himanshu Chaudhary implemented the algorithm in SpEC, produced the timing and scaling analysis, generated the figures, and wrote the manuscript draft.
- [2] Himanshu Chaudhary, Mark A. Scheel, Robert Owen, and Saul A. Teukolsky. “How accurate is SpEC? And what limits its accuracy?” To be submitted. 2026.
Himanshu Chaudhary studied the sources of error in SpEC binary-black-hole evolutions, generated the figures, and wrote the manuscript draft.

TABLE OF CONTENTS

Acknowledgements	iii
Abstract	iv
Published Content and Contributions	v
Table of Contents	v
List of Illustrations	viii
List of Tables	xiii
Chapter I: Introduction	1
Chapter II: Introduction to Numerical Relativity	8
2.1 Main equations of GR	8
2.2 Gravitational waves	10
2.3 Black holes	10
2.4 Numerical Relativity	12
2.5 Initial data	18
Chapter III: Spectral methods and Finite precision errors	23
3.1 Spectral Methods	24
3.2 Floating-point errors	27
Chapter IV: Finding black hole spins efficiently during a numerical binary evolution	31
4.1 Abstract	31
4.2 Introduction	31
4.3 AKV spin definition	32
4.4 Current AKV algorithm	35
4.5 Sparse AKV algorithm	36
4.6 Future improvements	40
4.7 Conclusion	42
4.8 Timing data PC	43
4.9 Matrix-free version of the AKV algorithm	45
4.10 Speed difference between finding the smallest and the largest eigen- values	46
Chapter V: How accurate is SpEC?	
And what limits its accuracy?	50
5.1 Abstract	50
5.2 Introduction	50
5.3 Simulation Setup and Spec Background	52
5.4 Accuracy metrics in SpEC	58
5.5 Improvements to SpEC	62
5.6 What limits our accuracy?	74
5.7 Conclusions	79
5.8 Generalized Harmonic System	81

5.9 Glitches due to grid changes	81
5.10 Single black hole tests	82
5.11 Errors in spectral derivatives	82
5.12 Parameters of the systems	84
Chapter VI: High-Accuracy BBH Simulations	91
6.1 Parameters of the BBH Systems	91
6.2 Results	92
6.3 Conclusions	96

LIST OF ILLUSTRATIONS

<i>Number</i>	<i>Page</i>
2.1 Lapse and shift in the 3+1 decomposition	14
3.1 This figure shows how the absolute value of the coefficients ($ a_n $) decays in the Chebyshev approximation of $f(x) = \sin(x) + \cos(x)$. We can see that, for large values of n , the decay is exponential as expected.	25
3.2 This figure shows how the error in our Chebyshev approximation of $f(x) = \sin(x) + \cos(x)$ decreases, as we increase the number of polynomials used. Note that the error decreases fairly uniformly across the interval in x . This mostly uniform distribution of errors is one of the main advantages of using Chebyshev polynomials.	26
3.3 This figure shows how the absolute value of the coefficients ($ a_n $) decays in the Chebyshev approximation of $f(x) = \sin(20x) + \cos(20x)$. Note that this time the exponential decay of the coefficients starts much later compared to Fig.(3.1), because we have a more variable function. In particular, we can see that the exponential decay starts approximately after $N \approx 20$, which matches our intuition: the asymptotic exponential decay starts once we have resolved all the dominant frequencies.	27
3.4 This figure shows how the absolute value of the coefficients ($ a_n $) decays in the Chebyshev approximation of $f(x) = \sin(x) + \cos(x)$. This time we are using two different precisions, orange is for the standard double precision with 64 bits, while the blue dots are for extended precision with more bits. We can see that the spectral coefficients stop decaying after $n = 15$ because of the floating-point errors.	28
4.1 Steps in the new AKV algorithm using LU decomposition and ARPACK. We first do the one-time setup, which requires assembling the matrices M and B and computing the LU decomposition of $(M - \sigma B)$. ARPACK needs the action of $(M - \sigma B)^{-1}B$ on the vector x provided by ARPACK. To construct this vector, we first apply B to obtain $v = Bx$, and then solve $(M - \sigma B)y = v$ for y using the LU factors. . .	39

- 4.2 Dependence of the condition number of the matrix $M - \sigma B$ on σ for different L . The matrices were generated for a test Kerr-Schild metric with spin equal to (0,0,0.5) and mass equal to 1. Note that the condition number also gets worse as the number of modes L required to represent the horizon increases. The behavior will depend on the shape of the black hole horizons, but for the systems we tested, having a small σ always showed this blowup because most of the time one of the eigenvalues is close to zero. 40
- 4.3 Number of LU solves required by ARPACK to converge. Note that ARPACK needs one LU solve per iteration, so this also represents the number of iterations to converge. Although this figure is for the same simple Kerr-Schild spacetime as Fig.(4.2), the number of iterations rarely goes beyond 30 even for much more complicated systems. On the other hand, not having a good enough value of σ makes the convergence very slow, as we can see from the graph. For the BBH simulations, we use $\sigma = 0.1$, which we have found has a good balance between the number of iterations required by ARPACK and the condition number of the matrix $M - \sigma B$ 41
- 4.4 Scaling of various parts of the algorithm with the highest spherical harmonic mode L . The runs were performed on a high-performance CPU cluster. The solid dots are the real data, while the dotted lines show the best fits. The fits were weighted such that the measurements at higher L have more weight as they are less noisy. The legends include the scaling from these fits. We see that the scaling of matrix generation and `dggev` is slightly worse than the theoretical expectation of $O(L^4)$ and $O(L^6)$, respectively. Note that the new AKV algorithm using ARPACK + LU decomposition (purple + black lines) is orders of magnitude faster than the current AKV algorithm using `dggev` (blue line). By a large margin, the current bottleneck is matrix generation (red line). 43

- 4.5 Scaling of various parts of the algorithm with the highest spherical harmonic mode L . The runs were performed on a consumer-grade processor (Intel(R) Core(TM) i9-10900X CPU @ 3.70GHz). Labels and conclusions are the same as for Fig.(4.4). Note that the LU decomposition is $\approx 10 \times$ slower while the matrix generation is $\approx 2 \times$ faster when compared to running on an HPC system, as shown in Fig.(4.4). 44
- 4.6 Matrix-free workflow for ARPACK in the generalized shift-invert case. Instead of explicitly forming M and B , we use the functions f_M and f_B , which are available in SpEC. At each iteration, ARPACK supplies a vector x , we compute $v = f_B(x)$, and then solve $(f_M - \sigma f_B)(y) = v$ using an iterative solver. The result $y = (M - \sigma B)^{-1} Bx$ is returned to ARPACK for the next iteration. 46
- 5.1 The $z = 0$ cross-section of the subdomains used in SpEC for simulating binary black holes. Spherical subdomains in red are the SphereAs, with SphereA0 being the one that contains the horizon of the black hole A. Similarly, green subdomains are SphereBs, and they contain the black hole B. The outermost spherical subdomains in blue are SphereCs, with SphereC0 being the smallest of these. Other subdomains, not colored in the figure, are distorted cylinders and filled cylinders that join the inner SphereA and SphereB subdomains with the outer SphereC subdomains. These subdomains are further divided into smaller subdomains by the AMR algorithm as required. Note that this figure is not to scale, and in general, a typical SphereC subdomain is an order of magnitude larger than a typical SphereA and SphereB subdomain. 54
- 5.2 Constraint violations before and after changing the ODE tolerance. The left panels show the L_2 norm over the full simulation domain, while the right panels show the L_∞ norm in SphereA0. The Old simulations show noisy high-resolution behavior near the horizon, while the simulations with the ODE tolerance fix show cleaner convergence. 63
- 5.3 Power spectrum of ψ in SphereA0 for a level 5 Old run. The curves a_i are the power monitors of the radial basis functions. The lower modes show exponential convergence, while the highest modes are dominated by noise. 65

5.4	Power spectrum of ψ in SphereA0 for a level 5 run with the ODE tolerance fix. Compared to the Old run, the highest retained modes have less noise, and the AMR algorithm drops noisy modes.	66
5.5	$\mathcal{E}_{\text{GH}}$ on SphereC6 vs. time for runs at levels 1-5.	66
5.6	Normalized constraint energy density in SphereC6 before and after the ODE tolerance fix. The ODE tolerance fix improves the near-horizon convergence, but the normalized constraints in the wave-extraction region still show poor convergence with level. This motivates adding both more resolution and more constraint damping in the outer subdomains.	67
5.7	$\widehat{\mathcal{E}}_{\text{GH}}$ vs. time in SphereC28. Red line is the level 5 run with the ODE tolerance fix, and we can see that the constraints are still going down and haven't saturated yet. Blue line is the level 5 New run, which has more constraint damping, and we can see that the constraints saturate much earlier.	67
5.8	$\widehat{\mathcal{E}}_{\text{GH}}$ vs time for SphereC6 in the New simulations.	69
5.9	Power spectrum of ψ in SphereC6 for a level 5 run with ODE fix. . .	70
5.10	Power spectrum of ψ in SphereC6 for a level 5 run with ODE fix and AMR changes.	70
5.11	Correlation between finite-radius waveform differences and differences in the black-hole trajectories. The horizontal axis at the bottom is the time-integrated absolute difference in the coordinate distance of black hole A from the origin between two levels. The horizontal axis at the top shows the pair of levels used for computing the waveform difference. Different colors represent the waveform difference for various modes. The red dots represent the waveform difference for all the modes, and they overlap with the waveform difference of the (2,2) mode, because that is the dominant mode. In both the Old and New simulations, larger trajectory differences correspond to larger waveform differences. The New simulations reduce waveform differences by two orders of magnitude.	72
5.12	Spectral coefficients of ψ in the subdomain SphereA0 for a Level 6 simulation.	75
5.13	Spectral coefficients of κ in the subdomain SphereA0 for a Level 6 simulation.	75

5.14	Power spectrum of κ in SphereC6 for a level 5 run with ODE fix and AMR changes.	78
5.15	$\widehat{\mathcal{E}}_{\text{GH}}$ for New runs in the SphereC22.	79
5.16	Single-black-hole tests showing errors in the outermost subdomains. The left panel shows that the error in κ increases when the asymptotic values of the constraint damping parameters are increased. The right panel shows that the three-index constraint reaches a roughly stationary floor near 10^{-15} until the error in κ becomes large enough to dominate the constraint calculation.	80
5.17	Normalized constraint-violation heatmaps for the Old simulation from the main branch and the New simulation with all the improvements. The vertical axis is simulation time, and the horizontal axis labels the various subdomains from the inner region to the outer boundary. The vertical grey areas are the location of the black hole horizons, which are excised from the simulation domain. The bright diagonal feature is the junk-radiation pulse propagating outward through the domain. Note how the constraints in the outer subdomains are much better for the New run once the junk radiation leaves the domain. . . .	83
6.1	Run 09 $L_2(\mathcal{E}_{\text{GH}})$ over the simulation domain.	94
6.2	Run 10 $L_2(\mathcal{E}_{\text{GH}})$ over the simulation domain.	94
6.3	Run 11 $L_2(\mathcal{E}_{\text{GH}})$ over the simulation domain.	95
6.4	Run 12 $L_2(\mathcal{E}_{\text{GH}})$ over the simulation domain.	95
6.5	Run 09 CCE and finite-radius waveform differences versus integrated distance-from-origin difference.	96
6.6	Run 09 finite-radius waveform difference versus integrated distance-from-origin difference.	97
6.7	Run 10 finite-radius waveform difference versus integrated distance-from-origin difference.	97
6.8	Run 11 finite-radius waveform difference versus integrated distance-from-origin difference. Note that Run 11 has not completed ringdown for level 6, so the waveform differences will be slightly lower than the final value.	98
6.9	Run 12 finite-radius waveform difference versus integrated distance-from-origin difference. Note that Run 12 has not completed ringdown for level 6, so the waveform differences will be slightly lower than the final value.	99

LIST OF TABLES

<i>Number</i>	<i>Page</i>
4.1 Timing data for matrix generation, LU decomposition, ARPACK iterations, and LAPACK's <code>dggev</code> on Caltech HPC (Intel(R) Xeon(R) Platinum 8276 CPU @ 2.20GHz). The column showing the ARPACK timing also includes the time taken by the LU solves required for each iteration.	42
4.2 Timing data for matrix generation, ARPACK, LU decomposition, and LAPACK's <code>dggev</code> on PC (Intel(R) Core(TM) i9-10900X CPU @ 3.70GHz). The column showing the ARPACK timing also includes the time taken by the LU solves required for each iteration.	45
5.1 Finite-radius waveform difference	71
5.2 Initial-data and relaxed parameters for the simulations used in this paper. The New and Old/ODE fix rows are measured after PBandJ.	82
6.1 Simulation parameters.	92
6.2 Run 09 waveform differences for both finite-radius and CCE waveforms, with each level compared to level 6. The finite-radius values are at $R = 300M$. Note that the CCE and finite-radius waveform differences agree quite well until we reach levels 5 and 6, where the CCE waveform difference is an order of magnitude larger. Although this is not the case for lower levels here, in general, we expect that the CCE waveform differences will be slightly larger because they go through CCE evolution, which introduces more errors.	93
6.3 Finite-radius waveform differences, with each level compared to level 6. Runs 09 and 10 have been fully completed, while runs 11 and 12 are still going through the ringdown. Therefore, we expect that the waveform difference values for runs 11 and 12 will increase when we have the full waveform. The full CCE evolution will introduce further errors, but we still expect that all these simulations will have a waveform difference of the order 10^{-8}	93

Chapter 1

INTRODUCTION

First formulated in 1915, General Relativity remains the most successful theory of gravity we have. Since then, it has passed a plethora of tests, but most of those were tests in the weak-field limits of GR. We know that GR can not be the final theory, and it should start to fail at some point. And we strongly suspect that those failures will first show up in regions of very strong gravitational fields.

The Einstein equations that describe gravity’s behavior are highly non-linear. This non-linearity makes them very hard to reason about analytically, and only a handful of full solutions are known. They also predict that situations with very strong gravitational fields and motion will produce gravitational waves (GW). Gravitational waves, unlike their electromagnetic counterparts, are not affected strongly by the interstellar materials and thus travel very far. Gravitational waves produced by the merger of binary compact objects are among the most energetic phenomena in the universe, and they provide a unique window into the strong-gravity regime we want to probe.

The highly non-linear nature of GR makes it very hard to study the motion of gravitational fields in highly dynamic situations, such as the merger of two black holes. The most successful way to study GR in a highly non-linear regime is to use computers and solve these equations numerically, a field of study aptly named Numerical Relativity (NR). The first successful NR simulation of two black holes orbiting each other was performed in 2005 [1]. As the first step in a long journey, these early simulations were neither accurate nor long enough to be used by the detectors in LVK (LIGO-VIRGO-KAGRA) [2, 3]. Since then, the field has progressed a lot, and we now have catalogs that contain thousands of these simulations [4–16].

Data analysis for a single LIGO event can use millions of waveforms densely sampled across the parameter space. This makes directly using NR for data analysis unfeasible, because generating a single waveform using NR can take months. For data analysis, we use other GW models like phenomenological models [17–21] and Effective-One-Body (EOB) models [22–28], which rely on NR waveforms for calibration and validation. The newest class of waveform models is surrogate models

[29–33], which cleverly interpolate in parameter space between NR simulations while mostly keeping the accuracy of the underlying simulations.

The next big step in the field of gravitational waves is the construction of the next generation of detectors, like LISA, ET, CE, etc [34–37]. These detectors will be much more accurate, which means our gravitational wave surrogate models have to improve greatly. Current NR catalogs, essential for generating these surrogates, do not match the accuracy requirements of these next-generation detectors [38, 39]. This thesis mostly focuses on solving Einstein’s equations efficiently and accurately on supercomputers.

In Chapter 2, we will first give a quick introduction to numerical relativity to establish conventions and define equations used later. Chapter 3 is a short introduction to the relevant parts of spectral methods and finite precision arithmetic that affect our NR simulations.

Chapter 4 is about the work I did to speed up black hole spin computations. The previous method used to compute the spin was very slow for highly deformed apparent horizons, making some of these simulations infeasible with current computer budgets. We implemented a new algorithm based on sparse solvers that reduced the cost of spin computations by an order of magnitude. The new algorithm also moves the bottleneck from an external library to our own code, which means we can get further speedup if required by better parallelization. We also outline a matrix-free version of this algorithm, which could be even faster once we better understand the system and can construct effective preconditioners.

Chapter 5 is the principal part of the thesis and describes the work I did to make our numerical relativity simulations much more accurate and have better convergence behavior. The main goal was to push the limits of the SpEC (Spectral Einstein Code) and see what limits us from doing LISA-accurate simulations. This was a focused study aimed at understanding the sources of error and the fundamental limits of using spectral methods for binary black hole simulations. We describe multiple improvements made to SpEC, leading to a two-order-of-magnitude improvement in the waveform mismatches without any significant increase in the computational cost. These improvements pave the way for performing LISA-accurate simulations using SpEC.

Chapter 6 is about the ongoing work we have been doing to produce the first set of LISA-accurate simulations using SpEC. We are doing 12 simulations: 4 non-

spinning systems with mass ratios 1–4 and 8 aligned-spin simulations that cover spins up to 0.8. The two simulations that have already been completed show that we can reach the required accuracy very efficiently. For more complex simulations, we will need to leverage the scaling offered by next-generation codes such as SpECTRE [40] to reduce the wall time.

References

- [1] Frans Pretorius. “Evolution of Binary Black-Hole Spacetimes”. In: *Phys. Rev. Lett.* 95.12, 121101 (Sept. 2005), p. 121101. DOI: 10.1103/PhysRevLett.95.121101. arXiv: gr-qc/0507014 [gr-qc].
- [2] LIGO Scientific Collaboration, J. Aasi, B. P. Abbott, et al. “Advanced LIGO”. In: *Classical and Quantum Gravity* 32.7, 074001 (Apr. 2015), p. 074001. DOI: 10.1088/0264-9381/32/7/074001. arXiv: 1411.4547 [gr-qc].
- [3] F. Acernese, M. Agathos, K. Agatsuma, et al. “Advanced Virgo: a second-generation interferometric gravitational wave detector”. In: *Classical and Quantum Gravity* 32.2, 024001 (Jan. 2015), p. 024001. DOI: 10.1088/0264-9381/32/2/024001. arXiv: 1408.3978 [gr-qc].
- [4] Mark A. Scheel et al. “The SXS collaboration’s third catalog of binary black hole simulations”. In: *Classical and Quantum Gravity* 42.19, 195017 (Oct. 2025), p. 195017. DOI: 10.1088/1361-6382/adfd34. arXiv: 2505.13378 [gr-qc].
- [5] Laura Cadonati, Benjamin Aylott, John G. Baker, et al. “Status of NINJA: the Numerical INjection Analysis project”. In: *Classical and Quantum Gravity* 26.11, 114008 (June 2009), p. 114008. DOI: 10.1088/0264-9381/26/11/114008. arXiv: 0905.4227 [gr-qc].
- [6] P. Ajith, Michael Boyle, Duncan A. Brown, et al. “The NINJA-2 catalog of hybrid post-Newtonian/numerical-relativity waveforms for non-precessing black-hole binaries”. In: *Classical and Quantum Gravity* 29.12, 124001 (June 2012), p. 124001. DOI: 10.1088/0264-9381/29/12/124001. arXiv: 1201.5319 [gr-qc].
- [7] Ian Hinder, Alessandra Buonanno, Michael Boyle, et al. “Error-analysis and comparison to analytical models of numerical waveforms produced by the NRAR Collaboration”. In: *Classical and Quantum Gravity* 31.2, 025012 (Jan. 2013), p. 025012. DOI: 10.1088/0264-9381/31/2/025012. arXiv: 1307.5307 [gr-qc].
- [8] Karan Jani, James Healy, James A. Clark, et al. “Georgia tech catalog of gravitational waveforms”. In: *Classical and Quantum Gravity* 33.20, 204001 (Oct. 2016), p. 204001. DOI: 10.1088/0264-9381/33/20/204001. arXiv: 1605.03204 [gr-qc].

- [9] Deborah Ferguson, Evelyn Allsup, Surendra Anne, et al. “Second MAYA catalog of binary black hole numerical relativity waveforms”. In: *Phys. Rev. D* 112.4, 044043 (Aug. 2025), p. 044043. doi: 10.1103/gk7x-9cds. arXiv: 2309.00262 [gr-qc].
- [10] James Healy et al. “The RIT binary black hole simulations catalog”. In: *Classical and Quantum Gravity* 34.22, 224001 (Nov. 2017), p. 224001. doi: 10.1088/1361-6382/aa91b1. arXiv: 1703.03423 [gr-qc].
- [11] James Healy, Carlos O. Lousto, Jacob Lange, et al. “Second RIT binary black hole simulations catalog and its application to gravitational waves parameter estimation”. In: *Phys. Rev. D* 100.2, 024021 (July 2019), p. 024021. doi: 10.1103/PhysRevD.100.024021. arXiv: 1901.02553 [gr-qc].
- [12] James Healy and Carlos O. Lousto. “Third RIT binary black hole simulations catalog”. In: *Phys. Rev. D* 102.10, 104018 (Nov. 2020), p. 104018. doi: 10.1103/PhysRevD.102.104018. arXiv: 2007.07910 [gr-qc].
- [13] James Healy and Carlos O. Lousto. “Fourth RIT binary black hole simulations catalog: Extension to eccentric orbits”. In: *Phys. Rev. D* 105.12, 124010 (June 2022), p. 124010. doi: 10.1103/PhysRevD.105.124010. arXiv: 2202.00018 [gr-qc].
- [14] E. A. Huerta, Roland Haas, Sarah Habib, et al. “Physics of eccentric binary black hole mergers: A numerical relativity perspective”. In: *Phys. Rev. D* 100.6, 064003 (Sept. 2019), p. 064003. doi: 10.1103/PhysRevD.100.064003. arXiv: 1901.07038 [gr-qc].
- [15] Eleanor Hamilton, Edward Fauchon-Jones, Mark Hannam, et al. “Catalog of precessing black-hole-binary numerical-relativity simulations”. In: *Phys. Rev. D* 109.4, 044032 (Feb. 2024), p. 044032. doi: 10.1103/PhysRevD.109.044032. arXiv: 2303.05419 [gr-qc].
- [16] Alireza Rashti, Rossella Gamba, Koustav Chandra, et al. “Binary black hole waveforms from high-resolution GR-Athena++ simulations”. In: *Phys. Rev. D* 111.10, 104078 (May 2025), p. 104078. doi: 10.1103/n5pz-qv3x. arXiv: 2411.11989 [gr-qc].
- [17] Cecilio García-Quirós et al. “Accelerating the evaluation of inspiral-merger-ringdown waveforms with adapted grids”. In: *Classical and Quantum Gravity* 38.1, 015006 (Jan. 2021), p. 015006. doi: 10.1088/1361-6382/abc36e. arXiv: 2001.10897 [gr-qc].
- [18] Jonathan E. Thompson, Edward Fauchon-Jones, Sebastian Khan, et al. “Modeling the gravitational wave signature of neutron star black hole coalescences”. In: *Phys. Rev. D* 101.12, 124059 (June 2020), p. 124059. doi: 10.1103/PhysRevD.101.124059. arXiv: 2002.08383 [gr-qc].

- [19] Tim Dietrich, Sebastian Khan, Reetika Dudi, et al. “Matter imprints in waveform models for neutron star binaries: Tidal and self-spin effects”. In: *Phys. Rev. D* 99.2, 024029 (Jan. 2019), p. 024029. doi: 10.1103/PhysRevD.99.024029. arXiv: 1804.02235 [gr-qc].
- [20] Sebastian Khan et al. “Phenomenological model for the gravitational-wave signal from precessing binary black holes with two-spin effects”. In: *Phys. Rev. D* 100.2, 024059 (July 2019), p. 024059. doi: 10.1103/PhysRevD.100.024059. arXiv: 1809.10113 [gr-qc].
- [21] Sebastian Khan et al. “Including higher order multipoles in gravitational-wave models for precessing binary black holes”. In: *Phys. Rev. D* 101.2, 024056 (Jan. 2020), p. 024056. doi: 10.1103/PhysRevD.101.024056. arXiv: 1911.06050 [gr-qc].
- [22] A. Buonanno and T. Damour. “Effective one-body approach to general relativistic two-body dynamics”. In: *Phys. Rev. D* 59.8, 084006 (Apr. 1999), p. 084006. doi: 10.1103/PhysRevD.59.084006. arXiv: gr-qc/9811091 [gr-qc].
- [23] Alessandra Buonanno and Thibault Damour. “Transition from inspiral to plunge in binary black hole coalescences”. In: *Phys. Rev. D* 62.6, 064015 (Sept. 2000), p. 064015. doi: 10.1103/PhysRevD.62.064015. arXiv: gr-qc/0001013 [gr-qc].
- [24] Alessandra Buonanno, Yanbei Chen, and Thibault Damour. “Transition from inspiral to plunge in precessing binaries of spinning black holes”. In: *Phys. Rev. D* 74.10, 104005 (Nov. 2006), p. 104005. doi: 10.1103/PhysRevD.74.104005. arXiv: gr-qc/0508067 [gr-qc].
- [25] Thibault Damour, Piotr Jaranowski, and Gerhard Schäfer. “Determination of the last stable orbit for circular general relativistic binaries at the third post-Newtonian approximation”. In: *Phys. Rev. D* 62.8, 084011 (Oct. 2000), p. 084011. doi: 10.1103/PhysRevD.62.084011. arXiv: gr-qc/0005034 [gr-qc].
- [26] Thibault Damour. “Introductory Lectures on the Effective One Body Formalism”. In: *International Journal of Modern Physics A* 23.8 (Jan. 2008), pp. 1130–1148. doi: 10.1142/S0217751X08039992. arXiv: 0802.4047 [gr-qc].
- [27] Thibault Damour. “Introductory Lectures on the Effective One Body Formalism”. In: *International Journal of Modern Physics A* 23.8 (Jan. 2008), pp. 1130–1148. doi: 10.1142/S0217751X08039992. arXiv: 0802.4047 [gr-qc].
- [28] Antoni Ramos-Buades, Alessandra Buonanno, Héctor Estellés, et al. “Next generation of accurate and efficient multipolar precessing-spin effective-one-body waveforms for binary black holes”. In: *Phys. Rev. D* 108.12, 124037

- (Dec. 2023), p. 124037. DOI: 10.1103/PhysRevD.108.124037. arXiv: 2303.18046 [gr-qc].
- [29] Jonathan Blackman, Scott E. Field, Mark A. Scheel, et al. “A Surrogate model of gravitational waveforms from numerical relativity simulations of precessing binary black hole mergers”. In: *Phys. Rev. D* 95.10, 104023 (May 2017), p. 104023. DOI: 10.1103/PhysRevD.95.104023. arXiv: 1701.00550 [gr-qc].
 - [30] Jonathan Blackman, Scott E. Field, Chad R. Galley, et al. “Fast and Accurate Prediction of Numerical Relativity Waveforms from Binary Black Hole Coalescences Using Surrogate Models”. In: *Phys. Rev. Lett.* 115.12, 121102 (Sept. 2015), p. 121102. DOI: 10.1103/PhysRevLett.115.121102. arXiv: 1502.07758 [gr-qc].
 - [31] Vijay Varma, Scott E. Field, Mark A. Scheel, et al. “Surrogate model of hybridized numerical relativity binary black hole waveforms”. In: *Phys. Rev. D* 99.6, 064045 (Mar. 2019), p. 064045. DOI: 10.1103/PhysRevD.99.064045. arXiv: 1812.07865 [gr-qc].
 - [32] Jonathan Blackman, Scott E. Field, Mark A. Scheel, et al. “Numerical relativity waveform surrogate model for generically precessing binary black hole mergers”. In: *Phys. Rev. D* 96.2, 024058 (July 2017), p. 024058. DOI: 10.1103/PhysRevD.96.024058. arXiv: 1705.07089 [gr-qc].
 - [33] Vijay Varma, Scott E. Field, Mark A. Scheel, et al. “Surrogate models for precessing binary black hole simulations with unequal masses”. In: *Physical Review Research* 1.3, 033015 (Oct. 2019), p. 033015. DOI: 10.1103/PhysRevResearch.1.033015. arXiv: 1905.09300 [gr-qc].
 - [34] Pau Amaro-Seoane, Heather Audley, Stanislav Babak, et al. “Laser Interferometer Space Antenna”. In: *arXiv e-prints*, arXiv:1702.00786 (Feb. 2017), arXiv:1702.00786. DOI: 10.48550/arXiv.1702.00786. arXiv: 1702.00786 [astro-ph.IM].
 - [35] Matthew Evans, Alessandra Corsi, Chaitanya Afle, et al. “Cosmic Explorer: A Submission to the NSF MPSAC ngGW Subcommittee”. In: *arXiv e-prints*, arXiv:2306.13745 (June 2023), arXiv:2306.13745. DOI: 10.48550/arXiv.2306.13745. arXiv: 2306.13745 [astro-ph.IM].
 - [36] Michele Maggiore, Chris Van Den Broeck, Nicola Bartolo, et al. “Science case for the Einstein telescope”. In: *J. Cosmology Astropart. Phys.* 2020.3, 050 (Mar. 2020), p. 050. DOI: 10.1088/1475-7516/2020/03/050. arXiv: 1912.02622 [astro-ph.CO].
 - [37] Adrian Abac, Raul Abramo, Simone Albanesi, et al. “The Science of the Einstein Telescope”. In: *J. Cosmology Astropart. Phys.* 2026.3, 081 (Mar. 2026), p. 081. DOI: 10.1088/1475-7516/2026/03/081. arXiv: 2503.12263 [gr-qc].

- [38] Michael Pürrer and Carl-Johan Haster. “Gravitational waveform accuracy requirements for future ground-based detectors”. In: *Physical Review Research* 2.2, 023151 (May 2020), p. 023151. DOI: 10.1103/PhysRevResearch.2.023151. arXiv: 1912.10055 [gr-qc].
- [39] LISA Consortium Waveform Working Group, Niayesh Afshordi, Sarp Akçay, et al. “Waveform modelling for the Laser Interferometer Space Antenna”. In: *Living Reviews in Relativity* 28.1, 9 (Oct. 2025), p. 9. DOI: 10.1007/s41114-025-00056-1. arXiv: 2311.01300 [gr-qc].
- [40] Nils Deppe et al. *SpECTRE*. Version 2026.04.01. Apr. 2026. DOI: 10.5281/zenodo.19373346. URL: <https://doi.org/10.5281/zenodo.19373346>.

Chapter 2

INTRODUCTION TO NUMERICAL RELATIVITY

In this chapter, the first section refreshes the basics of GR and establishes the notation. The next section covers the basics of Numerical Relativity, which deals with how to reformulate the Einstein equations so that they can be solved efficiently on a computer.

2.1 Main equations of GR

Throughout this thesis, we will work in geometric units where $G = c = 1$. The metric convention will be $(-, +, +, +)$ same as the one used in [1, 2]. Latin letters $a - h$ and $o - z$ will denote the 4-dimensional spacetime indices, while the letters $i - n$ will be used for 3-dimensional spatial indices. Quantities associated with the spacetime metric g_{ab} will also have have ⁽⁴⁾.

The fundamental starting point of GR is the spacetime metric tensor g_{ab} , which allows us to compute an invariant distance between any two nearby events.

$$ds^2 = g_{ab} dx^a dx^b \quad (2.1)$$

Einstein's equation relates the derivatives of this metric to the energy density of matter in the spacetime.

$$G_{ab} = 8\pi T_{ab} \quad (2.2)$$

The G_{ab} is called the Einstein tensor, and it describes the curvature of the spacetime, while the T_{ab} is the standard stress energy tensor. My work has primarily dealt with binary black hole systems in vacuum, which means we work with $T_{ab} = 0$.

The equations of GR are usually written in a coordinate-independent tensorial form. The partial derivative of a tensor is generally not a tensor. Therefore, we use another notion of derivative called the covariant derivative:

$$\nabla_c T^a_b = \partial_c T^a_b + {}^{(4)}\Gamma^a_{dc} T^d_b - {}^{(4)}\Gamma^d_{bc} T^a_d \quad (2.3)$$

${}^{(4)}\Gamma^a_{bc}$ is called the Christoffel symbol and is defined as:

$${}^{(4)}\Gamma^a_{bc} = g^{ad} {}^{(4)}\Gamma_{dbc} = \frac{1}{2} g^{ad} (\partial_c g_{db} + \partial_b g_{dc} - \partial_d g_{bc}) \quad (2.4)$$

The only way to differentiate between a complex coordinate choice and true space-time curvature is to compute the Riemann curvature tensor ${}^{(4)}R^a{}_{bcd}$. The Riemann curvature tensor is defined as:

$$\nabla_a \nabla_b v_c - \nabla_b \nabla_a v_c = v_d {}^{(4)}R^d{}_{cab} \quad (2.5)$$

or equivalently as,

$${}^{(4)}R^a{}_{bcd} = \partial_c {}^{(4)}\Gamma^a{}_{bd} - \partial_d {}^{(4)}\Gamma^a{}_{bc} + {}^{(4)}\Gamma^a{}_{ec} {}^{(4)}\Gamma^e{}_{bd} - {}^{(4)}\Gamma^a{}_{ed} {}^{(4)}\Gamma^e{}_{bc}. \quad (2.6)$$

${}^{(4)}R^a{}_{bcd}$ is uniformly zero if and only if we have a flat spacetime. Therefore, a non-zero Riemann tensor implies the existence of non-zero gravitational curvature, or equivalently, non-zero gravitational tidal effects.

We form the Ricci tensor and Ricci scalar by contracting the Riemann tensor:

$${}^{(4)}R_{ab} = {}^{(4)}R^c{}_{acb} \quad (2.7)$$

$${}^{(4)}R = {}^{(4)}R^a{}_a. \quad (2.8)$$

Another useful tensor defined using the Riemann tensor is the Weyl tensor:

$$\begin{aligned} {}^{(4)}C_{abcd} = & {}^{(4)}R_{abcd} - \frac{1}{2} \left(g_{ac} {}^{(4)}R_{bd} - g_{ad} {}^{(4)}R_{bc} - g_{bc} {}^{(4)}R_{ad} + g_{bd} {}^{(4)}R_{ac} \right) \\ & + \frac{1}{6} (g_{ac} g_{bd} - g_{ad} g_{bc}) {}^{(4)}R. \end{aligned} \quad (2.9)$$

The Weyl tensor has all the same algebraic symmetries as the Riemann tensor, in addition to being trace-free. Given the metric, the Weyl tensor together with the Ricci tensor contains the same information as the full Riemann tensor, which describes how spacetime curvature produces local tidal deformations. In this decomposition, the Weyl tensor encodes the trace-free part of the curvature, which changes the shape of an infinitesimal body, while the Ricci tensor encodes the trace part, which controls its change in volume.

The Einstein tensor G_{ab} is defined using,

$$G_{ab} = R_{ab} - \frac{1}{2} g_{ab} R. \quad (2.10)$$

In vacuum we have that $T_{ab} = 0$, which, using Einstein's equation, implies that $G_{ab} = 0$. By contracting with the RHS of Eq.(2.10), we can see that in vacuum $R_{ab} = 0$. From Eq.(2.9), we can see that in vacuum both the Riemann and the Weyl tensor are the same. In practical terms, this means that in a vacuum, gravity can distort the shape of an infinitesimal body, but it produces no local Ricci-driven change in its volume.

2.2 Gravitational waves

Gravitational waves are excitations in the gravitational fields that travel at the speed of light. Unlike their EM counterparts, gravitational waves interact very weakly with the interstellar material. Given the tolerances of the current gravitational wave detectors, we can treat them as if they were traveling through a vacuum. Far away from the source, we can model these waves as a perturbation (h_{ab}) on top of the flat spacetime metric.

$$g_{ab} = \eta_{ab} + h_{ab} \quad (2.11)$$

where $|h_{ab}| \ll 1$.

The equations become a lot simpler if we define the trace-reversed version of the perturbation $\bar{h}_{ab}$,

$$\bar{h}_{ab} \equiv h_{ab} - \frac{1}{2} h^c{}_c \eta_{ab}. \quad (2.12)$$

Under our assumption of vacuum and choice of Lorenz gauge $\nabla_b \bar{h}^{ab} = 0$, the Einstein equations become:

$$\nabla^c \nabla_c \bar{h}_{ab} = 0. \quad (2.13)$$

The Eq.(2.13) has 10 free components, but 4 of these gauge freedoms are limited by the Lorenz gauge, and another 4 are limited by working in transverse-traceless or TT gauge ($h^T{}_0 = 0$, $h^{TTa}{}_a = 0$). This leaves us with 2 real degrees of freedom that represent the two polarizations of the gravitational waves h_+ and $h_\times$.

$$\bar{h}^{TT}_{\mu\nu} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & h_+ & h_\times & 0 \\ 0 & h_\times & -h_+ & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (2.14)$$

2.3 Black holes

For a spherically symmetric spacetime, we have the Schwarzschild solution,

$$ds^2 = - \left(1 - \frac{2M}{r}\right) dt^2 + \left(1 - \frac{2M}{r}\right)^{-1} dr^2 + r^2 d\theta^2 + r^2 \sin^2 \theta d\phi^2. \quad (2.15)$$

The solution holds for all vacuum regions of a spherically symmetric spacetime, and this is called Birkhoff's theorem. This includes the exterior of non-spinning stars,

planets, and all other spherically symmetric distributions of mass and energy, even if the distribution itself is not static. The radius ($r = 2M$) is called the Schwarzschild radius. If the matter collapses inside this radius, then a black hole forms. The event horizon is located at the Schwarzschild radius, while the curvature singularity is located at the center ($r = 0$).

If we add spin to the spacetime, then we get the Kerr spacetime, which in the Boyer-Lindquist coordinates looks like:

$$ds^2 = - \left(1 - \frac{2Mr}{\Sigma} \right) dt^2 - \frac{4aMr \sin^2 \theta}{\Sigma} dt d\phi + \frac{\Sigma}{\Delta} dr^2 + \Sigma d\theta^2 + \left(r^2 + a^2 + \frac{2a^2 Mr \sin^2 \theta}{\Sigma} \right) \sin^2 \theta d\phi^2, \quad (2.16)$$

where, J is the angular momentum, $a \equiv J/M$, $\Delta \equiv r^2 - 2Mr + a^2$, $\Sigma \equiv r^2 + a^2 \cos^2 \theta$. These coordinates assume that the spin is along the z -axis in the $+\phi$ direction. Note that $\frac{1}{g_{rr}} = \nabla = 0$ has two solutions $r_+ = M + (M^2 - a^2)^{1/2}$ which is the event horizon, and $r_- = M - (M^2 - a^2)^{1/2}$ which is called the inner horizon or the Cauchy horizon. The reason it is called the Cauchy horizon is that the spacetime beyond this horizon cannot be described uniquely as a Cauchy problem from the original initial data. The inner horizon and the region inside it are believed to be not physically relevant because the inner horizon itself is unstable. However, it does pose a serious challenge for numerical relativity simulations, where we have to make sure that the excision surface stays inside the outer horizon and avoids the problematic inner-horizon region. Note that as the spin increases, the two horizons approach each other, making the already complex problem of numerically evolving high-spin systems even harder.

Also, there is no analog of Birkhoff's theorem for a spacetime with spin. This means that not all axisymmetric vacuum spacetimes are Kerr spacetimes. However, under physically reasonable conditions, such spacetimes asymptotically approach the Kerr spacetime as they settle down to equilibrium. The general understanding is that any physical collapse of matter will eventually settle down to the Kerr spacetime after radiating away gravitational waves. This makes the Kerr metric particularly important for Numerical Relativity, but we usually work in another coordinate system called the Kerr-Schild coordinates because they are horizon penetrating.

The Kerr metric in Kerr-Schild coordinates can be written as

$$g_{ab} = \eta_{ab} + 2Hl_al_b, \quad (2.17)$$

where η_{ab} is the flat Minkowski metric, l_a is a null one-form with respect to both η_{ab} and g_{ab} , and

$$H = \frac{Mr^3}{r^4 + a^2z^2}. \quad (2.18)$$

In Cartesian Kerr-Schild coordinates $x^a = (t, x, y, z)$, one commonly writes

$$l_adx^a = dt + \frac{rx + ay}{r^2 + a^2} dx + \frac{ry - ax}{r^2 + a^2} dy + \frac{z}{r} dz. \quad (2.19)$$

Here r is defined implicitly by

$$\frac{x^2 + y^2}{r^2 + a^2} + \frac{z^2}{r^2} = 1. \quad (2.20)$$

Equivalently,

$$r^2 = \frac{1}{2} \left[x^2 + y^2 + z^2 - a^2 + \sqrt{(x^2 + y^2 + z^2 - a^2)^2 + 4a^2z^2} \right]. \quad (2.21)$$

2.4 Numerical Relativity

To get the gravitational waveform from the evolution and merger of two binary black holes, we need to solve an initial data problem. The idea is to start from a configuration where the black holes are a few orbits away from the merger and then evolve the system until they merge and settle down. The two obvious hurdles are:

- Time is just another coordinate in GR, so we need to carefully describe what it means to evolve in time.
- We need to approximate the spacetime of two binary black holes in orbits.

We will first talk about the 3+1 decomposition, where we divide our spacetime into 1 time and 3 spatial directions. The next step is to obtain our spatial metric by projecting the spacetime metric perpendicular to the chosen time direction. This will allow us to talk about evolving the spatial metric forward in time. Then we will talk about getting the initial data, or equivalently, the spatial metric and its first time derivative. Once we know the spatial metric and its first time derivative, we can use the Einstein equations to get the second time derivatives and evolve forward in time.

3+1 decomposition

We start by choosing a global time function t that labels the spacelike surfaces Σ . It is assumed that we can choose these surfaces in such a way that they do not intersect each other. Using our choice of t we can define the lapse function Ω_a as,

$$\Omega_a = \nabla_a t. \quad (2.22)$$

Ω_a is normal to our spacelike surfaces Σ . Its norm α defined by,

$$\|\Omega\|^2 = g^{ab} \nabla_a t \nabla_b t \equiv -\frac{1}{\alpha^2} \quad (2.23)$$

tells us how much time has passed as we move from one spacelike surface to another. The normalized lapse functions can be defined as $\omega_a = \alpha \Omega_a$. Which allows us to define the unit normal vector to the spacelike surfaces,

$$n^a = -g^{ab} \omega_b. \quad (2.24)$$

n^a is normalized and timelike, but construction and allows us to define the spatial metric γ_{ab} on each surface Σ from the spacetime metric g_{ab}

$$\gamma_{ab} = g_{ab} + n_a n_b. \quad (2.25)$$

Using the spacetime metric γ_{ab} we can define two projections operators:

$$\gamma^a_b = g^a_b + n^a n_b \quad (2.26)$$

$$N^a_b = \delta^a_b - \gamma^a_b \quad (2.27)$$

$$= -n^a n_b \quad (2.28)$$

γ^a_b projects spacetime tensors on the surface Σ and N^a_b projects spacetime tensors onto the normal to surface. These two allow us to decompose any tensor into spatial and timelike parts. We can check this by how these operators affect the unit normal n_a ,

$$\gamma^a_b n_a = g^a_b n_a + n^a n_b n_a \quad (2.29)$$

$$= n_b - n_b \quad (2.30)$$

$$= 0 \quad (2.31)$$

$$N^a_b n_a = -n^a n_b n_a \quad (2.32)$$

$$= n_b. \quad (2.33)$$

Once we have access to these two projection operators, we can construct the equations that will allow us to evolve the spatial metric γ_{ab} forward in time. But note that n^a is not a natural time direction because it is not dual to Ω_a , $n^a \Omega_a = \frac{1}{\alpha}$. But, for any spatial vector β^a , t^a defined as

$$t^a = \alpha n^a + \beta^a \quad (2.34)$$

is dual to Ω_a ,

$$t^a \Omega_a = \alpha n^a \Omega_a + \beta^a \Omega_a = 1. \quad (2.35)$$

t^a maps how a spatial coordinate label moves in the surfaces Σ as we evolve forward in time, as shown in Fig.(2.1).

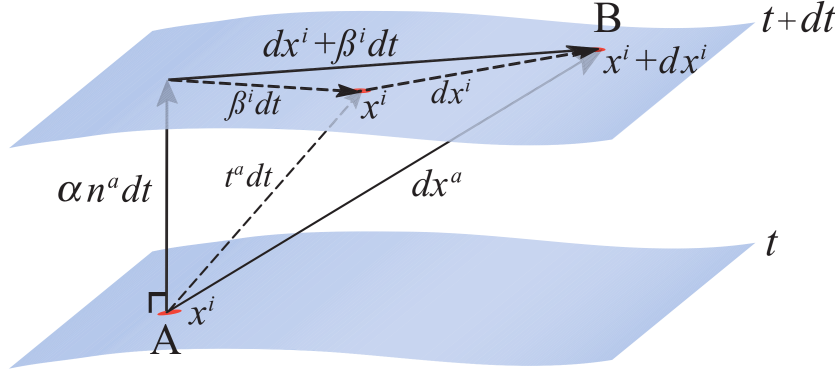


Figure 2.1: Figure 2.4 from [2]. Two adjacent spatial surfaces Σ labeled by t and $t + dt$. Note that the vector $t^a dt$ maps the spatial label x^i to the same spatial label on the next surface. The lapse αdt tells us how much proper time has passed between the surfaces along the normal vector n^a . Any infinitesimal displacement vector dx^a can be decomposed as $dx^a = t^a dt + dx^i = \alpha n^a dt + (dx^i + \beta^i dt)$.

For numerical simulations, we will eventually need to make some coordinate choices. We start with,

$$t^a = (1, 0, 0, 0) \quad (2.36)$$

which implies that the Lie derivatives along t^a become simple partial time derivatives, simplifying our equations a lot. We also choose spatial coordinates such that for purely spatial vectors (like β^a) the 0th component is zero, $\beta^a = (0, \beta^i)$.

This choice also means that,

$$\Omega_a t^a = 1 \implies \Omega_a = (1, 0, 0, 0) \quad (2.37)$$

$$\omega_a = \alpha \Omega_a \implies \omega_a = (\alpha, 0, 0, 0) \quad (2.38)$$

$$t^a = \alpha n^a + \beta^a \implies n^a = (\alpha^{-1}, -\alpha^{-1} \beta^i) \quad (2.39)$$

$$n^a n_a = -1 \implies n_a = (-\alpha, 0, 0, 0) \quad (2.40)$$

With these, we can decompose the full spacetime metric,

$$g^{ab} = \gamma^{ab} - n^a n^b = \begin{pmatrix} -\alpha^{-2} & \alpha^{-2} \beta^i \\ \alpha^{-2} \beta^j & \gamma^{ij} - \alpha^{-2} \beta^i \beta^j \end{pmatrix} \quad (2.41)$$

$$g_{ab} = \begin{pmatrix} -\alpha^2 + \beta_l \beta^l & \beta_i \\ \beta_j & \gamma_{ij} \end{pmatrix} \quad (2.42)$$

We will briefly cover the basics of the ADM formalism below, as it defines many quantities used in NR simulations. But it should be noted that they are not stable to solve numerically, because in the form described below, they are not strongly hyperbolic. After writing down the ADM equations, we will describe the Generalized Harmonic formalism, which is what SpEC uses to evolve Einstein's equations numerically.

To get the ADM equations, first we need to define the 3-dimensional covariant derivative by using the projection operators,

$$D_a f \equiv \gamma_a^b \nabla_b f, \quad (2.43)$$

$$D_a T_c^b \equiv \gamma_a^d \gamma_e^b \gamma_c^f \nabla_d T_f^e. \quad (2.44)$$

Now, we define the extrinsic curvature K_{ab} as,

$$K_{ab} = -\frac{1}{2} L_n \gamma_{ab} \quad (2.45)$$

Note that we cannot choose γ_{ab} and K_{ab} independently of each other; they should be compatible with the full spacetime metric g_{ab} . To get the constraints that they should satisfy, we start with the Gauss and Codazzi equations, which relate the 4-dimensional Riemann tensor ${}^{(4)}R_{pqrs}$ to the equivalent quantities on the spatial surface:

$$R_{abcd} + K_{ac} K_{bd} - K_{ad} K_{cb} = \gamma_a^p \gamma_b^q \gamma_c^r \gamma_d^s {}^{(4)}R_{pqrs} \quad (2.46)$$

$$D_b K_{ac} - D_a K_{bc} = \gamma_a^p \gamma_b^q \gamma_c^r n^{(4)} R_{pqrs} \quad (2.47)$$

With these two and Einstein's equation in vacuum, we can derive the Hamiltonian constraint,

$$R + K^2 - K_{ab} K^{ab} = 0 \quad (2.48)$$

the momentum constraints,

$$D_b K^b_a - D_a K = 0. \quad (2.49)$$

and the evolution equation for the extrinsic curvature,

$$\mathcal{L}_t K_{ab} = -D_a D_b \alpha + \alpha (R_{ab} - 2K_{ac} K_b^c + K K_{ab}) + \mathcal{L}_\beta K_{ab} \quad (2.50)$$

From the definition of the extrinsic curvature, we get the evolution equation for the 3-metric γ_{ab} ,

$$\mathcal{L}_t \gamma_{ab} = -2\alpha K_{ab} + \mathcal{L}_\beta \gamma_{ab} \quad (2.51)$$

Finally, the choice of our basis simplifies these equations further because all the time components of the spatial tensors are just zero and $\mathcal{L}_t = \partial_t$. At the end, we have the AMD evolution and constraints equations in vacuum:

$$\begin{aligned} \partial_t K_{ij} &= \alpha \left(R_{ij} - 2K_{ik} K_j^k + K K_{ij} \right) - D_i D_j \alpha \\ &\quad + \beta^k \partial_k K_{ij} + K_{ik} \partial_j \beta^k + K_{kj} \partial_i \beta^k \end{aligned} \quad (2.52)$$

$$\partial_t \gamma_{ij} = -2\alpha K_{ij} + D_i \beta_j + D_j \beta_i \quad (2.53)$$

$$R + K^2 - K_{ij} K^{ij} = 0 \quad (2.54)$$

$$D_j (K^{ij} - \gamma^{ij} K) = 0, \quad (2.55)$$

Generalized Harmonic formalism

The starting point for the generalized harmonic formalism [3–6] is Einstein's equation in vacuum,

$${}^{(4)}R_{ab} = 0 \quad (2.56)$$

By expanding the Ricci tensor in terms of the spacetime metric g_{ab} and the Christoffel symbols, we get,

$$\begin{aligned} & \frac{1}{2} g^{cd} \partial_d \partial_c g_{ab} - g_{c(a} \partial_{b)} {}^{(4)}\Gamma^c - {}^{(4)}\Gamma^c {}^{(4)}\Gamma_{(ab)c} \\ & - 2g^{ed} {}^{(4)}\Gamma_{e(a} {}^{(4)}\Gamma_{b)cd} - g^{cd} {}^{(4)}\Gamma_{ad} {}^{(4)}\Gamma_{ecb} = 0. \end{aligned} \quad (2.57)$$

Now we fix the coordinate gauge by specifying a gauge source function H^a and requiring that ${}^{(4)}\Gamma^a = -H^a$. In theory, this gauge source function can be chosen freely, but to get a symmetric hyperbolic system, it is important that H^a does not depend on the metric derivatives. In terms of the lapse and shift defined in the ADM formalism, we have that,

$$\begin{aligned} \partial_t \alpha - \beta^k \partial_k \alpha &= -\alpha (H_t - \beta^i H_i + \alpha K), \\ \partial_t \beta^i - \beta^k \partial_k \beta^i &= \alpha \gamma^{ij} \left[\alpha (H_j + \gamma^{kl} \Gamma_{jkl}) - \partial_j \alpha \right]. \end{aligned} \quad (2.58)$$

Thus, by fixing the gauge source function, we define how the α and β^i evolve in time. Using the gauge source function in Eq.(2.57) we get,

$$g^{cd} \partial_d \partial_c g_{ab} + 2\partial_{(a} g^{cd} \partial_{c} g_{b)d} + 2H_{(a,b)} - 2H_d {}^{(4)}\Gamma_{ab}^d + 2{}^{(4)}\Gamma_{bd}^c {}^{(4)}\Gamma_{ac}^d = 0. \quad (2.59)$$

Now we need to cast these equations into the first-order form. To do this, we introduce two new variables,

$$\begin{aligned} \Phi_{iab} &= \partial_i g_{ab} \\ \Pi_{ab} &= -n^c \partial_c g_{ab} = -\frac{1}{\alpha} \partial_t g_{ab} + \frac{\beta^i}{\alpha} \partial_i g_{ab}. \end{aligned} \quad (2.60)$$

Using these new variables we can write the Eq.(2.59) in the first order form,

$$\partial_t g_{ab} - N^k \partial_k g_{ab} \simeq 0, \quad (2.61)$$

$$\partial_t \Pi_{ab} - N^k \partial_k \Pi_{ab} + N g^{ki} \partial_k \Phi_{iab} \simeq 0, \quad (2.62)$$

$$\partial_t \Phi_{iab} - N^k \partial_k \Phi_{iab} + N \partial_i \Pi_{ab} \simeq 0, \quad (2.63)$$

Here, we have only written the derivative terms of the first-order formalism. The full equations can be found in [6]. Note that we have introduced the gauge source function and new variables to get the first-order form of the generalized harmonic system. This introduces new constraints that must be satisfied for the whole system to be equivalent to Einstein's equations. These constraints are:

$$C_a = H^a + {}^{(4)}\Gamma^a \quad (2.64)$$

$$C_{iab} = \partial_t g_{ab} - \Phi_{iab} \quad (2.65)$$

While the system in Eq.(2.63) is symmetric hyperbolic, it will still blow up if we try to solve it because small constraint violations will exponentially grow in time. To fix this, we have to introduce constraint damping by adding multiples of the constraint C_{iab} to the equations.

$$\partial_t g_{ab} - (1 + \gamma_1) N^k \partial_k g_{ab} \simeq 0, \quad (2.66)$$

$$\partial_t \Pi_{ab} - N^k \partial_k \Pi_{ab} + N g^{ki} \partial_k \Phi_{iab} - \gamma_3 N^k \partial_k g_{ab} \simeq 0, \quad (2.67)$$

$$\partial_t \Phi_{iab} - N^k \partial_k \Phi_{iab} + N \partial_i \Pi_{ab} - \gamma_2 N \partial_i g_{ab} \simeq 0. \quad (2.68)$$

Again, we are only showing the terms with derivatives of the metric. To keep the system symmetric hyperbolic for all choices of γ_1 and γ_2 , we need that $\gamma_3 = \gamma_2 \gamma_1$. For $\gamma_2 > 0$, the constraint C_{iab} is exponentially suppressed during the evolution. Also, $\gamma_1 = -1$ makes the system linearly degenerate, preventing shock formation as long as we have smooth initial data. With these choices, we have a system that can evolve stably in time. The final missing ingredient that we need is the initial data, which we cover briefly in the next section.

2.5 Initial data

The initial data problem in NR is all about approximating the spatial metric and its first time derivative that satisfy the Hamiltonian Eq.(2.48) and momentum constraints Eq.(2.49). There are no good analytical expressions for the metric of two black holes orbiting each other a few orbits away from the merger. This means we rely on solving things numerically and iterating until we reach the desired parameters and accuracy.

We use the XCTS (Extended conformal thin sandwich) [7, 8] formalism to generate initial data for SpEC binary black hole simulations. We will again focus only on the vacuum case. The goal is to decompose γ_{ij} and K_{ij} into smaller and smaller parts until we end up with a set of quantities that can be easily specified and a set of equations that can be easily solved. First, we will define conformally transformed equivalents of the spatial tensors.

$$\gamma_{ij} = \psi^4 \bar{\gamma}_{ij}. \quad (2.69)$$

$\bar{\gamma}_{ij}$ is called the conformal metric, and Ψ is the conformal factor which defines the overall scaling of the metric. The power of 4 for the conformal factor is chosen to

simplify the equations. We can now derive other conformal quantities:

$$\gamma^{ij} = \Psi^{-4} \bar{\gamma}^{ij} \quad (2.70)$$

$$\Gamma_{jk}^i = \bar{\Gamma}_{jk}^i + 2 \left(\delta^i_j \bar{D}_k \ln \psi + \delta^i_k \bar{D}_j \ln \psi - \bar{\gamma}_{jk} \bar{\gamma}^{il} \bar{D}_l \ln \psi \right) \quad (2.71)$$

$$\begin{aligned} R_{ij} = & \bar{R}_{ij} - 2 \left(\bar{D}_i \bar{D}_j \ln \psi + \bar{\gamma}_{ij} \bar{\gamma}^{lm} \bar{D}_l \bar{D}_m \ln \psi \right) \\ & + 4 \left((\bar{D}_i \ln \psi) (\bar{D}_j \ln \psi) - \bar{\gamma}_{ij} \bar{\gamma}^{lm} (\bar{D}_l \ln \psi) (\bar{D}_m \ln \psi) \right) \end{aligned} \quad (2.72)$$

$$R = \psi^{-4} \bar{R} - 8 \psi^{-5} \bar{D}^2 \psi \quad (2.73)$$

By using the Hamiltonian constraint Eq.(2.48) and the equation for the Ricci scalar, we get that,

$$8 \bar{D}^2 \psi - \psi \bar{R} - \psi^5 K^2 + \psi^5 K_{ij} K^{ij} = 0. \quad (2.74)$$

We also decompose K_{ij} into its trace K and traceless part A_{ij} , before defining their conformal equivalents.

$$K_{ij} = A_{ij} + \frac{1}{3} \gamma_{ij} K, \quad (2.75)$$

$$\begin{aligned} A^{ij} &= \psi^{-10} \bar{A}^{ij} \\ K &= \bar{K}. \end{aligned} \quad (2.76)$$

Again, the powers of the conformal factors were chosen to simplify the final equations. With these choices, the Hamiltonian and momentum constraints become,

$$8 \bar{D}^2 \psi - \psi \bar{R} - \frac{2}{3} \psi^5 K^2 + \psi^{-7} \bar{A}_{ij} \bar{A}^{ij} = 0 \quad (2.77)$$

and

$$\bar{D}_j \bar{A}^{ij} - \frac{2}{3} \psi^6 \bar{\gamma}^{ij} \bar{D}_j K = 0 \quad (2.78)$$

respectively.

Now we further decompose the conformal traceless tensor $\bar{A}^{ij}$ as,

$$\bar{A}^{ij} = \bar{A}_{TT}^{ij} + \bar{A}_L^{ij}, \quad (2.79)$$

where $\bar{A}_{TT}^{ij}$ is the transverse-traceless part which has zero divergence,

$$\bar{D}_j \bar{A}_{TT}^{ij} = 0. \quad (2.80)$$

The longitudinal part $\bar{A}_L^{ij}$ can be written as a symmetric, traceless gradient of a vector potential W^i ,

$$\bar{A}_L^{ij} = \bar{D}^i W^j + \bar{D}^j W^i - \frac{2}{3} \bar{\gamma}^{ij} \bar{D}_k W^k \equiv (\bar{L}W)^{ij}. \quad (2.81)$$

This decomposition allows us to write divergence of A^{ij} in terms of W^i as,

$$\bar{D}_j A^{ij} = \bar{D}_j A_L^{ij} = \bar{D}_j (\bar{L}W)^{ij} = \bar{D}^2 W^i + \frac{1}{3} \bar{D}^i (\bar{D}_j W^j) + \bar{R}^i_j W^j \equiv (\bar{\Delta}_L W)^i \quad (2.82)$$

Now that we are done with the conformal part of the XCTS, we turn to the thin sandwich part. The idea is that if we want to generate initial data for quasiequilibrium situations, like two binary black holes orbiting each other, then it is more natural to provide γ_{ij} and its time derivative instead of γ_{ij} and K_{ij} . To do this we start by defining u_{ij} as,

$$u_{ij} \equiv \gamma^{1/3} \partial_t \left(\gamma^{-1/3} \gamma_{ij} \right), \quad (2.83)$$

which by using the ADM evolution equation for γ_{ij} becomes,

$$u^{ij} = -2\alpha A^{ij} + (L\beta)^{ij}. \quad (2.84)$$

We also uniquely define its conformal equivalent using these two equations,

$$\bar{u}_{ij} \equiv \partial_t \bar{\gamma}_{ij} \quad (2.85)$$

$$\bar{\gamma}^{ij} \bar{u}_{ij} \equiv 0. \quad (2.86)$$

With these we can show that $u_{ij} = \Psi^4 \bar{u}_{ij}$. And $\bar{A}^{ij}$ can be written in terms of $\bar{u}_{ij}$ as,

$$\bar{A}^{ij} = \frac{\psi^6}{2\alpha} ((\bar{L}\beta)^{ij} - \bar{u}^{ij}). \quad (2.87)$$

We can simplify the equations by introducing the “densitized” lapse $\alpha = \Psi^6 \bar{\alpha}$,

$$\bar{A}^{ij} = \frac{1}{2\bar{\alpha}} ((\bar{L}\beta)^{ij} - \bar{u}^{ij}). \quad (2.88)$$

All this allows us to get an equation for the shift using the momentum constraint,

$$(\bar{\Delta}_L \beta)^i - (\bar{L} \beta)^{ij} \bar{D}_j \ln(\bar{\alpha}) = \bar{\alpha} \bar{D}_j \left(\bar{\alpha}^{-1} \bar{u}^{ij} \right) + \frac{4}{3} \bar{\alpha} \psi^6 \bar{D}^i K \quad (2.89)$$

Finally, by using the Hamiltonian constraints and the evolution equation for K_{ij} , we can get,

$$\begin{aligned} \bar{D}^2(\alpha\psi) = & \alpha\psi \left(\frac{7}{8} \psi^{-8} \bar{A}_{ij} \bar{A}^{ij} + \frac{5}{12} \psi^4 K^2 + \frac{1}{8} \bar{R} + 2\pi\psi^4(\rho + 2S) \right) \\ & - \psi^5 \partial_t K + \psi^5 \beta^i \bar{D}_i K. \end{aligned} \quad (2.90)$$

Having all the ingredients in place, we can now construct the initial data by following these steps:

- Specify $\bar{\gamma}_{ij}$, K and their time derivatives $\bar{u}_{ij}$, $\partial_t K$.
- Solve Eq.(2.89) to get β^i .
- Solve Eq.(2.77) to get the conformal factor.
- Solve Eq.(2.90) to get the α .
- Construct $\bar{A}^{ij}$ using Eq.(2.88).
- Construct $\gamma_{ij} = \Psi^4 \bar{\gamma}_{ij}$
- Construct $K_{ij} = \psi^{-2} \bar{A}_{ij} + \frac{1}{3} \gamma_{ij} K$.

The final parameters, like the masses and spins, of the binary system depend on the initial choices of $\bar{\gamma}_{ij}$, K , $\bar{u}_{ij}$, and $\partial_t K$. If we want to simulate a specific BBH system, we need to solve the initial-data system multiple times in an optimization procedure that converges to the desired parameters [9, 10].

References

- [1] Charles W. Misner et al. *Gravitation*. 2018.
- [2] Thomas W. Baumgarte and Stuart L. Shapiro. *Numerical Relativity*. 2010.
- [3] Helmut Friedrich. “On the hyperbolicity of Einstein’s and other gauge field equations”. In: *Communications in Mathematical Physics* 100.4 (Dec. 1985), pp. 525–543. DOI: 10.1007/BF01217728.

- [4] David Garfinkle. “Harmonic coordinate method for simulating generic singularities”. In: *Phys. Rev. D* 65.4, 044029 (Feb. 2002), p. 044029. DOI: 10.1103/PhysRevD.65.044029. arXiv: gr-qc/0110013 [gr-qc].
- [5] Frans Pretorius. “Numerical relativity using a generalized harmonic decomposition”. In: *Classical and Quantum Gravity* 22.2 (Jan. 2005), pp. 425–451. DOI: 10.1088/0264-9381/22/2/014. arXiv: gr-qc/0407110 [gr-qc].
- [6] Lee Lindblom et al. “A new generalized harmonic evolution system”. In: *Classical and Quantum Gravity* 23.16 (Aug. 2006), S447–S462. DOI: 10.1088/0264-9381/23/16/S09. arXiv: gr-qc/0512093 [gr-qc].
- [7] James W. York Jr. “Conformal “Thin-Sandwich” Data for the Initial-Value Problem of General Relativity”. In: *Phys. Rev. Lett.* 82.7 (Feb. 1999), pp. 1350–1353. DOI: 10.1103/PhysRevLett.82.1350. arXiv: gr-qc/9810051 [gr-qc].
- [8] Harald P. Pfeiffer and James W. York. “Extrinsic curvature and the Einstein constraints”. In: *Phys. Rev. D* 67.4, 044022 (Feb. 2003), p. 044022. DOI: 10.1103/PhysRevD.67.044022. arXiv: gr-qc/0207095 [gr-qc].
- [9] Luisa T. Buchman et al. “Simulations of unequal-mass black hole binaries with spectral methods”. In: *Phys. Rev. D* 86.8, 084033 (Oct. 2012), p. 084033. DOI: 10.1103/PhysRevD.86.084033. arXiv: 1206.3015 [gr-qc].
- [10] Serguei Ossokine et al. “Improvements to the construction of binary black hole initial data”. In: *Classical and Quantum Gravity* 32.24, 245010 (Dec. 2015), p. 245010. DOI: 10.1088/0264-9381/32/24/245010. arXiv: 1506.01689 [gr-qc].

Chapter 3

SPECTRAL METHODS AND FINITE PRECISION ERRORS

In Chapter 2, we have already emphasized why we need to solve Einstein's equations numerically. The first hurdle one faces when trying to solve anything on a computer is the transition from the continuous world of mathematical equations to the discrete world of transistors.

To solve a differential equation on a computer, we first have to discretize it. The first step in discretization is to choose a discrete representation or approximation for our functions. This can be done in several ways. The simplest approach is to represent a function by storing its values at fixed intervals. For example, if we have a function $f(x)$ defined on $[-1, 1]$, we could store its values at intervals of 0.1. These discrete values then become our approximation to the function.

Another approach is to approximate our functions with functions we understand better. For example, we can approximate a function as a sum of polynomials, a sum of trigonometric functions, and so on. The idea is that we understand how to take derivatives and integrals and how to perform various operations on polynomials, which makes everything easier. To represent functions this way, we write something like

$$f(x) \approx \sum_{n=1}^N a_n \phi_n(x), \quad (3.1)$$

where $\phi_n(x)$ is the basis function that we choose.

In the discussion that follows, we will mostly use Chebyshev polynomials as our placeholder basis functions. It turns out that Chebyshev polynomials belong to a class of polynomial functions that are particularly powerful for approximating smooth functions, as we will see. In SpEC, we work with 3D geometries, and the basis functions are tensor products of more fundamental ones. We will not discuss 3D geometries or tensor products of basis functions here, but the conclusions that we draw from the Chebyshev example will still hold. In the next section, we will talk about spectral methods, focusing in particular on the coefficients a_n . We will see how their rate of decay can tell us a lot about the function and the quality of our approximation. In the second section, we will talk about floating-point errors,

which are the fundamental source of errors in arithmetic operations coming from the hardware that we use.

3.1 Spectral Methods

To better understand spectral methods, let us start with a concrete example. Suppose we want to represent the function

$$f(x) = \sin(x) + \cos(x) \quad (3.2)$$

using Chebyshev polynomials on the interval $x \in [-1, 1]$. The first step is to get the values of the function at special points called collocation points. For Chebyshev polynomials, there are several choices of collocation points, each with its own advantages and disadvantages. The choice that we will use is the zeros of the Chebyshev polynomial. The N zeros of the N th Chebyshev polynomial ($\phi_N(x)$) are,

$$x_i = \cos\left(\frac{(2i+1)\pi}{2N}\right), \quad i = 0, \dots, N-1. \quad (3.3)$$

So, if we want to represent the function using N Chebyshev polynomials, we will need the value of $f(x)$ at these N zeros. Let us represent them as $f_i = f(x_i)$. The coefficients a_n in the expansion are called the spectral values of the function, and the values f_i are called the collocation values of the function. For our choice of collocation points, there are very efficient algorithms based on fast Fourier transforms (FFTs)¹. The situation is very similar to Fourier transforms, where we can move between the frequency and physical representations of a function.

The big advantage of spectral methods is that the truncation error in the approximation,

$$E_N(x) = f(x) - \sum_{n=1}^N a_n \phi_n(x) \approx a_{N+1} \approx e^{-(N+1)}, \quad (3.4)$$

falls off exponentially with N if $f(x)$ is a smooth function. In practice, this means that we need very few spectral coefficients to approximate the function very accurately.

For the function $f(x) = \sin(x) + \cos(x)$, Fig.(3.1) shows that we need only about 18 coefficients to reach an error of 10^{-20} . In practice, we rarely need our functions to be resolved to this extent. Most of our simulations target errors of order 10^{-10} to 10^{-13} in each domain at best.

¹For small values N , there are matrix-based algorithms that are faster. In SpEC, we choose between these two based on timing information, which allows us to go back and forth between these two representations.

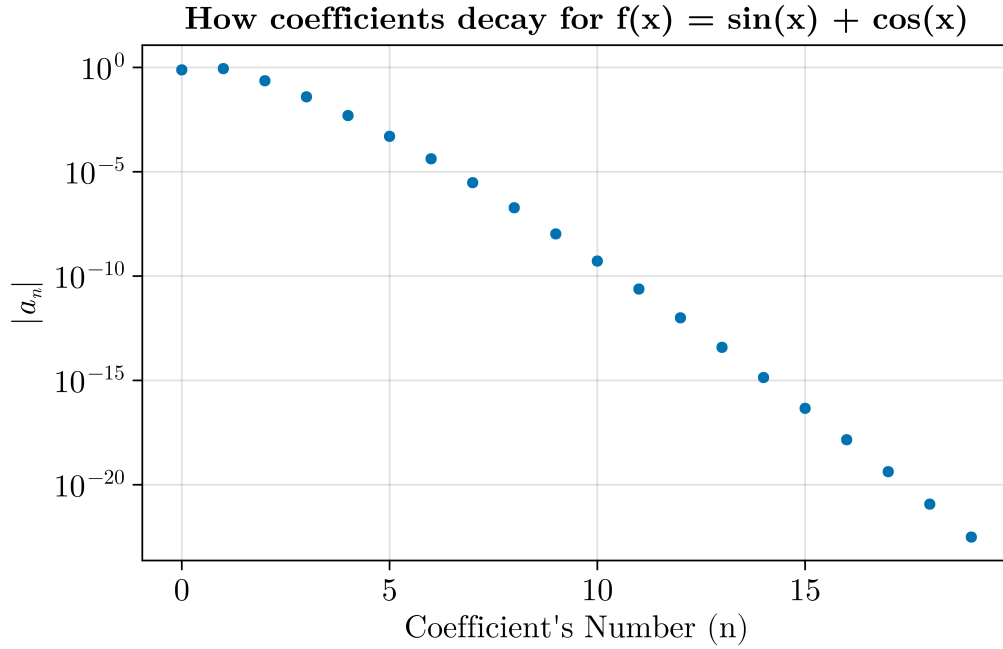


Figure 3.1: This figure shows how the absolute value of the coefficients ($|a_n|$) decays in the Chebyshev approximation of $f(x) = \sin(x) + \cos(x)$. We can see that, for large values of n , the decay is exponential as expected.

From the Chebyshev approximation errors in Fig.(3.2), we can see that the actual error over $x \in [-1, 1]$ is roughly equal to the next coefficient, a_{N+1} . In other words, the truncation error defined in Eq.(3.4) is usually close to the size of the first coefficient that we have dropped. This is another useful property of spectral methods: the truncation-error estimate and the actual error are often very close.

Even for a smooth function, the rate at which the coefficients a_n decay depends on the rate of variation of the function. For the example in Fig.(3.3), we can see that to reach the same error of 10^{-20} , we need close to 60 coefficients.

Now that we have decided how to express the function in terms of polynomials, the next thing we need to figure out is how to take its derivative. For Chebyshev polynomials, there are analytic formulas for the matrix that map the collocation values of a function at the Chebyshev nodes to the collocation values of its derivative at the same points. The actual differentiation matrix depends on the choice of the

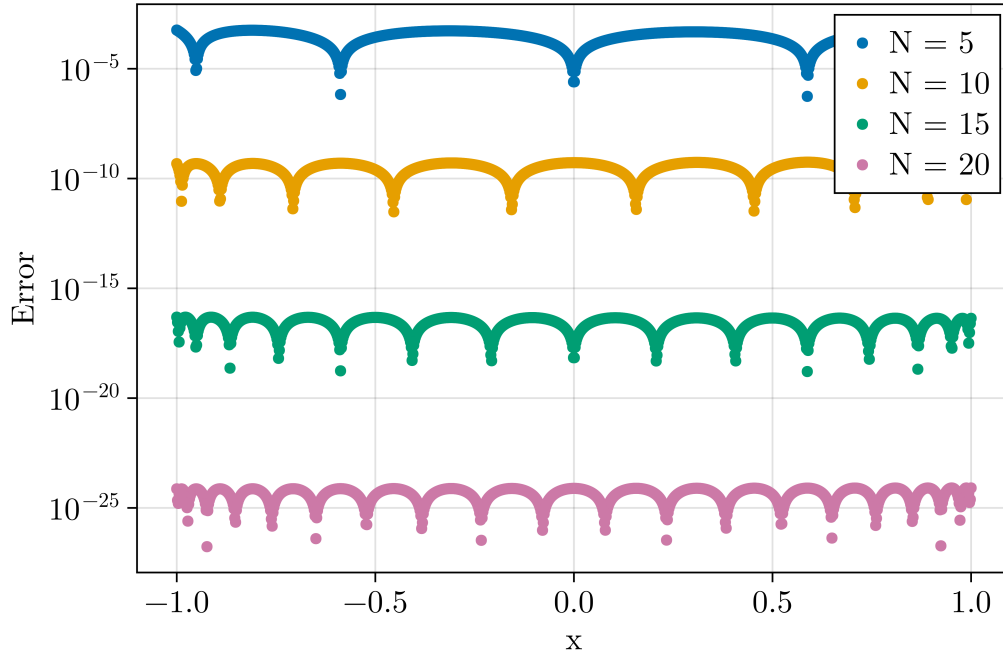


Figure 3.2: This figure shows how the error in our Chebyshev approximation of $f(x) = \sin(x) + \cos(x)$ decreases, as we increase the number of polynomials used. Note that the error decreases fairly uniformly across the interval in x . This mostly uniform distribution of errors is one of the main advantages of using Chebyshev polynomials.

collocation points. For our choice, Eq.(3.3), we can write

$$f'_i \equiv \frac{df}{dx}(x_i) \approx \sum_{j=0}^{N-1} D_{ij} f_j, \quad (3.5)$$

where the Chebyshev differentiation matrix is

$$D_{ij} = \begin{cases} \frac{(-1)^{i+j} \sin \theta_j}{\sin \theta_i (x_i - x_j)}, & i \neq j, \\ \frac{x_i}{2(1 - x_i^2)}, & i = j. \end{cases} \quad (3.6)$$

One thing that we have brushed under the rug until now is floating-point errors. If we look at Fig.(3.1), we see that the coefficients start at order unity and go as low as order 10^{-20} . If we try to do the same calculation using standard floating type Double (float64), the spectral coefficients will stop decaying once they hit $\approx 2 * 10^{-16}$. This phenomenon can be seen in Fig.(3.4), and we will discuss why it happens in the next section.

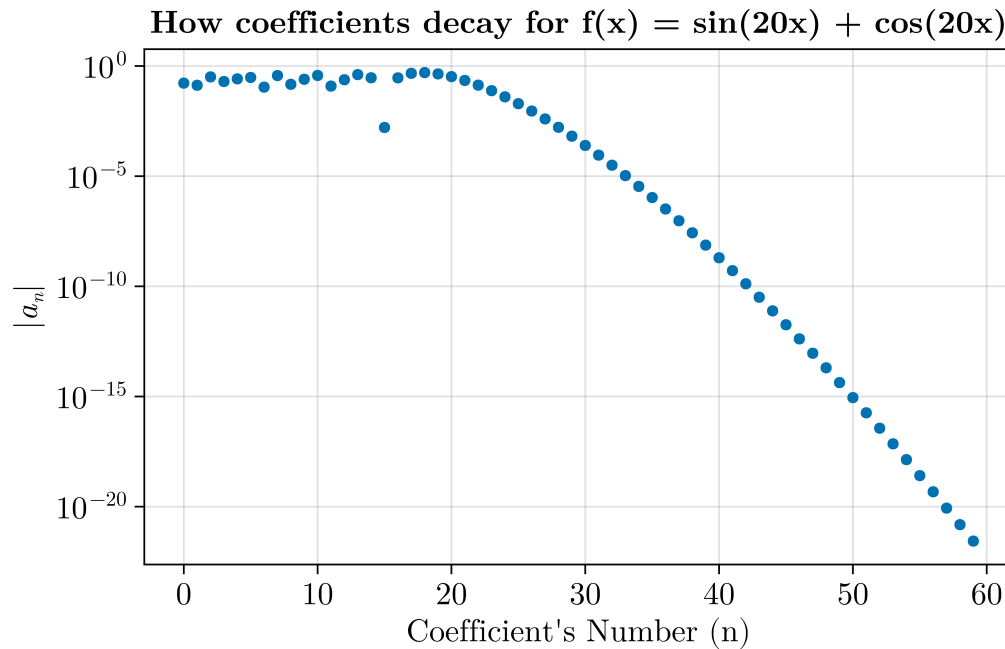


Figure 3.3: This figure shows how the absolute value of the coefficients ($|a_n|$) decays in the Chebyshev approximation of $f(x) = \sin(20x) + \cos(20x)$. Note that this time the exponential decay of the coefficients starts much later compared to Fig.(3.1), because we have a more variable function. In particular, we can see that the exponential decay starts approximately after $N \approx 20$, which matches our intuition: the asymptotic exponential decay starts once we have resolved all the dominant frequencies.

3.2 Floating-point errors

The fundamental issue is that we have to represent real numbers using a fixed number of bits on a computer, and most real numbers have an infinite number of digits. The standard data type used to represent real numbers is double precision, which has 64 bits. Out of these 64 bits, we have:

- 1 sign bit, which tells us the sign of the number,
- 11 exponent bits, which store the scale of the number,
- 52 significand bits, which store the significant digits of the number.

In practice, this means that we can represent real numbers between 10^{-308} and 10^{308} with around 15 decimal digits of accuracy. The fact that we can only store the first

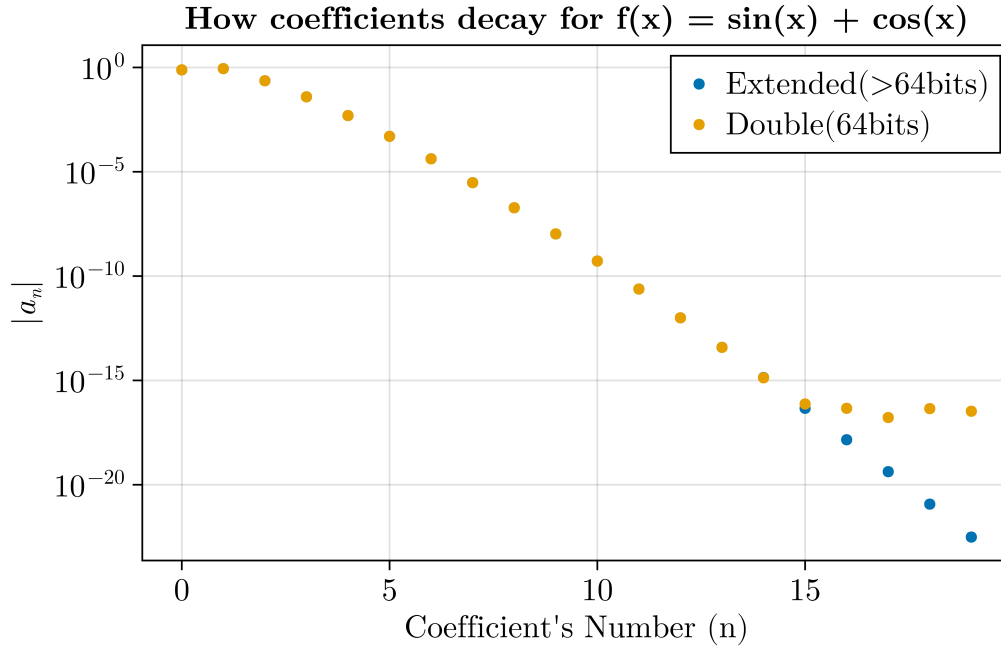


Figure 3.4: This figure shows how the absolute value of the coefficients ($|a_n|$) decays in the Chebyshev approximation of $f(x) = \sin(x) + \cos(x)$. This time we are using two different precisions, orange is for the standard double precision with 64 bits, while the blue dots are for extended precision with more bits. We can see that the spectral coefficients stop decaying after $n = 15$ because of the floating-point errors.

15 or so digits of a real number means that almost all real numbers are truncated when stored in double precision. This truncation of real numbers is the source of floating-point errors. These floating-point errors mean that whenever we perform an arithmetic operation, we introduce a relative error of approximately 2×10^{-16} . This is why, in Fig.(3.4), the orange dots representing the spectral coefficients in double precision saturate at around 2×10^{-16} .

There are three interconnected reasons why we usually try to deal with floating-point errors instead of simply using higher precision:

- Double-precision arithmetic is embedded directly in the silicon of modern processors, which means that these operations are very fast. In contrast, using higher-precision types can easily be multiple orders of magnitude slower.
- Most simulations do not require a level of accuracy where a per-operation relative error of order 10^{-16} matters.

- Most numerical libraries assume that we are working with double precision and do not work with higher-precision types.

The effects of floating-point errors on arithmetic operations can be easily seen in the following Python code snippet:

```

1  from math import pi
2
3  1.0 + 1e-20
4  # 1.0
5
6  1.0 + 1e-10 - 1.0
7  # 1.0000000082740371e-10
8
9  (1.0 + pi/1e6 - 1.0)*1e6 - pi
10 # -1.262989712813578e-11

```

In the last computation, we get an error of approximately 10^{-11} because we add a small number, $\pi \times 10^{-6}$, to an order-unity number and then subtract the order-unity part back out. This kind of loss of precision is very common in Numerical Relativity simulations, where some components of the metric can contain order-unity background terms alongside much smaller corrections. We will study its implications in much more detail in later chapters.

It is worth reiterating that floating-point errors need not be small in absolute terms. A good proxy for the size of these errors is the spacing between adjacent representable floating-point numbers. This spacing can be seen using the *nextafter* function from the *numpy* library:

```

1  import numpy as np
2
3  np.nextafter(1.0, np.inf) - 1.0
4  # 2.220446049250313e-16
5
6  np.nextafter(1e10, np.inf) - 1e10
7  # 1.9073486328125e-06
8
9  np.nextafter(1e100, np.inf) - 1e100
10 # 1.942668892225729e+84

```

Note that the absolute spacing, and therefore the typical absolute scale of floating-point errors, becomes very large for large numbers, even though the relative spacing remains close to 10^{-16} .

Chapter 4

FINDING BLACK HOLE SPINS EFFICIENTLY DURING A NUMERICAL BINARY EVOLUTION

4.1 Abstract

The dynamics of a binary black hole system depend on its masses and spins. For such a system, it is not possible to define these quantities in an unambiguous way; however, we have several reasonable definitions that reduce to the expected values in the limit of infinite separation. Approximate Killing Vector (AKV) spin is one of the spin definitions we use in our numerical relativity code SpEC. As the name suggests, these are constructed using the approximate Killing vectors of the apparent horizons. Finding these approximate Killing vectors requires solving a generalized eigenvalue problem, which scales as $O(L^6)$, where L is the highest spherical harmonic mode used to represent the apparent horizon. This scaling means that the cost of computing AKV spins becomes very high as we simulate systems at higher resolutions, especially systems with high spins or mass ratios. In this paper, we describe a new algorithm for computing AKV spins that is much faster than the current algorithm.

4.2 Introduction

Until now, the accuracy of Numerical Relativity (NR) simulations has been good enough for the gravitational wave detections [1–5] made by LIGO-Virgo-KAGRA (LVK) [6, 7] collaboration. However, with the new generation of gravitational-wave detectors coming online [8–10], we will need to redo many of our simulations at much higher resolutions. Often, running a simulation at a higher resolution is just a matter of using more computing resources, but other issues may show up that were not present before. In the case of the SpEC code, one of the issues that started showing up when trying to simulate more complicated systems was the increasing wall-time cost of spin computations. While SpEC, like other numerical relativity codes, is designed to run on multiple cores, the spin computations are still performed on a single core for algorithmic reasons. This means that when the spin computation is slow, all the other cores are effectively wasting resources waiting for it to complete. This is already an issue with SpEC, which runs on $O(100)$ cores, but it will become a bigger issue with next-generation codes like SpECTRE [11] running on many

thousands of cores.

The outline of this paper is as follows. Section 4.3 has some mathematical background about how the AKV spins are defined. Then, in Section 4.4, we briefly describe the existing algorithm and why it slows down rapidly as the systems being evolved get complicated. In Section 4.5, we describe the new algorithm, which is much faster. Finally, we discuss possible future improvements in Section 4.6 and present conclusions in Section 4.7.

4.3 AKV spin definition

In this section, we only define the quantities relevant to understanding the algorithm. The reader should refer to Appendix A of [12] and Ref. [13] for more background and references.

There is no unambiguous way to define the spin of a dynamical black hole horizon in a BBH system. We usually settle for a quasilocal spin definition that reduces to the expected values in the limit of infinite separation. AKV spin (χ_{AKV}) is one of these quasilocal spin definitions, which is constructed by using approximate Killing vectors of the black hole horizons.

We start by setting up the conventions used throughout the paper. We work in the standard 3+1 decomposition [14–16], with spatial slice denoted by Σ . The spatial metric, extrinsic curvature, and the two-dimensional apparent horizon are denoted by g_{ij} , K_{ij} , and $\mathcal{H}$, respectively. Lower-case Latin letters index the spatial tensors, and upper-case Latin letters index the tensors on the two-dimensional horizon surface. We also put a ring on top of the geometric quantities related to $\mathcal{H}$ to separate them from the geometric quantities on the spatial slice Σ .

In NR, the standard definition [17] of spin angular momentum is given by

$$S = \frac{1}{8\pi} \oint_{\mathcal{H}} \phi^i s^j K_{ij} dA \quad (4.1)$$

where s^j is the outgoing normal of $\mathcal{H}$. Here $\vec{\phi}$ is an “azimuthal” vector field, tangent to $\mathcal{H}$, which carries the information about the spin axis.

It can be shown that Eq.(4.1) is conserved [17] when $\vec{\phi}$ is a Killing vector of the dynamical horizon world tube. For an isolated Kerr black hole, this Killing vector defines the spin axis of the horizon, but there is no reason why such a Killing vector field should exist for the horizon of a black hole in a binary system. The next best thing one can use is the vector field that is closest to being a Killing vector field—in

other words, an approximate Killing vector. We will now describe the method used in SpEC for constructing these approximate Killing vector fields. We start from Killing's equation for a vector field ϕ_B

$$D_{(A}\phi_{B)} = 0 \quad (4.2)$$

This condition can be split into two conditions: first, that the field has no expansion

$$\Theta := \dot{g}^{AB} D_A \phi_B = 0 \quad (4.3)$$

and second, that it has no shear

$$\sigma_{AB} := D_{(A}\phi_{B)} - \frac{1}{2}\dot{g}_{AB}\Theta = 0 \quad (4.4)$$

where $\dot{g}_{AB}$ is the metric on the apparent horizon and D_A is the covariant derivative associated with that metric. If $\vec{\phi}$ is not a Killing vector field, then we cannot satisfy both of these conditions at once, so we will choose a $\vec{\phi}$ that is expansion-free and has the smallest possible shear. Any expansion-free vector field tangent to the apparent horizon can be written as

$$\phi^A = \epsilon^{AB} D_B z, \quad (4.5)$$

where z is some smooth function and ϵ^{AB} is the Levi-Civita tensor. The problem of finding the approximate Killing vector then reduces to finding the z that gives the smallest shear, or equivalently, minimizes the norm $|\sigma|^2$:

$$\|\sigma\|^2 := \oint_{\mathcal{H}} \sigma_{BC} \sigma^{BC} dA. \quad (4.6)$$

Using Eqs.(4.4) and (4.5) and integrating by parts, we get

$$\|\sigma\|^2 = \oint_{\mathcal{H}} z H z dA. \quad (4.7)$$

Here H is

$$Hz = D^4 z + \dot{R} D^2 z + D^4 \dot{R} D_A z \quad (4.8)$$

where D^2 is the Laplacian on the apparent horizon and $\dot{R}$ is the Ricci scalar. If we try to minimize Eq.(4.7) with respect to z , we obtain the trivial solution that z is constant and $\vec{\phi}$ is zero. To avoid this, we will require that the minimized $\vec{\phi}$ has a positive norm,

$$\oint_{\mathcal{H}} \phi^A \phi_A dA \quad (4.9)$$

Using a Lagrange multiplier λ and the fact that $\phi^A \phi_A = D^A z D_A z$, we get

$$I[z] := \oint_{\mathcal{H}} z H z dA + \lambda \left(\oint_{\mathcal{H}} D^A z D_A z dA - N \right), \quad (4.10)$$

where N is a still-undetermined positive constant. Fixing the value of N is equivalent to normalizing the approximate Killing vector field, which determines not only the spin axis but also the magnitude of the spin.

Minimizing the functional (4.10) with respect to z gives a generalized eigenvalue problem:

$$H z = \lambda D^2 z \quad (4.11)$$

We can try to discretize Eq.(4.11) and solve it numerically. The only remaining issue is that both H and D^2 still share a kernel where z is a constant, which makes this eigenvalue problem singular. The standard way to deal with such issues is to remove the common kernel space and then solve for z . In our case, this is trivial because we are discretizing using spherical harmonics and can just remove the Y_{00} mode.

From Eq. (4.7) and Eq. (4.11) we can see that the amount of shear is proportional to the magnitude of λ . Therefore, once we have discretized Eq. (4.11), the next step is to find the three eigenvalues with the smallest magnitude and their corresponding eigenvectors. We can use these eigenvectors z to compute the spin axis. However, we still need to determine the spin magnitude by normalizing the eigenvectors, which is equivalent to fixing the value of N in Eq. (4.10). To fix this normalization, we use

$$\oint_{\mathcal{H}} (z - \langle\langle z \rangle\rangle)^2 dA = \frac{A^3}{48\pi^2} \quad (4.12)$$

where $\langle\langle z \rangle\rangle$ is the average of z over the surface and A is the surface area of the apparent horizon. We use this equation to normalize the three eigenvectors, after which their Euclidean norms give the spin magnitude. Mathematically, Eq. (4.12) holds only for a Kerr metric, where z is a true rotation generator of the horizon. Therefore, when using it for normalization, we are assuming that the black holes being evolved are approximately Kerr.

Using the AKV z and the normalization condition Eq.(4.12) is the most robust way we know to get the spin magnitude, and it is what we use by default in SpEC.

Determining the spin axis is much more complicated, and not all definitions are based on z . Various definitions of spin axis along with their tradeoffs are discussed in [13].

Now that we have some understanding of why we need to solve a generalized eigenvalue problem, we next describe the current algorithm and explain why it becomes problematic for complex systems.

4.4 Current AKV algorithm

Call the matrices formed by discretizing the operators H and D^2 in Eq.(4.11) M and B , respectively. Now we have a generalized eigenvalue problem $Mx = \lambda Bx$, and the obvious way to get the eigenvectors is to use a standard library like LAPACK. To understand why this becomes an issue, we need to understand how big these matrices are and how they grow with an increase in the complexity of the apparent horizons.

Apparent horizons in SpEC are represented using spherical harmonics. When the black holes are far away from each other, their apparent horizons are mostly spherical and can be approximated using a small number of spherical harmonics. On the other hand, when the black holes are about to merge, the horizons become very distorted, and we need a larger number of spherical harmonics to approximate them. We will denote the highest spherical harmonic used in representing the apparent horizon as L . For the simulations that we have done so far, L can be as small as 15 during the inspiral, but in some complex cases it can get as large as 90 near the merger. As we push to evolve more and more complex systems at higher and higher resolution, we expect the largest value of L we need to increase even further.

The number of rows in M and B matrices is $N = (L - 1)^2 - 1$. The reason for $L - 1$ is that even though we have $L + 1$ modes, we take second derivatives and that pollutes the two highest modes. The term -1 is because we ignore Y_{00} , as described in Sec.(4.3). We only need the eigenvectors corresponding to the three smallest eigenvalues, but using LAPACK's dense solver (`dggev`) means we always compute all eigenvalues (L^2 in number)¹. The theoretical time complexity of Lapack's `dggev` solver is $O(N^3)$, or equivalently $O(L^6)$. This is the reason why the cost of

¹There are variations of LAPACK's `dggev` that can find just the eigenvectors corresponding to the smallest three eigenvalues, but they provide very little speedup as explained at <https://www.netlib.org/lapack/lug/node70.html>. In short, before finding the eigenvectors, LAPACK does a reduction of the eigensystem, and this ends up being the most costly operation. Once the reduction is done finding the eigenvectors itself is comparatively cheap, which means that finding just a few eigenvectors is not significantly faster than finding all of them.

`dggev` is irrelevant for most of the inspiral until we get close to merger, where it starts to increase rapidly. In our simulations, we have found the scaling to be closer to $O(L^7)$, as can be seen from Figs.(4.4) and (4.5). We suspect that the slightly worse scaling may be due to memory pressure, although we have not rigorously verified this.

From Table(4.1), we see that the runtime cost of `dggev` goes from 0.07s to 13100s as L goes from 15 to 100. This poor scaling of `dggev` is particularly problematic for SpEC since, because of its parallelization infrastructure, we have to run `dggev` on a single core. For some simulations, each such spin calculation takes over 30 minutes, while the rest of the cores waste resources waiting for it to finish. This waste of computing resources will only get worse with the need to run more accurate simulations. In the next section, we outline a new algorithm that uses the iterative eigensolver ARPACK to efficiently compute the three required eigenvectors, providing a large speedup.

4.5 Sparse AKV algorithm

The sparse AKV algorithm has three major parts:

- Assembling the matrices M and B
- Applying the shift-invert transform to the eigenvalue problem
- Using cached LU decomposition for the action of the matrix inverse

We will discuss each of these components one by one.

Assembling the matrices

In Sec.(4.4), we started the discussion from the point where we already have the matrices M and B . In practice, we first have to assemble these matrices. SpEC has two functions, $f_M(x)$ and $f_B(x)$, that give the action of the matrices M and B on a vector x . Thus, to generate the matrices, we need to operate on N cartesian basis vectors, which can be costly. From Table(4.1), we see that the cost of constructing the matrix is actually more than the cost of `dggev` for $L < 25$. We expect the time complexity of generating the matrices M and B to be around $O(N^2)$, one N because we need to act on N basis vectors and another N because the cost of functions $f_M(x)$ and $f_B(x)$ is $O(N)$. From Figs. (4.4) and (4.5), we see that the cost lies between $O(L^{4.5}) - O(L^5)$ depending on the system we are running on. This scaling behavior

means that the cost of `dggev` quickly overtakes the cost of constructing the matrices, becoming the main bottleneck.

It is important to consider the cost of the matrix generation because, in theory, we do not even need to construct these matrices when using ARPACK. Like many iterative algorithms, ARPACK can work directly with the functions $f_M(x)$ and $f_B(x)$, saving both memory and computation. Working in this matrix-free manner was our original plan, but it failed for reasons described in detail in Sec. (4.9).

The final version we ended up using involves constructing the full matrices M and B , and we will assume that we have access to those in the later sections.

Shift Invert transform

The number of iterations needed by ARPACK to converge depends on the distribution of the eigenvalues. If we are trying to find the first few largest eigenvalues and they are well separated, then the convergence is very fast. On the other hand, if we are trying to find the smallest eigenvalues, convergence is very slow, and it might not even converge for some systems. For problems like ours, where we need to find the smallest eigenvalues, the way to maintain good convergence is to use the shift-invert transform.

The shift-invert transform is a method for transforming our eigenvalue problem into another in which the desired eigenvalues become large and well separated from their neighbors. The original problem is

$$Mx = \lambda Bx \quad (4.13)$$

After a shift-invert transform:

$$(M - \sigma B)^{-1} Bx = \nu x, \quad (4.14)$$

where

$$\nu = \frac{1}{\lambda - \sigma} \quad (4.15)$$

The first thing to observe is that both problems have the same eigenvectors, while the corresponding eigenvalues are related via Eq.(4.15). From Eq.(4.15), we see that among the new eigenvalues 4.14, ν is largest for the values of λ that are closest to σ . So if we want to find the three smallest eigenvalues, we should just set $\sigma = 0$ and solve the eigensystem (4.14). This will work until one of the eigenvalues gets too close to σ , and then the condition number of the matrix $M - \sigma B$ will diverge.

One could consider shifting the value of σ a little away from zero, but then we will find the three eigenvalues closest to σ and not the three smallest eigenvalues.

This is not an issue if we are finding these eigenvectors in isolation, because we can always adjust the value of σ slightly and then try again. In a long-running simulation where the eigenvalues keep changing, we cannot afford to suddenly have slightly wrong eigenvectors because one of the eigenvalues drifts too close to the value of σ . Fortunately, since we are using the norm described in Eq.(4.9), the matrices M and B are positive semidefinite and negative semidefinite, respectively, which implies that the generalized eigenvalues λ are non-positive. Thus any positive value of σ will always give us the correct results. If we choose a small value for σ , then the eigenvalues we need will be well separated and large, making ARPACK very efficient in finding them. But we do not want σ to be too small because that makes the matrix $M - \sigma B$ ill-conditioned. This means that there is an ideal value of σ that makes the smallest eigenvalues as big as possible without making the matrix $M - \sigma B$ too ill-conditioned. Figs.(4.2) and (4.3) show how the condition number of $(M - \sigma B)$ and the number of iterations taken by ARPACK depend on σ . For these two figures, M and B were generated using the horizon of a Kerr-Schild metric with spin equal to (0,0,0.5) and mass equal to 1. These figures change for different shapes of the apparent horizons, but the same overall trend is always present. For now, we have decided to choose $\sigma = 0.1$, which has worked well for the systems we have simulated.

Note that not all norms lead to eigenvalue problems where all the eigenvalues are negative. For example, the norm used by Cook and Whiting [18] will not have this property, making it hard to choose a stable and effective value for σ .

Action of $(M - \sigma B)^{-1}$

A typical iteration of ARPACK solving a shift-invert system is shown in Fig.(4.1). Note that it requires us to solve a linear system $(M - \sigma B)y = v = Bx$. For $\sigma = 0.1$, ARPACK usually needs around 30 iterations to get the three smallest eigenvalue and their corresponding eigenvectors. Of course, we do not want to take the inverse of the matrix, so the two options are using LU decomposition or iterative methods. For reasons already mentioned in Sec. 4.9, we will stay away from the iterative methods and focus on using LU decomposition for solving the linear system.

Now the theoretical issue with using LU decomposition is that it scales as $O(N^3) = O(L^6)$, which is the same as dggev. This means that scaling-wise, we have just

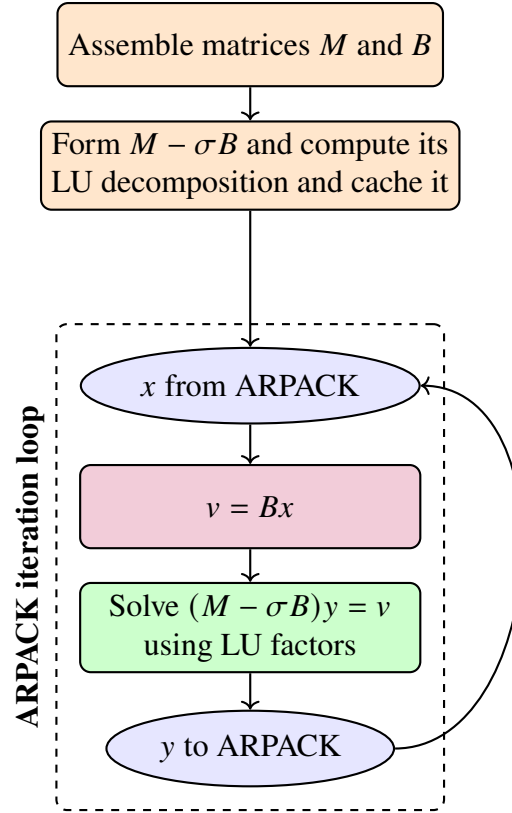


Figure 4.1: Steps in the new AKV algorithm using LU decomposition and ARPACK. We first do the one-time setup, which requires assembling the matrices M and B and computing the LU decomposition of $(M - \sigma B)$. ARPACK needs the action of $(M - \sigma B)^{-1}B$ on the vector x provided by ARPACK. To construct this vector, we first apply B to obtain $v = Bx$, and then solve $(M - \sigma B)y = v$ for y using the LU factors.

pushed the problem from `dggev` to the LU solve. But unlike `dggev`, LU factorization is much more optimized², and runs at least two orders of magnitude faster than `dggev`, as we can see from Table 4.1.

Of course, because of the $O(L^6)$ scaling for very large L , the cost of LU decomposition will eventually overtake the cost of matrix generation. For the systems we are solving, this will happen only when $L \sim O(1000)$, well beyond any conceivable application. And at that point, we would have much bigger issues, like the tiny time step that would be required in the simulations.

Note that while ARPACK requires the actions of $(M - \sigma B)^{-1}$ for each iteration it

²Part of the reason is that LU decomposition makes very good use of BLAS Level 3 routines while `dggev` does not. And in general, the number of flops required to do an LU decomposition is much smaller than that required by `dggev`. Thus, in practice, LU decomposition is much faster, even if both theoretically scale as $O(N^3)$.

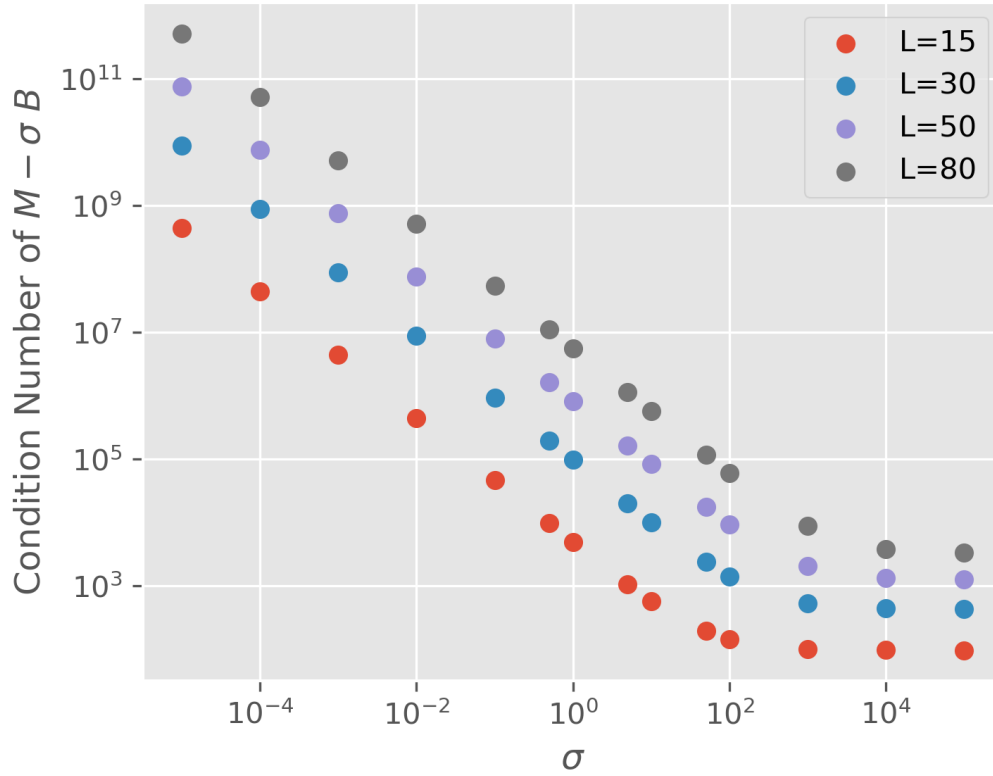


Figure 4.2: Dependence of the condition number of the matrix $M - \sigma B$ on σ for different L . The matrices were generated for a test Kerr-Schild metric with spin equal to (0,0,0.5) and mass equal to 1. Note that the condition number also gets worse as the number of modes L required to represent the horizon increases. The behavior will depend on the shape of the black hole horizons, but for the systems we tested, having a small σ always showed this blowup because most of the time one of the eigenvalues is close to zero.

takes, we only need to do the costly LU factorization once and then cache it. Getting the action of $(M - \sigma B)^{-1}$ on a vector from a cached LU factorization is as cheap as a simple matrix-vector product. From Table 4.1, we can see that the cost of doing the LU decomposition is negligible compared to the cost of constructing the matrices.

4.6 Future improvements

The simplest first possible improvement to try would be to get the matrix-free version of the algorithm to work. The idea would be to generate the full matrices every 100th step or so to use as a preconditioner for the iterations. If the spin does not change very rapidly, then the matrix should also change slowly, and the same preconditioner

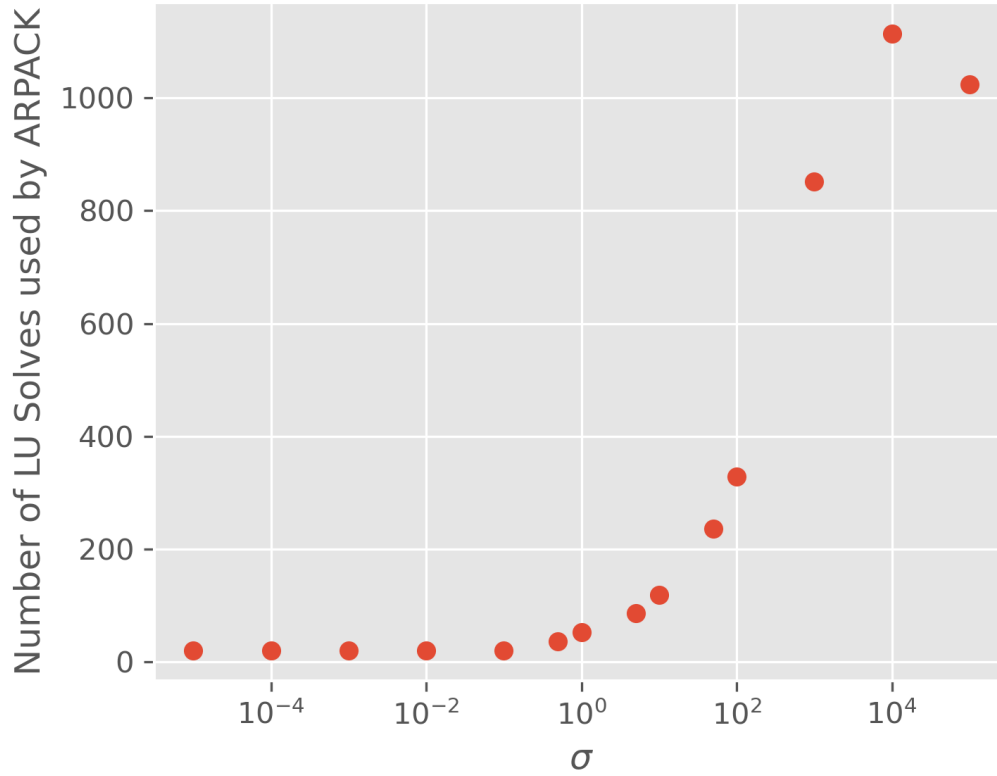


Figure 4.3: Number of LU solves required by ARPACK to converge. Note that ARPACK needs one LU solve per iteration, so this also represents the number of iterations to converge. Although this figure is for the same simple Kerr-Schild spacetime as Fig.(4.2), the number of iterations rarely goes beyond 30 even for much more complicated systems. On the other hand, not having a good enough value of σ makes the convergence very slow, as we can see from the graph. For the BBH simulations, we use $\sigma = 0.1$, which we have found has a good balance between the number of iterations required by ARPACK and the condition number of the matrix $M - \sigma B$.

could be used for multiple time steps. The main issue is that close to the merger spins change rapidly, and there we may end up calculating the full matrices much more frequently.

Another possibility is to try to get a better preconditioner using the analytical properties of matrices, which will allow us to work in a matrix-free manner, avoiding the cost of building the matrices M and B . The construction of the matrices requires L^2 calls to the functions f_M and f_B , which is the bottleneck in the new algorithm. When using a matrix-free version, we need to use an iterative linear

L	Mat gen (s)	LU (s)	Arpack (s)	dggev (s)
15	0.117	0.00244	0.00419	0.0734
20	0.428	0.00158	0.00286	0.301
25	1.12	0.00594	0.00788	1.29
30	2.77	0.015	0.0145	3.62
35	4.76	0.0357	0.0309	9.95
40	10.3	0.0754	0.0646	26.4
50	26.7	0.276	0.183	125
60	71	0.806	0.404	361
70	147	2.01	0.775	949
80	228	4.47	1.35	3.02e+03
90	407	9.01	2.06	5.95e+03
100	965	18	4.7	1.31e+04

Table 4.1: Timing data for matrix generation, LU decomposition, ARPACK iterations, and LAPACK’s dggev on Caltech HPC (Intel(R) Xeon(R) Platinum 8276 CPU @ 2.20GHz). The column showing the ARPACK timing also includes the time taken by the LU solves required for each iteration.

solver like GMRES, which will call these functions multiple times to get the action of $(M - \sigma B)^{-1}$. In our testing, we have found that with $\sigma = 0.1$, ARPACK typically requires fewer than 30 iterations to converge. For each ARPACK iteration, we need the linear solver to converge in less than $O(L)$ iterations (otherwise it will be faster to generate the full matrix), which is very hard to do without a good preconditioner.

Note that the matrix generation is an embarrassingly parallel problem where we apply the functions f_M and f_B on unit vectors one by one to construct the matrices. For large values of L , we can gain a good speedup by doing this in parallel.

4.7 Conclusion

We have shown that the new algorithm based on the sparse eigensystem solver (ARPACK) and LU decomposition is close to two orders of magnitude faster than the original one based on the dense eigensystem solver (LAPACK). If we include the time required to construct the matrices, which is required by both algorithms, then the speedup is closer to one order of magnitude. We tried to use a fully matrix-free algorithm, but were unable to find preconditioners for which the overall cost of the iterations and the preconditioner construction was lower than the cost of constructing the full matrix. However, the idea of using the fixed full matrix for a number of timesteps as a preconditioner is a promising future improvement to explore.

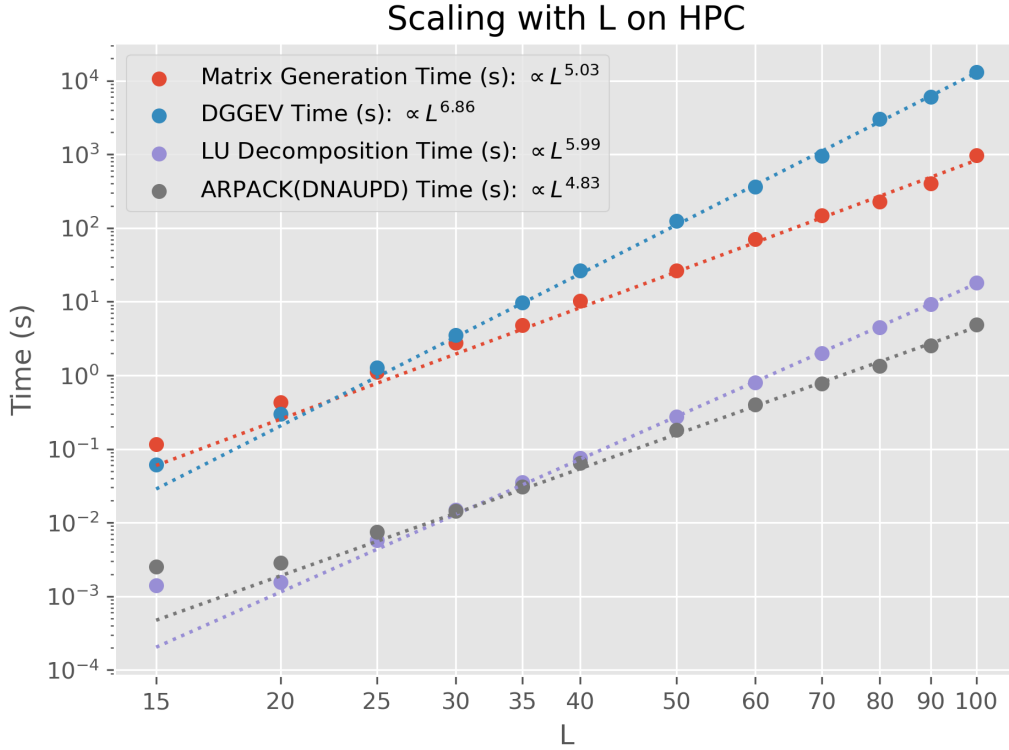


Figure 4.4: Scaling of various parts of the algorithm with the highest spherical harmonic mode L . The runs were performed on a high-performance CPU cluster. The solid dots are the real data, while the dotted lines show the best fits. The fits were weighted such that the measurements at higher L have more weight as they are less noisy. The legends include the scaling from these fits. We see that the scaling of matrix generation and `dggev` is slightly worse than the theoretical expectation of $O(L^4)$ and $O(L^6)$, respectively. Note that the new AKV algorithm using ARPACK + LU decomposition (purple + black lines) is orders of magnitude faster than the current AKV algorithm using `dggev` (blue line). By a large margin, the current bottleneck is matrix generation (red line).

4.8 Timing data PC

In this section, we show the same data as in Table 4.1 and Fig. 4.4, but generated on a PC (Intel(R) Core(TM) i9-10900X CPU @ 3.70GHz). This was done to get an idea about how things scale with differences in the libraries and memory bandwidth, etc. The first thing to note is that matrix generation is around twice as fast now, while LAPACK operations are slower (LU is around 10 times slower and `dggev` is around 2 times slower). The LAPACK operations being slower is expected because HPC is using vendor-optimized libraries and has access to more SIMD lanes. The

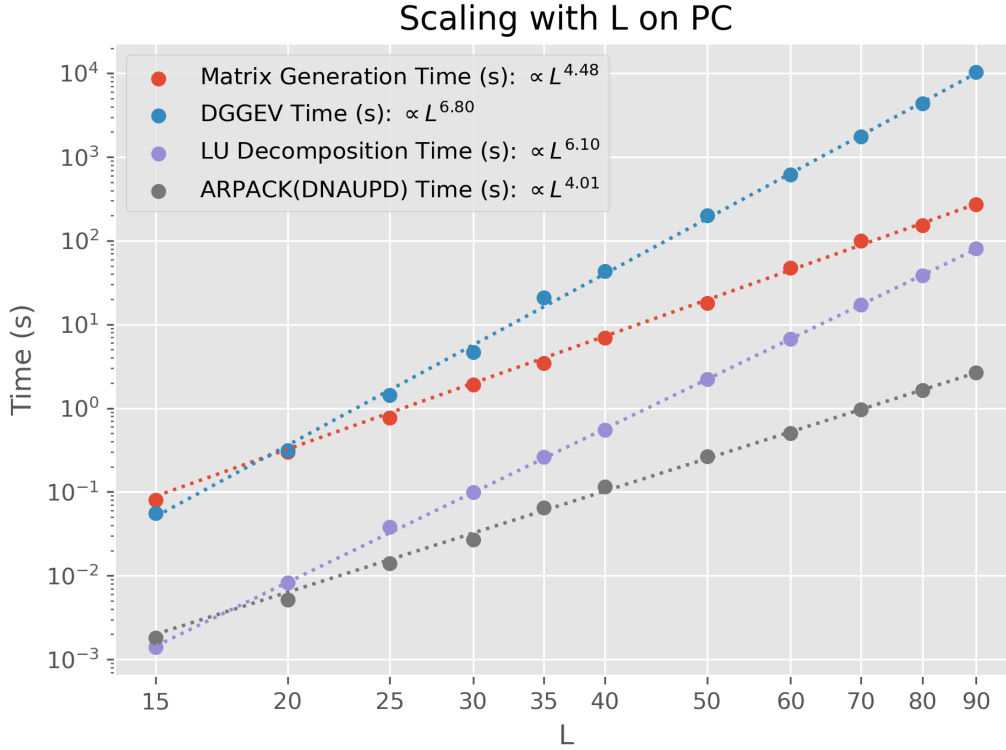


Figure 4.5: Scaling of various parts of the algorithm with the highest spherical harmonic mode L . The runs were performed on a consumer-grade processor (Intel(R) Core(TM) i9-10900X CPU @ 3.70GHz). Labels and conclusions are the same as for Fig.(4.4). Note that the LU decomposition is $\approx 10 \times$ slower while the matrix generation is $\approx 2 \times$ faster when compared to running on an HPC system, as shown in Fig.(4.4).

matrix generation being faster on a local PC is not too surprising either, because for single-core jobs, desktop CPUs are faster than cluster ones and have access to more memory bandwidth per core. Note that the slowdown in LU factorization is much more severe than the slowdown in `dggev`. We suspect this is because the LU factorization, being a much more widely used algorithm, is more aggressively optimized than `dggev`, and losing those optimizations hurts it more.

The primary conclusion remains the same, though. The ARPACK + LU factorization path is much faster than the LAPACK (`dggev`) path, and the main bottleneck now is the matrix generation. What does change is the value of L where LU decomposition becomes the bottleneck. This goes from $O(1000)$ to ≈ 200 . For all practical purposes, $L \approx 200$ is still too big, and step size collapse will be a bigger worry at

that point. For all current HPC systems, we will be in the regime shown in Fig.(4.4), where the matrix generation remains the bottleneck for much larger values of L .

L	Mat gen (s)	LU (s)	Arpack (s)	dggev (s)
15	0.0809	0.00139	0.00177	0.0464
20	0.304	0.00827	0.0053	0.306
25	0.761	0.0319	0.0125	1.44
30	1.93	0.101	0.0275	5.05
35	3.46	0.273	0.0686	21.8
40	6.97	0.545	0.108	45.4
50	18.2	2.18	0.269	200
60	47.5	6.7	0.503	615
70	99.9	17.2	0.954	1.77e+03
80	153	38.4	1.64	4.34e+03
90	274	79	2.59	1.03e+04

Table 4.2: Timing data for matrix generation, ARPACK, LU decomposition, and LAPACK’s dggev on PC (Intel(R) Core(TM) i9-10900X CPU @ 3.70GHz). The column showing the ARPACK timing also includes the time taken by the LU solves required for each iteration.

4.9 Matrix-free version of the AKV algorithm

As already mentioned in section 4.5, the original goal was not to construct the matrices M and B at all and directly use the functions f_M and f_B . This is shown in Fig.(4.6), where we skip the matrix construction and the LU decomposition.

We need to solve the linear problem $(f_M - \sigma f_B)y = v$ to get the vector y required by ARPACK. This can be done using iterative solvers like GMRES or BiCGSTAB. In almost all cases, these iterative algorithms require a good preconditioner to converge at an acceptable rate. This is particularly true in our case because the condition number of the matrix $(M - \sigma B)$ is fairly big for $\sigma = 0.1$, and if we try to make σ larger, then the number of iterations required by ARPACK increases significantly, as shown in Figs. 4.2 and 4.3. Note that we need the number of iterations required by ARPACK times the number of iterations required by the linear solver to stay below L^4 . Otherwise, the whole algorithm will be as costly as constructing the full matrix.

We tried a few simple preconditioners, and the algorithm worked very well most of the time. However, the convergence became very slow in some edge cases when the horizon was very deformed. Since the spin computation is ideally only a small part of a very large simulation, we would rather have a stable but slightly inefficient

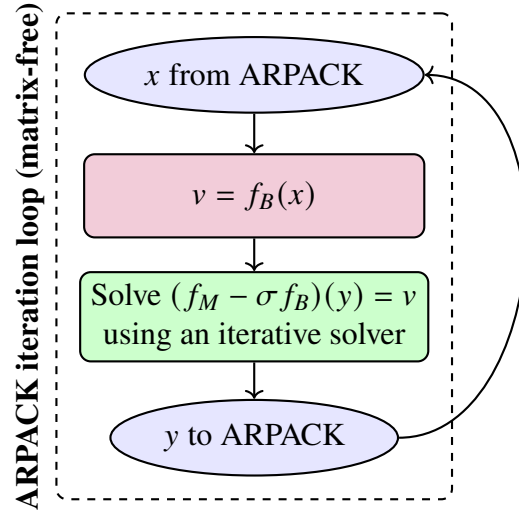


Figure 4.6: Matrix-free workflow for ARPACK in the generalized shift-invert case. Instead of explicitly forming M and B , we use the functions f_M and f_B , which are available in SpEC. At each iteration, ARPACK supplies a vector x , we compute $v = f_B(x)$, and then solve $(f_M - \sigma f_B)(y) = v$ using an iterative solver. The result $y = (M - \sigma B)^{-1} Bx$ is returned to ARPACK for the next iteration.

algorithm over a mostly stable but fast algorithm. And while constructing the matrices is costly, the LU decomposition is a very stable and consistent algorithm even when the condition number gets large. Thus, we decided to swallow the cost of constructing the full matrices M and B and then use LU factorization to efficiently apply the action of $(M - \sigma B)^{-1}$.

To use the matrix-free version in practice requires having a good preconditioner that can be constructed cheaply. When there is no underlying structure and matrices are dense, the standard methods for constructing non-trivial preconditioners from $f_M(x)$ are usually as costly as constructing the full matrix itself.

4.10 Speed difference between finding the smallest and the largest eigenvalues

ARPACK converges much faster when we solve for the largest eigenvalues, and they are well separated. This can be seen in the code below, where we construct an artificial matrix and use ARPACK to find the largest and the smallest eigenvalues.

We generate a diagonal matrix with entries $1, 4, 9, 16, \dots, N^2$.

Listing 4.1: Matrix generation code

```

import scipy.sparse.linalg as ss
import numpy as np

```

```

N = 500
vals = np.arange(1, N + 1, dtype=np.float64)
mat = np.diag(vals**2)

```

Next, we ask ARPACK to find the largest three and the smallest three eigenvalues of this matrix,

Listing 4.2: ARPACK timing comparison

```

# LM: largest three
ss.eigs(mat, k=3, which="LM")
#19.6 ms

# SM: smallest three
ss.eigs(mat, k=3, which="SM")
#1.93 s

# Shift-invert near sigma = 0
ss.eigs(mat, k=3, sigma=0.0, which="LM")
#5.5 ms

```

We see that finding the three smallest eigenvalues is much slower than finding the three largest eigenvalues (19.3ms vs 1930ms). But if we use the shift-invert transform, which essentially transforms the smallest eigenvalues into the largest ones, ARPACK becomes much more efficient. This is just an artificial example to illustrate the importance of the shift-invert transform and the speedups that one can achieve. In our case, for a black hole with spin $(0, 0, 0.5)$, we get a speedup of around 5 times for $L = 15$ and around 10 times for $L = 55$, with the speedup continuing to increase with the matrix size.

References

- [1] A. G. Abac et al. “GWTC-4.0: An Introduction to Version 4.0 of the Gravitational-Wave Transient Catalog”. In: *ApJ* 995.1 (2025), p. L18. DOI: 10.3847/2041-8213/ae0c06. arXiv: 2508.18080 [gr-qc].
- [2] B. P. Abbott, R. Abbott, T. D. Abbott, et al. “GW150914: First results from the search for binary black hole coalescence with Advanced LIGO”. In: *Phys. Rev. D* 93.12, 122003 (June 2016), p. 122003. DOI: 10.1103/PhysRevD.93.122003. arXiv: 1602.03839 [gr-qc].

- [3] B. P. Abbott, R. Abbott, T. D. Abbott, et al. “Observation of Gravitational Waves from a Binary Black Hole Merger”. In: *Phys. Rev. Lett.* 116.6, 061102 (Feb. 2016), p. 061102. doi: 10.1103/PhysRevLett.116.061102. arXiv: 1602.03837 [gr-qc].
- [4] B. P. Abbott, R. Abbott, T. D. Abbott, et al. “GW150914: The Advanced LIGO Detectors in the Era of First Discoveries”. In: *Phys. Rev. Lett.* 116.13, 131103 (Apr. 2016), p. 131103. doi: 10.1103/PhysRevLett.116.131103. arXiv: 1602.03838 [gr-qc].
- [5] B. P. Abbott, R. Abbott, T. D. Abbott, et al. “Properties of the Binary Black Hole Merger GW150914”. In: *Phys. Rev. Lett.* 116.24, 241102 (June 2016), p. 241102. doi: 10.1103/PhysRevLett.116.241102. arXiv: 1602.03840 [gr-qc].
- [6] LIGO Scientific Collaboration, J. Aasi, B. P. Abbott, et al. “Advanced LIGO”. In: *Classical and Quantum Gravity* 32.7, 074001 (Apr. 2015), p. 074001. doi: 10.1088/0264-9381/32/7/074001. arXiv: 1411.4547 [gr-qc].
- [7] F. Acernese, M. Agathos, K. Agatsuma, et al. “Advanced Virgo: a second-generation interferometric gravitational wave detector”. In: *Classical and Quantum Gravity* 32.2, 024001 (Jan. 2015), p. 024001. doi: 10.1088/0264-9381/32/2/024001. arXiv: 1408.3978 [gr-qc].
- [8] Matthew Evans, Alessandra Corsi, Chaitanya Afle, et al. “Cosmic Explorer: A Submission to the NSF MPSAC ngGW Subcommittee”. In: *arXiv e-prints*, arXiv:2306.13745 (June 2023), arXiv:2306.13745. doi: 10.48550/arXiv.2306.13745. arXiv: 2306.13745 [astro-ph.IM].
- [9] Adrian Abac, Raul Abramo, Simone Albanesi, et al. “The Science of the Einstein Telescope”. In: *J. Cosmology Astropart. Phys.* 2026.3, 081 (Mar. 2026), p. 081. doi: 10.1088/1475-7516/2026/03/081. arXiv: 2503.12263 [gr-qc].
- [10] Pau Amaro-Seoane, Heather Audley, Stanislav Babak, et al. “Laser Interferometer Space Antenna”. In: *arXiv e-prints*, arXiv:1702.00786 (Feb. 2017), arXiv:1702.00786. doi: 10.48550/arXiv.1702.00786. arXiv: 1702.00786 [astro-ph.IM].
- [11] Nils Deppe et al. *SpECTRE*. Version 2026.04.01. Apr. 2026. doi: 10.5281/zenodo.19373346. URL: <https://doi.org/10.5281/zenodo.19373346>.
- [12] Geoffrey Lovelace et al. “Binary-black-hole initial data with nearly extremal spins”. In: *Physical Review D* 78 (2008), p. 084017. doi: 10.1103/PhysRevD.78.084017. URL: <https://link.aps.org/doi/10.1103/PhysRevD.78.084017>.
- [13] Robert Owen et al. “Black Hole Spin Axis in Numerical Relativity”. In: *Physical Review D* 99.8 (2019), p. 084031. doi: 10.1103/PhysRevD.99.084031. arXiv: 1708.07325 [gr-qc].

- [14] Thomas W. Baumgarte and Stuart L. Shapiro. *Numerical Relativity*. 2010.
- [15] Masaru Shibata. *Numerical Relativity*. 2016. doi: 10.1142/9692.
- [16] Miguel Alcubierre. *Introduction to 3+1 Numerical Relativity*. 2008.
- [17] Olaf Dreyer et al. “Introduction to isolated horizons in numerical relativity”. In: *Phys. Rev. D* 67.2, 024018 (Jan. 2003), p. 024018. doi: 10.1103/PhysRevD.67.024018. arXiv: gr-qc/0206008 [gr-qc].
- [18] Gregory B. Cook and Bernard F. Whiting. “Approximate Killing vectors on S^2 ”. In: *Phys. Rev. D* 76.4, 041501 (Aug. 2007), p. 041501. doi: 10.1103/PhysRevD.76.041501. arXiv: 0706.0199 [gr-qc].

Chapter 5

HOW ACCURATE IS SPEC? AND WHAT LIMITS ITS ACCURACY?

5.1 Abstract

The improved sensitivity of upcoming gravitational wave detectors will require correspondingly more accurate binary black hole (BBH) simulations. Given the high cost of such simulations, it is essential to understand the sources of error and use computational resources efficiently. We study the sources of error in SpEC's (Spectral Einstein Code) BBH simulations and seek to understand what limits its accuracy. In particular, we focus on the inspiral phase of the Cauchy evolution because it is the longest-running part of the simulation and accounts for the majority of the error in the waveforms. We identify and implement improved adaptive mesh refinement (AMR) and constraint damping strategies, leading to up to two orders of magnitude improvement in finite-radius waveform differences and in the magnitude of constraint violations. We also show that even for the simplest BBH systems, convergence in some parts of the simulation domain is affected by floating-point errors. While the improvements found here are aimed at the SpEC code, the general ideas should apply to any numerical relativity code that uses adaptive refinement in space or time.

5.2 Introduction

The LIGO–Virgo–KAGRA Collaboration [1, 2] has detected over 150 binary black hole (BBH) mergers since the first detection in 2015 [3–11]. These observations have been used to study the evolution of these binary black hole systems and their population distribution in our universe. They also provide some of the most accurate tests of General Relativity [12], by allowing us to probe the highly non-linear regimes of the theory. The next generation gravitational wave detectors, like Laser Interferometer Space Antenna (LISA) [13], Cosmic Explorer (CE) [14], and Einstein Telescope (ET) [15, 16] will further increase the number and quality of these gravitational wave signals.

Analyzing data from these detectors requires accurate theoretical models of BBH mergers, both for searching for events and for determining parameters (e.g. black hole masses and spins) of the system that produced the observed signal. The only

ab initio method of producing such models for inspiral, merger, and ringdown is numerical relativity: the direct solution of Einstein’s equations on the computer.

The high computational cost of numerically evolving binary black hole mergers means that numerical relativity simulations are not used directly for data analysis. Instead, they are used to construct surrogate models [17–21], which interpolate gravitational waveforms to the gaps in the parameter space not covered directly by numerical simulations. They are also used as the ground truth for calibrating other models like Effective-One-Body (EOB) models [22–28].

Given the importance of numerical relativity waveforms, it is important that these simulations are accurate enough for the next generation of detectors. According to some estimates [29, 30], waveform mismatches need to improve by at least an order of magnitude to be accurate enough for LISA. Many new codes that leverage newer hardware and scale better are being developed to simulate these systems at much higher resolutions [31]. In this paper, we will focus on the NR code SpEC, which was used for the simulations in the SXS catalog [32]. A large fraction of the waveforms in the catalog do not meet the LISA requirements, which means we need to rerun those at much higher accuracy. The primary goal of this paper is to address the question: How accurately can we evolve a BBH in SpEC if we had infinite computing resources and time? If the only limiting factor is compute time, then the new generation codes that scale better will allow us to reach LISA accuracies. On the other hand, if there are more fundamental issues in the formalism or the numerical algorithms, then we need to resolve those before we do more high-resolution simulations.

Outline

This paper is divided into four parts. In Section 5.3, we briefly describe the simulation setup and terminology used in the later sections. In Section 5.4, we describe constraint violations and waveform differences, the two main quantities used to quantify the accuracy of SpEC simulations. In Section 5.5, we discuss the changes we made to SpEC and how they improve these accuracy metrics. Finally, in Section 5.6, we discuss factors that limit our accuracy and potential future improvements.

Throughout the paper, we will use Latin alphabet letters $a, b, \dots$ for spacetime indices and $i, j, k, \dots$ for spatial indices.

5.3 Simulation Setup and Spec Background

Generalized Harmonic System

To solve Einstein's equations numerically, most codes recast them into first-order hyperbolic form since good algorithms exist for such formulations. This can be done in multiple ways [33, 34], and the one used in SpEC is called the Generalized Harmonic (GH) formalism [35–38]. The starting point of the GH formalism is the identity

$$R_{ab} = -\frac{1}{2}\psi^{cd}\partial_c\partial_d\psi_{ab} + \nabla_{(a}\Gamma_{b)} + \psi^{cd}\psi^{ef}(\partial_e\psi_{ca}\partial_f\psi_{db} - \Gamma_{ace}\Gamma_{bdf}), \quad (5.1)$$

where ψ_{ab} is the spacetime metric and $\Gamma_a \equiv \psi^{bc}\Gamma_{abc}$ is the trace of the Christoffel symbol. To fix the coordinate gauge freedom, we choose a gauge source function H^a that satisfies,

$$\psi_{ab}\nabla^c\nabla_c x^b = H_a. \quad (5.2)$$

From Eq.(5.2), we can see that by fixing H_a , along with appropriate initial and boundary conditions, we can fix all the coordinate gauge freedoms. Because of the identity

$$\psi_{ab}\nabla^c\nabla_c x^b = -\Gamma_a, \quad (5.3)$$

the gauge source functions H^a is related to the trace of the Christoffel symbol (Γ_a) as $H_a = -\Gamma_a$. Using Eq.(5.2) in Eq.(5.1) gives,

$$R_{ab}^H = -\frac{1}{2}\psi^{cd}\partial_c\partial_d\psi_{ab} + \nabla_{(a}H_{b)} + \psi^{cd}\psi^{ef}(\partial_e\psi_{ca}\partial_f\psi_{db} - \Gamma_{ace}\Gamma_{bdf}) \quad (5.4)$$

To convert these equations into a first-order form, we define two new variables Π_{ab} and Φ_{iab} :

$$\begin{aligned} N\Pi_{ab} &= -\partial_t\psi_{ab} + \gamma_1 N^i\Phi_{iab} \\ \partial_i\psi_{ab} &= \Phi_{iab}. \end{aligned} \quad (5.5)$$

Using these, we can write down the principal part of the GH system,

$$\begin{aligned} \partial_t\psi_{ab} - (1 + \gamma_1)N^k\partial_k\psi_{ab} &\simeq 0 \\ \partial_t\Pi_{ab} - N^k\partial_k\Pi_{ab} + Ng^{ki}\partial_k\Phi_{iab} \\ &\quad - \gamma_1\gamma_2 N^k\partial_k\psi_{ab} \simeq 0 \\ \partial_t\Phi_{iab} - N^k\partial_k\Phi_{iab} + N\partial_i\Pi_{ab} - \gamma_2 N\partial_i\psi_{ab} &\simeq 0. \end{aligned} \quad (5.6)$$

Here γ_1 and γ_2 are free parameters used for constraint damping and for adjusting grid speeds [38]. Constraint damping is discussed further in Section 5.4. Full equations for the GH system are given in Section 5.8. In SpEC, we save Π_{ab} and Φ_{iab} as a single variable called κ_{abc} ,

$$\begin{aligned}\kappa_{ibc} &= \partial_i \psi_{bc} \\ \kappa_{0bc} &= \Pi_{bc}.\end{aligned}\tag{5.7}$$

Together, κ_{abc} and ψ_{ab} form the set of fundamental variables in our first-order GH system. Variables H_a , ψ_{ab} , and κ_{abc} are not independent of each other and should satisfy constraint equations which will be discussed later in Section 5.4.

Evolution domain in SpEC

To solve Einstein's equations, we decompose the 3D volume into a set of simple shapes that fill the volume without overlapping: spherical shells, deformed cylindrical shells, and deformed cylinders. These simple shapes are called subdomains, and Figure 5.1 is a 2D cross-section that shows how these subdomains are joined together. This simulation domain is primarily designed to handle the two excisions that remove the black hole singularities [39–42]. There are spherical shells around each excision surface, called SphereAn and SphereBn, where n is an integer. SphereA0 is the subdomain that contains the apparent horizon of the larger black hole, and it is contained inside SphereA1, and so on. Similarly, in the wave zone, we have a set of spherical subdomains denoted by SphereCn, with SphereC0 being the innermost spherical subdomain. The innermost SphereC, the outermost SphereA, and the outermost SphereB subdomains are joined using cylindrical subdomains oriented along the axis connecting the centers of the holes, and distorted to fill the space in between. The handling of boundary conditions at inter-domain boundaries and at the outer boundary is described in [38, 43–45]. More information about the domain structure used in SpEC can be found in Section 3 of [46] and the appendix of [47]. This paper will primarily focus on the SphereA0, which contains one of the apparent horizons, and SphereC6, one of the subdomains from which gravitational waveform data is extracted.

Pseudo-Spectral Methods and Power Monitors

In each subdomain, we write the metric (and other functions) as a sum of (time-independent) spatial basis functions multiplied by (time-dependent) spectral coefficients. For example, in a spherical shell, the basis functions are Chebyshev polynomials in radius and spherical harmonics in angles. We take spatial derivatives

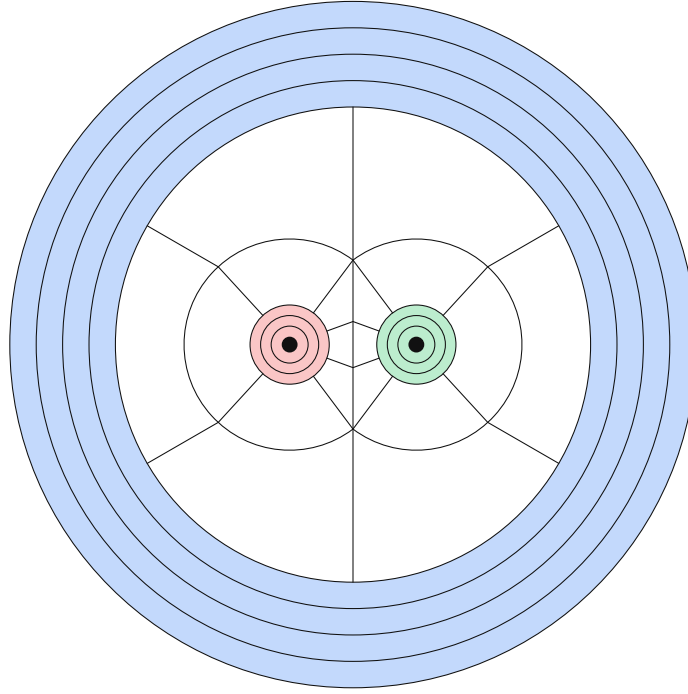


Figure 5.1: The $z = 0$ cross-section of the subdomains used in SpEC for simulating binary black holes. Spherical subdomains in red are the SphereAs, with SphereA0 being the one that contains the horizon of the black hole A. Similarly, green subdomains are SphereBs, and they contain the black hole B. The outermost spherical subdomains in blue are SphereCs, with SphereC0 being the smallest of these. Other subdomains, not colored in the figure, are distorted cylinders and filled cylinders that join the inner SphereA and SphereB subdomains with the outer SphereC subdomains. These subdomains are further divided into smaller subdomains by the AMR algorithm as required. Note that this figure is not to scale, and in general, a typical SphereC subdomain is an order of magnitude larger than a typical SphereA and SphereB subdomain.

by differentiating the basis functions, which we can do exactly; the approximation is that we use only a finite number of basis functions, and the “resolution” of the simulation corresponds to how many basis functions we retain. In this method, the set of time-dependent GH evolution equations can be thought of as a set of (complicated) coupled ODEs in time for the spectral coefficients.

To solve the ODEs, we use the Dormand–Prince 5(4) method [48], which is an ODE solver from the Runge-Kutta family. This is an adaptive time-stepping method that is fifth-order accurate, and an embedded fourth-order estimate is used to compute the local error. This local error is used to change the step size adaptively and keep

the error in our solution below the given ODE tolerance. For the simulations done till now, we have used a fixed ODE tolerance of 10^{-8} . For typical simulations, the error is dominated by spatial truncation error and not temporal truncation error, so for a wide range of ODE tolerances, the adaptive timestepper tends to automatically settle to the Courant limit: taking a timestep slightly larger than the Courant limit induces an instability, momentarily increasing the error rapidly until the timestepper responds by decreasing the timestep again. We will see in section 5.5 that these occasional brief Courant violations cause noise in the solution.

The spectral coefficients (C_{k_1, k_2, k_3}) used in our simulations form a three-index object, with each index associated with one coordinate direction in the three-dimensional domain. For example, on a cylindrical shell, the basis is constructed as a tensor product of Chebyshev polynomial bases in the radial and axial directions and a Fourier basis in the angular direction. Thus, for the coefficients (C_{k_1, k_2, k_3}) on a cylindrical shell, (k_2) labels the Fourier modes, while (k_1) and (k_3) label the Chebyshev modes. We define power monitors $(P_{k_1}, P_{k_2}, P_{k_3})$ using spectral coefficients as

$$P_{k_1} = \sqrt{\frac{1}{N_2 N_3} \sum_{k_2, k_3} |C_{k_1, k_2, k_3}|^2} \quad (5.8)$$

Here, N_1 , N_2 , and N_3 are the number of spectral coefficients in each direction. Power monitors for other indices of C_{k_1, k_2, k_3} , that is, P_{k_2} and P_{k_3} are defined similarly. Note that there are N_1 power monitors for the k_1 direction and that they are functions of time because the spectral coefficients are functions of time.

These power monitors are used to define another important quantity called the truncation error,

$$T[P_k] \equiv \frac{\max(P_1, P_2)}{\prod_{j=1}^{N_k} P_j^{w_j / \sum_{i=1}^{N_k} w_i}} \quad (5.9)$$

$$w_j \equiv \exp\left(-\left(j - N_k + \frac{1}{2}\right)^2\right).$$

The AMR algorithm uses $T[P_k]$ to control the grid resolution, which we discuss next.

AMR tolerance and definition of levels

We need a good AMR algorithm for a BBH simulation because the gravitational field changes rapidly near the black holes and is comparatively smoother in the wave

zone. The AMR algorithm tries to change the grid resolution so that the truncation errors of the two fundamental variables (ψ_{ab} and κ_{abc}) stay below the specified AMR tolerance. To study the convergence, we evolve the same initial data multiple times, but with different AMR tolerances. The base value of the AMR tolerance for a simulation is controlled by a single integer, called its level. We denote this base value, which is also the AMR tolerance in the outer regions, by

$$\epsilon_{\text{AMR}}^{\text{out}} = 2.17 \times 10^{-4} \times 4^{-\text{level}}. \quad (5.10)$$

This means that if the levels of two resolutions differ by 1, then their base AMR tolerance differs by a factor of 4. Once we have the base AMR tolerance, the AMR tolerance near the black holes is set to be a factor of 100 smaller with an exponential falloff to the base value as we move away. Furthermore, the level of a simulation also sets a lower limit on the number of grid points each subdomain is allowed to have. Finally, if the constraints are too large in a subdomain, then the AMR algorithm locally reduces the AMR tolerance of that subdomain.

Most of the simulations used for making the latest SXS catalog [32] had multiple resolutions, typically between levels 1 and 5. A higher level implies a more accurate simulation for the same initial data, but it is harder to make such comparisons when the initial data is different. BBH systems with high spin or high mass ratio usually need a higher level to reach the same accuracy as the simple systems. For the simulations considered in this paper, we only go up to level 6 because we are looking at an equal mass non-spinning BBH system, which is the simplest possible BBH system.

Junk radiation

An unsolved problem in numerical relativity is how to construct BBH initial data that truly corresponds to a snapshot of a BBH that has been inspiraling since the infinite past. Instead, although BBH initial data are physical (i.e., they solve Einstein’s equations), they have different horizon shapes and different (typically zero) gravitational waves compared to a true snapshot. As a result, evolving BBH initial data results in a transient high-frequency pulse of gravitational radiation, commonly referred to as “junk radiation”. Because junk radiation is not part of the solution we are looking for, in BBH simulations, we make no attempt to resolve it. Instead, we wait for it to propagate out of the simulation domain, and we discard the part of the simulation that contains the junk. Because we do not resolve the junk radiation, it produces very large constraint violations. We can easily identify the

junk front in the heatmap in Figure 5.17b as the bright red radiation pulse starting from the center at $t = 0$ and propagating outwards.

Note that even after the junk front has left a subdomain, the constraint violations in that subdomain do not fall to their ‘expected’ value immediately. This can be seen in Figure 5.17b, where the constraints are still high for 2000M after the junk front has passed. We suspect this is caused by reflections of the junk off subdomain boundaries as it propagates through the subdomains. The last of these reflections occurs at the outermost boundary and then propagates inwards. The outer boundary for this simulation was 1000M; therefore, it takes around 2000M after the junk front has passed for the residual junk to be entirely gone from a subdomain. This can be seen clearly in Figure 5.17b where the residual junk leaves SphereC0 around $t=2000M$ but leaves SphereC29, which is at a distance of 1000M from the center, closer to $t=3000M$.

Because of the junk radiation, the BBH parameters (masses, spins, orbital parameters) we get from the initial data solver are different from those we measure after the junk radiation has propagated away. We call the parameters we measure after the junk radiation has passed the ‘relaxed’ parameters. The relaxed parameters depend on the behavior of the junk radiation, and the junk radiation is unresolved, which means that its behavior changes with the resolution. Therefore, the relaxed parameters differ for different levels, even if we start from the same initial data. These slight differences in the relaxed parameters adversely affect the waveform mismatch, merger time, etc. To avoid this issue with junk radiation, we first evolve the system at the highest level we want until the junk front radiation has propagated out of the domain. Then, we change the levels to lower values, allowing the AMR to reduce the resolution. We call this process “perform branching after junk” (PBandJ) [32]. For the simulations considered here, PBandJ was done at $t = 1200M$, which can be seen in Figure 5.17b by the sudden change in the constraint violations around that time.

Binary black hole system parameters

For this study, we evolved an equal-mass, non-spinning binary black hole (BBH) system through the inspiral. All times and distances are quoted in units of the total

mass M of the binary. Inputs to the initial-data solver [49, 50] were

$$\begin{aligned}
 M &= 1, \\
 q &= 1, \\
 d &= 18M, \\
 \Omega_0 &= 0.012226459838/M, \\
 \dot{a}_0 &= -1.72797015626 \times 10^{-5}/M, \\
 \chi_A^{\text{input}} &= (0, 0, 0), \\
 \chi_B^{\text{input}} &= (0, 0, 0).
 \end{aligned}$$

The resulting initial data solution had the following parameters:

$$\begin{aligned}
 M_A + M_B &= 0.9999999426, \quad E_{\text{ADM}} = 0.9934279614, \\
 M_A &= 0.4999999739, \quad M_B = 0.4999999688,
 \end{aligned}$$

and the dimensionless spins are

$$\chi_A = \begin{pmatrix} -9.194 \times 10^{-11} \\ -9.288 \times 10^{-12} \\ -2.138 \times 10^{-8} \end{pmatrix}, \quad \chi_B = \begin{pmatrix} 2.648 \times 10^{-11} \\ 3.037 \times 10^{-11} \\ -2.545 \times 10^{-8} \end{pmatrix}.$$

All simulations in this paper start from this same initial data solution. But the unresolved junk radiation changes the measured masses, spins, and orbital parameters. The relaxed parameters measured after the junk radiation has passed are listed in Section 5.12.

5.4 Accuracy metrics in SpEC

A BBH evolution can be roughly divided into three distinct phases: the inspiral, the merger, and the ringdown. Evolving each of these phases poses unique challenges and requires very different strategies to handle. In terms of coordinate time evolved per CPU hour, the merger and the ringdown parts of the simulation are slower than the inspiral part. But the inspiral part ends up consuming most of our computing resources because it runs the longest. We will focus on the inspiral phase of the simulation because it is the first phase, and most of the error accumulation happens during this phase.

In Section 5.4, we talk about the constraint violations and spectral convergence. In Section 5.4, we discuss how spectral coefficients and power monitors can be used to

check the accuracy of the fundamental variables ψ_{ab} and κ_{abc} . Finally, in Section 5.4 we define waveform difference, the quantity we will use to estimate the errors in our waveforms.

Constraint violations and Constraint damping

When we reformulate Einstein's equations into the first-order Generalized Harmonic (GH) formalism (Section 5.3), they split into time-dependent *evolution equations*, and time-independent *constraint equations*. In practice, we solve the constraint equations at the initial time, and we solve only the evolution equations to advance the solution to subsequent times, ignoring the constraints. This is allowed because of the following property: if the constraint equations are solved exactly at $t = 0$, then solving the evolution equations exactly will guarantee that the constraint equations are satisfied at all subsequent times. In a numerical simulation, we never solve the evolution equations exactly, but only to some finite accuracy. So the constraint equations will not be solved exactly either. We check errors in our solution by monitoring the constraints. The full set of constraints are $\{C_a, \mathcal{F}_a, C_{ia}, C_{iab}, C_{ijab}\}$ in terms of the GH variables:

$$C_a = H_a + \Gamma_a \quad (5.11)$$

$$\mathcal{F}_a \approx t^c \partial_c C_a \quad (5.12)$$

$$C_{ia} \approx \partial_i C_a \quad (5.13)$$

$$C_{iab} = \partial_i \psi_{ab} - \Phi_{iab} \quad (5.14)$$

$$C_{ijab} = 2\partial_{[i} \Phi_{j]ab} = 2\partial_{[j} C_{i]ab}. \quad (5.15)$$

While looking at the individual constraints is informative, it is helpful to define a generalized-harmonic constraint energy density, $\mathcal{E}_{\text{GH}}$ that captures the effect of all these individual constraints:

$$\begin{aligned} \mathcal{E}_{\text{GH}} = & \left[\delta^{ab} C_a C_b + \delta^{ab} \mathcal{F}_a \mathcal{F}_b + g^{ij} \delta^{ab} C_{ia} C_{jb} \right. \\ & \left. + g^{ij} \delta^{ac} \delta^{bd} C_{iab} C_{jcd} + g^{ij} g^{kl} \delta^{ac} \delta^{bd} C_{ikab} C_{jlcd} \right]^{1/2}, \end{aligned} \quad (5.16)$$

where g^{ij} is the spatial metric. Whenever we talk about constraint violations without qualifications, we mean the quantity $\mathcal{E}_{\text{GH}}$.

Some fundamental variables in the GH formalism, like κ_{011} , go from being $O(1)$ near the black holes to being $O(10^{-6})$ at the outer boundary. This makes $\mathcal{E}_{\text{GH}}$ unsuitable for comparing different parts of the simulation domain with each other.

Therefore, we also define a normalization, $\mathcal{N}_{\text{GH}}$, as

$$\begin{aligned} \mathcal{N}_{\text{GH}} = & \left[g^{ij} \delta^{ac} \delta^{bd} \partial_i \psi_{ab} \partial_j \psi_{cd} \right. \\ & + g^{ij} \delta^{ac} \delta^{bd} \partial_i \Pi_{ab} \partial_j \Pi_{cd} \\ & \left. + g^{ij} g^{kl} \delta^{ac} \delta^{bd} \partial_i \Phi_{kab} \partial_j \Phi_{lcd} \right]^{1/2}. \end{aligned} \quad (5.17)$$

Using this normalization, we can define the normalized constraint energy density, $\widehat{\mathcal{E}}_{\text{GH}}$, as

$$\widehat{\mathcal{E}}_{\text{GH}} = \frac{\mathcal{E}_{\text{GH}}}{\mathcal{N}_{\text{GH}}}. \quad (5.18)$$

Because of the finite resolution used in numerical simulations, the constraint violations will always have a small non-zero value. These small constraint violations can grow rapidly as we evolve the system forward in time, causing the simulation to fail. Constraint damping prevents this catastrophic growth by exponentially damping the small constraint violations and is essential for a stable evolution [51]. In the Generalized Harmonic formalism (Section 5.8), the constraint damping happens if we choose a positive value of γ_0 and γ_2 . γ_0 controls the time scale over which the 1-index constraints C_a are exponentially damped and γ_2 does the same for the 3-index constraints C_{iab} .

While one can choose the values of γ_0 and γ_2 separately, in SpEC they are usually chosen to be the same for simplicity. The maximum growth in the constraints happens near the black holes, where the fundamental variables are changing rapidly. Therefore, compared to the outer subdomains, the values of γ_0 and γ_2 near the black holes are chosen to be much larger with a Gaussian falloff. To write this profile, define

$$r_A^2 = |\mathbf{x} - \mathbf{x}_A|^2, \quad (5.19)$$

$$r_B^2 = |\mathbf{x} - \mathbf{x}_B|^2, \quad (5.20)$$

$$r^2 = |\mathbf{x}|^2, \quad (5.21)$$

where $\mathbf{x}_A$ and $\mathbf{x}_B$ are the coordinate centers of the two black holes. For the equal-mass

BBH system considered here, a simplified version of the spatial profile is

$$\begin{aligned}\gamma_0(\mathbf{x}) = \gamma_2(\mathbf{x}) \approx & 8 \exp\left(-\frac{r_A^2}{3.5^2}\right) \\ & + 8 \exp\left(-\frac{r_B^2}{3.5^2}\right) \\ & + 0.075 \exp\left(-\frac{r^2}{(50)^2}\right) \\ & + 0.001.\end{aligned}\tag{5.22}$$

The two near-hole Gaussians in Eq. (5.22) have $O(10)$ amplitude, while the outer-subdomain floor is 0.001. The broader Gaussian has amplitude 0.075 and width $50M$. It extends to $O(100)M$ from the center of the simulation domain. In the actual simulations, all terms except the constant 0.001 are time-dependent and increase as the black holes get closer.

Spectral coefficients

One advantage of using spectral methods is that we have access to the spectral coefficients and power monitors of our fundamental GH variables ψ_{ab} and κ_{abc} . For any smooth function, the value of the successive power monitors should decrease exponentially, that is, $P_k \approx e^{-k}$. One can also calculate the truncation error (Eq.5.9) and check whether the variables are well resolved or not. Truncation errors limit the overall accuracy of our simulations because they represent the inherent error introduced by discretization. While we can get similar information by looking at the constraint violations, it is not as fine-grained as the information that we get from looking at the spectral coefficients and the truncation errors. In particular, constraint violations measure the difference in two variables that should be the same in an ideal system but provide no information about which one of them is wrong. On the other hand, truncation error measures how well a certain variable is resolved in a subdomain, independent of everything else.

Waveform Difference

Another measure of accuracy that we will focus on is the waveform difference as defined in Eq.(5.23). This quantity measures the difference between two waveforms over a chosen time interval and angular sphere. Even when evolving the same initial data, we do not expect to have a zero waveform difference because of various numerical errors, but just like the constraint violations, we expect the difference to

improve as we increase the resolution.

$$\Delta(h_1, h_2) = \frac{1}{2} \frac{\|h_1 - h_2\|^2}{\sqrt{\|h_1\|^2 \|h_2\|^2}} \quad (5.23)$$

$$\|h_1\| = \sqrt{\int_{t_1}^{t_2} \int_{S^2} |h_1|^2 d\Omega dt}$$

For waveform differences, we use finite-radius gravitational waveforms computed with a Regge–Wheeler–Zerilli extraction procedure [52–56]. The gravitational waveforms in the SXS catalog [32], however, are computed by extrapolating a series of these finite-radius Regge–Wheeler–Zerilli waveforms to future null infinity, and then correcting for center-of-mass effects and gravitational-wave memory effects. Because the goal of this paper is to study the errors in the methods of solving the Einstein equations and not the errors in processing and correcting the waveforms, we choose to analyze the simpler finite-radius waveforms here. Keep in mind, however, that for physics applications one should use the waveforms computed by the extrapolation/correction procedure, or even better by Cauchy-Characteristic Extraction (CCE) [57–62].

5.5 Improvements to SpEC

In this section, we describe the changes we made to SpEC and how these changes improved the accuracy metrics discussed in Section 5.4. In Section 5.5, we describe how the adaptive ODE tolerance improved convergence near the black hole horizons. In Section 5.5, we discuss how more constraint damping and grid resolution improved the constraint violations in the outer subdomains. In Section 5.5, we show that finite-radius waveform differences improve by around two orders of magnitude when we add more resolution to the subdomains connecting the two black holes. We also show that the waveform differences are strongly correlated with deviations in the black holes’ trajectories across different resolutions. Finally, in Section 5.5 we discuss the computational cost of these improvements and explain why the cost increase is modest compared to the accuracy gain.

In the following sections, we compare three sets of simulations. We use the following labels:

- “Old” refers to simulations run with the main branch of SpEC.

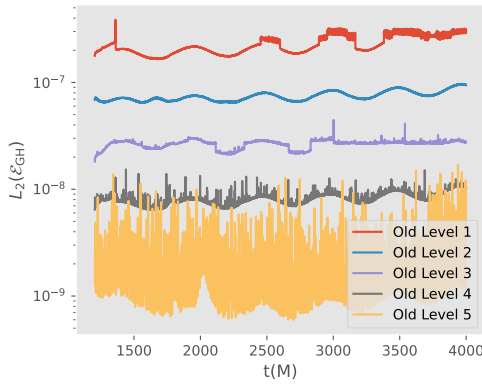
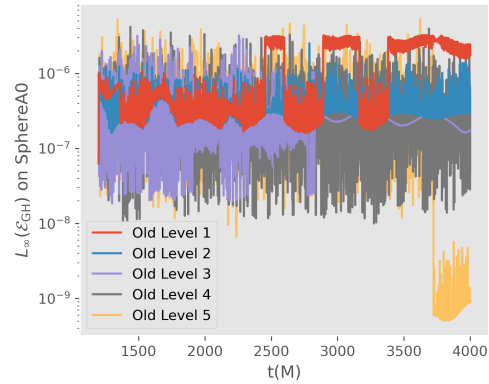
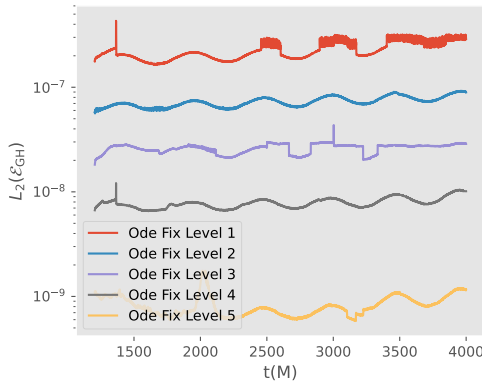
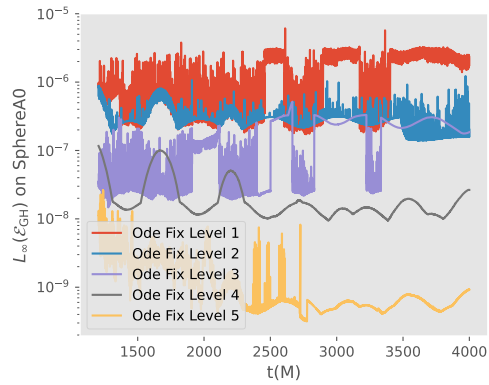
(a) $L_2(\mathcal{E}_{GH})$ vs. time for runs at levels 1-5.(b) $L_\infty(\mathcal{E}_{GH})$ on SphereA0 vs. time for runs at levels 1-5.(c) $L_2(\mathcal{E}_{GH})$ vs. time for runs at levels 1-5 with ode tolerance fix.(d) $L_\infty(\mathcal{E}_{GH})$ vs. time for runs at levels 1-5 with ode tolerance fix.

Figure 5.2: Constraint violations before and after changing the ODE tolerance. The left panels show the L_2 norm over the full simulation domain, while the right panels show the L_∞ norm in SphereA0. The Old simulations show noisy high-resolution behavior near the horizon, while the simulations with the ODE tolerance fix show cleaner convergence.

- “ODE fix” refers to simulations in which the ODE tolerance is changed from the default value of 10^{-8} to $\epsilon_{\text{AMR}}^{\text{out}}/2000$. This makes the ODE tolerance adaptive, allowing it to change with the AMR tolerance.
- “New” refers to simulations that include the ODE tolerance change, increasing the asymptotic value of the constraint damping from 0.001 to 0.01, and setting the AMR tolerance to the uniform value $\epsilon_{\text{AMR}}^{\text{out}}/100$ throughout the domain.

Adaptive ODE Tolerance for Near-Horizon Constraint Convergence

When looking at the constraint violations, we will focus on two regions of the domain defined in Section 5.3: one near the black holes (subdomain SphereA0) and one in the wave extraction region (subdomain SphereC6).

Figure 5.2a shows the L_2 norm of $\mathcal{E}_{\text{GH}}$ over the whole simulation domain for different levels. We can see that the L_2 norm goes down as we increase the resolution. While we are technically converging, there is a lot of noise in the highest-resolution level 5 simulation. To better understand the source of this noise, we need to look at each subdomain separately. We will primarily look at the L_∞ norm of the constraints because the L_2 norm averages over all the points, hiding some potential issues we might have.

For these simulations, the noise is present in the subdomains SphereA0 (Figure 5.2b) and SphereB0 (not shown). The constraint violations in SphereA0 do not converge at all, particularly as we go to higher resolutions like level 4 and level 5. This non-convergence was caused by SpEC not scaling the ODE solver tolerance correctly for high-resolution runs. By using the adaptive ODE tolerance ($= \epsilon_{\text{AMR}}^{\text{out}}/2000$), we can see a clean exponential convergence of the constraint violations in SphereA0 (Figure 5.2d). Furthermore, by comparing Figures 5.2a and 5.2c, we can see that the noise in the L_2 norm of $\mathcal{E}_{\text{GH}}$ over the whole domain is reduced as well, most notably in the high resolution simulations.

The adaptive ODE tolerance formula gives a tolerance smaller than the previous value of 10^{-8} for levels above 2. Since the noise in the constraints improved as we made the ODE tolerance smaller, we believe the noise came from the ODE stepper occasionally taking too-large steps that violated the Courant limit. There is a sweet spot for the ODE tolerance where the Courant noise is eliminated, but the average step size does not decrease. In this regime, we get more accurate simulations without any extra computational cost.

To find the optimal ODE tolerance for these simulations, we decreased the tolerance until the actual step size started to decrease. As one would expect, the optimal ODE tolerance values are roughly proportional to the AMR tolerance values. The heuristic formula that works well for these simulations is ODE tolerance = AMR tolerance/2000. This simple formula will not be optimal for more complex simulations because the AMR algorithm adaptively changes the local AMR tolerance based on the local constraint violations. The Dormand–Prince 5(4) ODE solver computes a local error estimate that is used to adjust the time steps. This local error

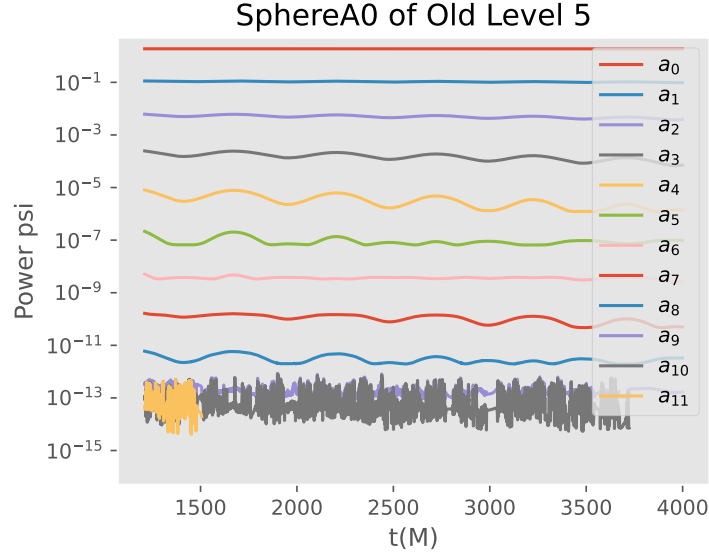


Figure 5.3: Power spectrum of ψ in SphereA0 for a level 5 Old run. The curves a_i are the power monitors of the radial basis functions. The lower modes show exponential convergence, while the highest modes are dominated by noise.

is computed at each grid point and then reduced to a single value using a norm. One possible improvement is to change this norm so that it gives more weight to the inner subdomains, where the noise appears. We have not explored this in detail, so we leave it for future work.

While the improvements in the constraint violations are encouraging, it is very instructive to look at the spectral coefficients before and after the ODE tolerance change. Consider Figure 5.3, which shows the power spectrum of the ψ variable in the SphereA0 subdomain for a level 5 run. We can immediately see that the two highest modes a_{10} and a_{11} are almost all noise, which in part explains the high constraint violations that we see in Figure 5.2b. On comparing this with Figure 5.4, which has the ODE tolerance fix, we can see that the mode a_{10} has comparatively less noise and is dropped by the AMR algorithm earlier around $2700M$. We can also see that the magnitude of the noise/error in the coefficient a_{10} in Figure 5.3 is several orders of magnitude smaller than the constraint violations in Figure 5.2b. This is because constraint violations involve taking derivatives of the fundamental variables, which amplifies the error.

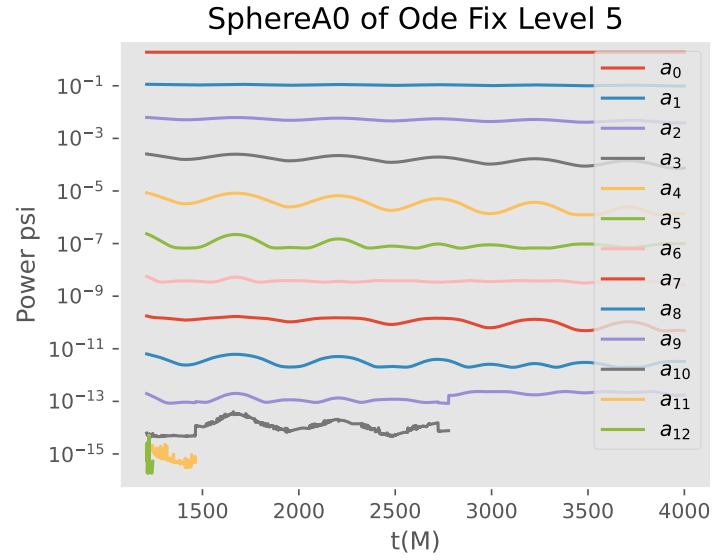


Figure 5.4: Power spectrum of ψ in SphereA0 for a level 5 run with the ODE tolerance fix. Compared to the Old run, the highest retained modes have less noise, and the AMR algorithm drops noisy modes.

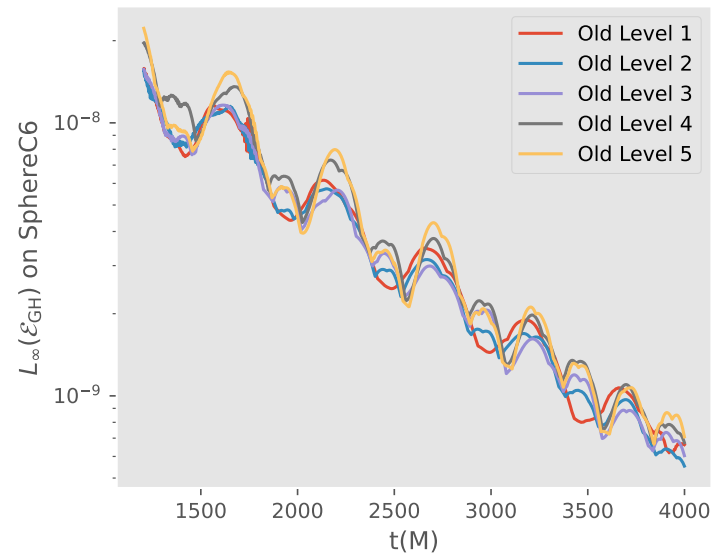


Figure 5.5: $\mathcal{E}_{GH}$ on SphereC6 vs. time for runs at levels 1-5.

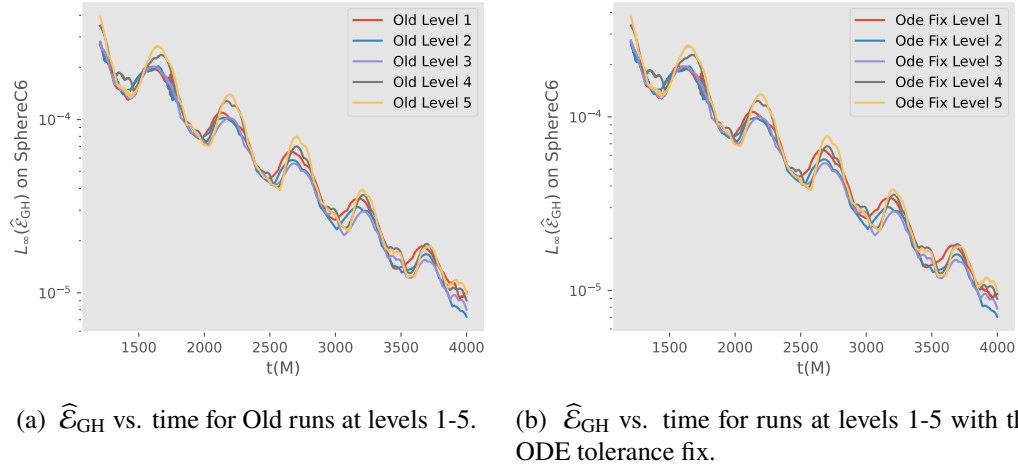


Figure 5.6: Normalized constraint energy density in SphereC6 before and after the ODE tolerance fix. The ODE tolerance fix improves the near-horizon convergence, but the normalized constraints in the wave-extraction region still show poor convergence with level. This motivates adding both more resolution and more constraint damping in the outer subdomains.

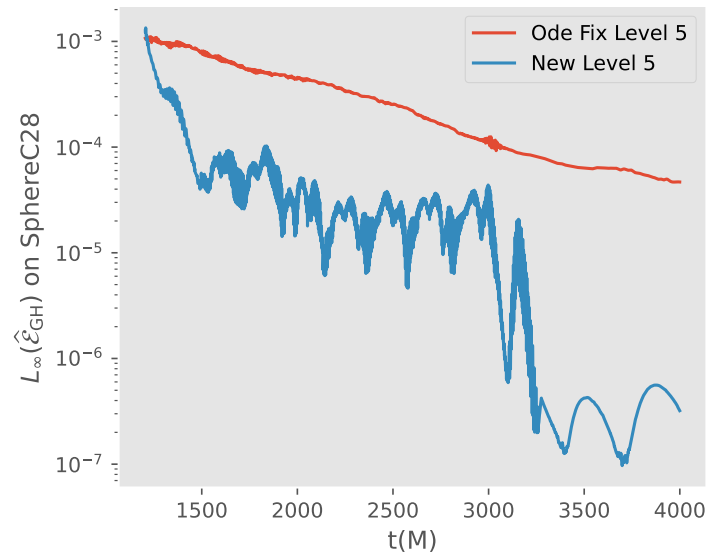


Figure 5.7: $\hat{\mathcal{E}}_{\text{GH}}$ vs. time in SphereC28. Red line is the level 5 run with the ODE tolerance fix, and we can see that the constraints are still going down and haven't saturated yet. Blue line is the level 5 New run, which has more constraint damping, and we can see that the constraints saturate much earlier.

More Constraint Damping and Resolution in the Outer Subdomain

Having resolved the issue in the subdomains near the black holes, we can now focus on the subdomains in the wave extraction region. For now, we will focus on the subdomain SphereC6, one of the first subdomains from which we extract the gravitational wave data. The constraint violations in the SphereC6 are shown in Figure 5.5. Note that these constraint violations are small compared to the constraint violations in the inner subdomain, although there is hardly any convergence as we increase the level. Furthermore, we cannot compare the constraint violations in two different subdomains without normalization because the fields may have very different values in those subdomains. To account for this, we will look at the normalized constraint energy density $\widehat{\mathcal{E}}_{\text{GH}}$ defined in Eq.(5.18). The normalized constraint violations for SphereC6 are shown in Figure 5.6b. Compared to the unnormalized constraint violations, the normalized constraint violations are smaller by a factor of 10 in the inner subdomains and larger by a factor of 10^4 in the regions we extract our gravitational waves from. This is because the fundamental fields fall off rapidly as we move away from the location of the black holes, and normalizing the constraints captures this effect.

As explained in Section 5.3, we try to have more resolution in the inner subdomains because it is the most nonlinear region of the simulation and the region where gradients of the fundamental variables are the largest. However, the large normalized constraint violations in the outer regions suggest that we should have more resolution in the outer subdomains. We did a few test simulations and found that simply increasing the resolution in the outer subdomains does not help until very late into the simulations. From the red curve in Figure 5.7, we can see that the constraints are going down with time, but the rate is very slow, and we end up reaching the merger before the constraints have decayed. This tells us that we need both more resolution and more constraint damping in the outer subdomains.

The amount of constraint damping also changes spatially, as shown in Eq. (5.22). The inner subdomains have more constraint damping than the outer subdomains. Because the asymptotic damping value is small, the constraints in the outer subdomains decay slowly. In short simulations, the run can end before the noise in the outer subdomains has fully decayed. To deal with this, we increased the asymptotic value of the constraint damping in Eq. (5.22) from 0.001 to 0.01. With this extra constraint damping and the increased resolution, we were able to reduce the constraint violations in the outer subdomains, as shown in Figure 5.8. Note, however,

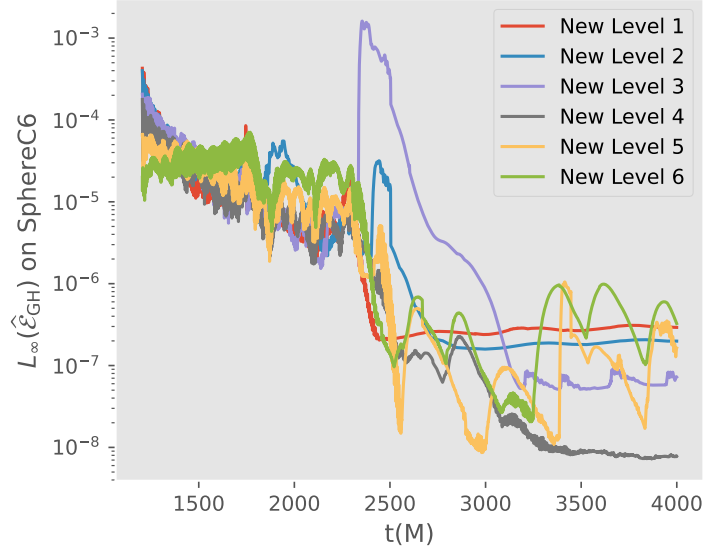


Figure 5.8: $\hat{\mathcal{E}}_{\text{GH}}$ vs time for SphereC6 in the New simulations.

that the constraint violations for levels 5 and 6 are worse than those for level 4. We strongly suspect this is caused by noise in the AMR procedure, described in Section 5.9, because it happens randomly and only in some subdomains.

A natural question is why we cannot increase the asymptotic value of the constraint damping even further. There are two reasons. First, stronger constraint damping makes the junk radiation worse as it propagates through the outer subdomains. Second, stronger constraint damping can make the constraints worse in some situations, as we will see in Section 5.6.

It is also instructive to look at the power monitor plots in Figures 5.9 and 5.10. The simulation shown in Figure 5.10 has more constraint damping, which makes the junk reflections in the subdomains slightly worse. We can see this by comparing a_5 between the two figures. However, after $t = 2500M$, the modes in Figure 5.10 are much better resolved, while the modes in Figure 5.9 are still decaying slowly. The power monitor a_{16} also has a bump around $t = 3400M$, which matches the bump in the level 5 constraint violations in Figure 5.8. We suspect this is caused by an issue with the AMR algorithm, but we have not explored it further.

More Resolution in the Connecting Subdomains for better Waveform Difference

Consider the finite-radius waveform difference between different levels during the early inspiral, starting from $1200M$ (i.e. after most of the junk) and ending at

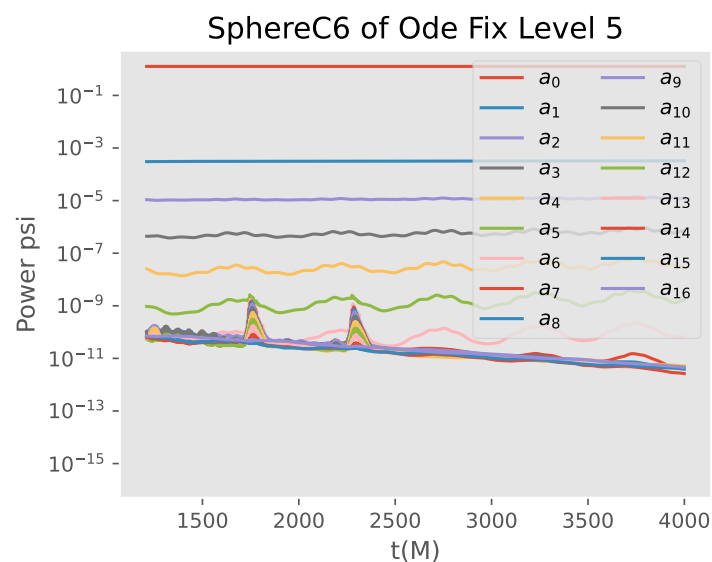


Figure 5.9: Power spectrum of ψ in SphereC6 for a level 5 run with ODE fix.

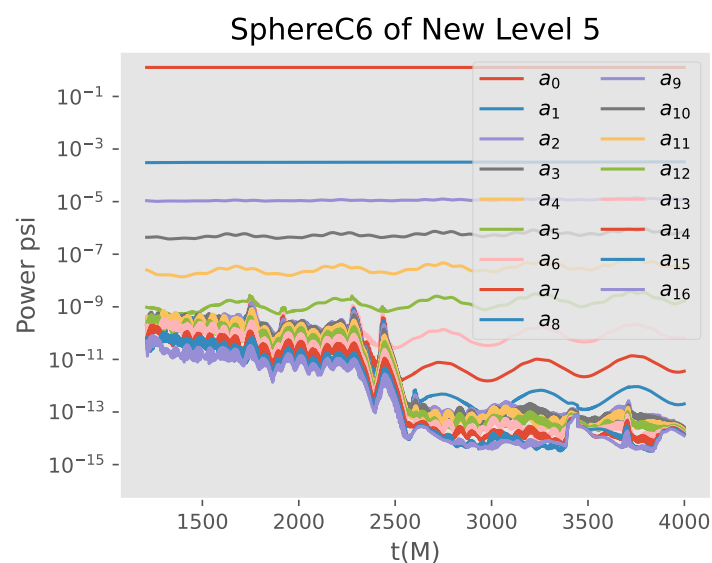


Figure 5.10: Power spectrum of ψ in SphereC6 for a level 5 run with ODE fix and AMR changes.

4000M. The black holes complete around six full orbits in this time frame, allowing us to study the contributions to the waveform difference from the inspiral part of the simulation. Because we only use a few orbits and do not consider the merger part, which has larger errors, the waveform difference values we get are much smaller than the typical waveform differences between two full simulations. Furthermore, we are looking at the waveforms just after PBandJ, and the phase errors did not have enough time to accumulate and increase the waveform difference [63]. These considerations mean that the waveform differences we see are not typical for those of a full simulation; however, they are still useful to diagnose improvements in the algorithm.

Table 5.1 shows the waveform difference for the Old and New runs. We can see that the waveform differences have improved by around two orders of magnitude.

Table 5.1: Finite-radius waveform difference

	Old	New
1 vs 6	3.07×10^{-6}	9.74×10^{-8}
2 vs 6	9.81×10^{-6}	1.68×10^{-7}
3 vs 6	3.07×10^{-7}	5.76×10^{-9}
4 vs 6	5.51×10^{-10}	4.50×10^{-12}
5 vs 6	1.02×10^{-8}	1.81×10^{-11}

Because waveform difference is used as the main predictor of the accuracy of a simulation for data analysis, it is crucial to understand how to reduce it. The general expectation is that the waveform difference should decrease as we increase the resolution. We can see that this is mostly true in Table 5.1. But note that in both cases, the waveform difference between level 2 and level 3 is higher than the waveform difference between level 2 and level 1, even though the constraint violations improve as we increase the level. This behavior is because the waveform difference is more correlated with the difference in the trajectories of the black holes between two levels rather than the constraint violations. By comparing Figures 5.11b and 5.11a, we can see that the waveform difference is strongly correlated with the difference in the trajectories. The x -axis in Figures 5.11b and 5.11a is the time-integrated absolute difference in the coordinate distance of one of the black holes from the origin between two levels, and is supposed to capture the effect of the black holes taking a different trajectory because of different levels.

The main reason for this huge improvement of “New” compared to “Old” is that we are using a constant AMR tolerance for the whole simulation domain in the New

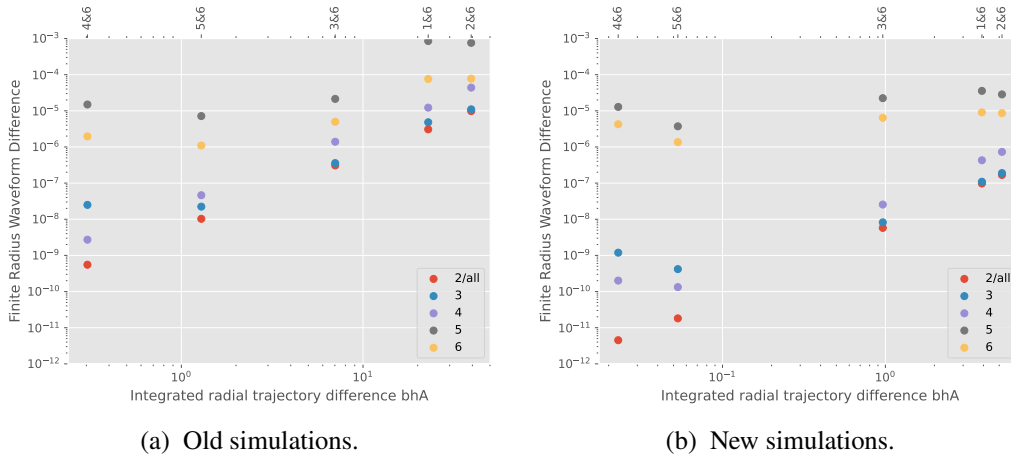


Figure 5.11: Correlation between finite-radius waveform differences and differences in the black-hole trajectories. The horizontal axis at the bottom is the time-integrated absolute difference in the coordinate distance of black hole A from the origin between two levels. The horizontal axis at the top shows the pair of levels used for computing the waveform difference. Different colors represent the waveform difference for various modes. The red dots represent the waveform difference for all the modes, and they overlap with the waveform difference of the (2,2) mode, because that is the dominant mode. In both the Old and New simulations, larger trajectory differences correspond to larger waveform differences. The New simulations reduce waveform differences by two orders of magnitude.

simulations. In particular, this means that the subdomains connecting the two black holes have more resolution for “New”. This extra resolution makes the trajectories of the black holes much more stable across different levels, which improves the waveform difference by a large factor. It should be noted that we have done separate simulations where we did not add more resolution and constraint damping in the outer subdomains, but added more resolution in the subdomains connecting the two black holes. The results were the same: we got two orders of magnitude improvement in the waveform differences. Thus, we can conclude that the primary way to improve the waveform difference is by reducing the drift in the trajectories of the black holes across levels due to numerical truncation error.

We emphasize that waveform difference measures the similarity between two waveforms without any reference to the underlying physics. A small waveform difference means very little unless we also have convergence and small constraint violations. Another limitation of waveform difference is that it measures the worst-case scenario for a given set of simulations. Consider the example of comparing the waveform

difference between four different resolutions: level 1, level 2, level 3, and level 4 as shown in Table 5.1. All these waveform differences are very close in value, and there is no obvious convergence. The best we can say is that all these simulations have similar accuracy, with waveform differences between 10^{-7} and 5×10^{-9} . However, once we have access to level 5 and level 6, we can see that the level 4 simulation was much better than the lower-level simulations. Unfortunately, there is no easy way to tell this in advance; we need level 5 to conclude that level 4 is two orders of magnitude better than level 3. Note that levels 4, 5, and 6 have very similar waveform differences, and level 6 may actually be two orders of magnitude better than level 4, but we cannot claim that without actually doing a level 7 simulation.

For all of our BBH simulations, the waveform difference is dominated by the waveform's $(2, 2)$ mode. This is especially true for the inspiral part of the system we are analyzing. If we neglect all the higher modes, the waveform difference will hardly change. In fact, while the waveform difference for the New simulations improved by two orders of magnitude, the higher-order waveform modes did not improve. We can see this in Figures 5.11b and 5.11a, where modes higher than the fourth mode show little improvement even though the overall waveform difference improved by two orders of magnitude in most cases. We suspect this is mostly caused by the issues discussed later in Section 5.6, which prevent these higher modes from being properly resolved. Note that we are only considering the inspiral portion of an equal-mass, non-spinning simulation far from merger. Close to the merger, these higher modes become much larger and are therefore better resolved.

Computational Cost of these Improvements

Because the improved waveform differences come from adding more grid points to the simulation domain, it is important to ask how much this increases the computational cost. There are two possible reasons that can increase the cost. The first is that the ODE solver might need to take smaller time steps after we add more points because of the Courant limit. The second is that each ODE step might take more wall time because there are more grid points to evolve. For these simulations, the first effect was not important: the ODE step size is limited by the subdomains closest to the black hole horizons, which we did not change. Therefore, the extra resolution in the connecting subdomains did not decrease the time step.

The extra grid points increase the total work per step, so each step becomes somewhat more expensive. However, the increase in wall time per step is modest. SpEC

distributes work across cores in chunks of subdomains. Large subdomains get their own cores, while smaller subdomains are grouped together on a core. The goal is to give each core roughly the same amount of work. However, the work in a single subdomain is not split across multiple cores. For much of the simulation, the core with the largest subdomain takes the longest, while the other cores finish early and wait. The subdomains connecting the two black holes have far fewer grid points than the subdomains near the horizons. Adding resolution to these connecting subdomains, therefore, uses cores that would otherwise have spent more time waiting. In this sense, the improvement partly comes from using SpEC's parallelization more efficiently, so the wall-time increase is much smaller than one might expect from the added grid points alone.

We expect the same argument to hold for other BBH systems, but we have not run a sufficiently large set of simulations with these improvements to quantify the overall cost increase. The key point is that the step size is not reduced, and the increase in cost per step is small enough that the improvements in constraint violations and waveform differences remain worthwhile.

5.6 What limits our accuracy?

Having looked at some improvements, we will now try to understand what prevents us from doing even more accurate simulations in SpEC. Rephrased differently, can we keep on increasing the resolution and expect the accuracy to keep on improving, provided computational resources are not a concern? As one might have guessed, the answer is no, and we go through some reasons for this below.

Truncation errors and floating-point errors

Because we use 64-bit floating-point numbers, there is a fundamental limit to how accurate we can be. Each arithmetic operation can introduce roundoff error of order $\epsilon_{\text{mach}} \sim 10^{-16}$. In practice, each time step in SpEC requires a series of complex operations, which means that the cumulative effect of the floating-point errors is usually much larger. Furthermore, operations like filtering or taking derivatives of the variables also have errors that increase with the resolution, discussed more in Section 5.11.

These floating-point errors are usually small enough to be ignored, but there are subdomains in SpEC simulations where this becomes an issue. For example, in Figure 5.12, the truncation error of the variable ψ in the subdomain SphereA0 is around 10^{-14} , which is close to the finite-precision noise floor, and we have reached

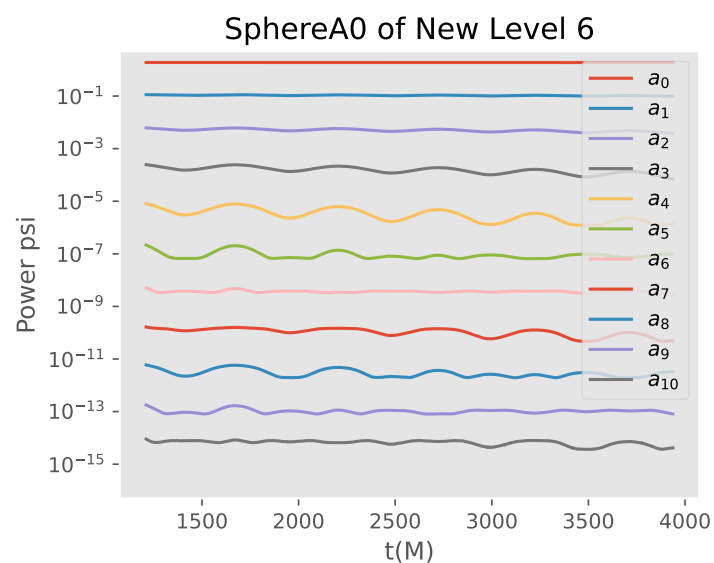


Figure 5.12: Spectral coefficients of ψ in the subdomain SphereA0 for a Level 6 simulation.

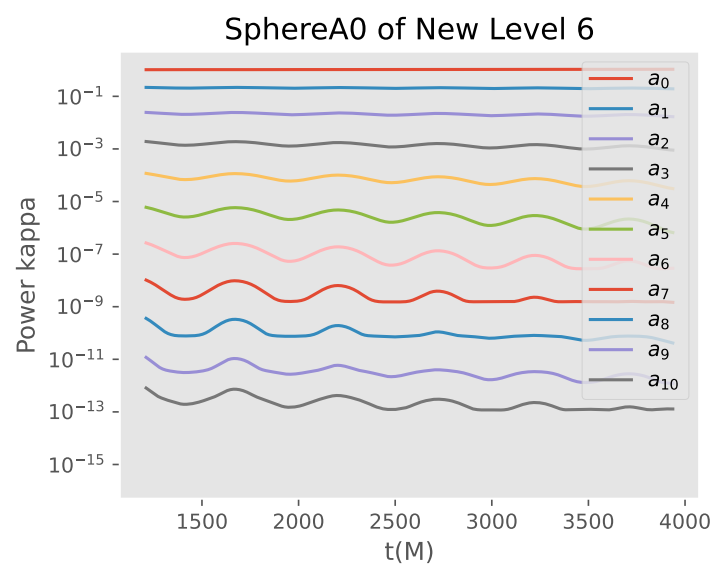


Figure 5.13: Spectral coefficients of κ in the subdomain SphereA0 for a Level 6 simulation.

this by using only 10 points in the radial direction. This is important because it indicates that we cannot just keep on adding more points and expect the simulation to get more and more accurate. Once our truncation error is comparable to this finite-precision noise floor, adding more points not only makes the constraint violations worse but also makes the time step we can take smaller. Not all subdomains have truncation errors comparable to the finite-precision noise floor because, in addition to the truncation error, the SpEC AMR algorithm also considers things like the constraint violations and the accuracy required to track the apparent horizon (Section 5.3). In particular, some subdomains connecting the two black holes converge much more slowly, so we can keep adding more points in those subdomains without hitting the finite-precision limit. As we increase the levels further, we want the AMR to be more selective about the subdomains where it adds more points, which is a topic of future work.

From Figure 5.13, we can also see that the truncation error in the variable κ is only 10^{-13} , which is an order of magnitude larger than that of ψ . This is true in general: the truncation error in the variable κ is always almost an order of magnitude larger than the truncation error in the variable ψ .

Difference in scales of the evolution variables

At a large distance away from the black holes, the spacetime metric(ψ_{ab}) looks like a perturbation to the Minkowski metric.

$$\psi_{ab} \approx \eta_{ab} + \delta\psi_{ab}, \quad \text{where} \\ |\delta\psi_{ab}| \ll |\eta_{ab}|.$$

This means that the diagonal elements of the ψ_{ab} are $O(1)$ while other components are much smaller in magnitude. On the other hand, all the components of the derivative of the spacetime metric are much smaller: $|\kappa_{abc}| \ll 1$. All these variables are used during the evolution in SpEC, and are tightly coupled to each other, either by the way they appear in the GH formalism or because of things like constraint damping.

To understand why this matters, consider the value of these variables at $r \approx 250M$ for a BBH simulation. At this radius, the 00 component of the metric (ψ_{00}) is of $O(1)$, so its spectral derivative can have a roundoff error of $O(10^{-16})$. When we compute the three-index constraint violation (Eq. 5.14) using this derivative, it will inherit this $O(10^{-16})$ error, which means that our three-index constraint will not fall

below an $O(10^{-16})$ finite-precision floor even if we start with the analytical solution. This is something we can see even if we try to evolve a Schwarzschild black hole numerically as discussed in Section 5.10.

One might argue that this error is only present in the computation of the constraint violations, and the fields would still be accurate. However, when we do constraint damping(5.24), the errors in the constraint damping get coupled with the time derivatives of the evolution variables. This coupling means that the evolution variables have a constant source of absolute error, irrespective of the actual field values. The fields themselves fall rapidly as we go away from the center; therefore, the effect of this constant source of error is that the relative error in the fields grows as we go further away from the center. This can be seen in the spectral coefficients of the fields in the outer subdomains. For example, in Figure 5.14, we can see that the highest coefficients have a noise floor around 10^{-15} , and we have no convergence beyond that point, even though the scale of κ_{abc} in this subdomain is about 10^{-5} . The direct implication of this noise floor is that the truncation *relative* error grows as we move further away from the center, and we cannot improve it by adding more points. Note that all this only becomes relevant once the residual junk radiation has either left the domain or has been damped away, which happens around $t = 2500M$ in Figure 5.14. This also tells us that there is a sweet spot for how much constraint damping we can do because more constraint damping will only help up to a certain extent, after which the errors from the three index constraints will start dominating. Unfortunately, this sweet spot depends on the resolution of the grid (which controls the error in the derivative) and the value of the fields, which change with time. Ideally, we want to have an algorithm that controls the amount of constraint damping based on these considerations, but a more detailed study of the problem is required before we can implement that.

While having a better algorithm for the constraint damping is the easiest first thing to try, it does not solve the more fundamental problem of having very different scales in the evolution variables. Ideally, we want to scale our fundamental variables to have similar magnitudes in the region of interest, but it is unclear how to do that in the GH system.

Another possibility is to make the spacetime minus the Minkowski metric one of our fundamental variables, $\bar{\psi}_{ab} = \psi_{ab} - \eta_{ab}$. This will not entirely solve the problem because ψ and κ still fall off at a different rate, but it improves things by a few orders of magnitude because we eliminate the constant 1 from the diagonal terms.

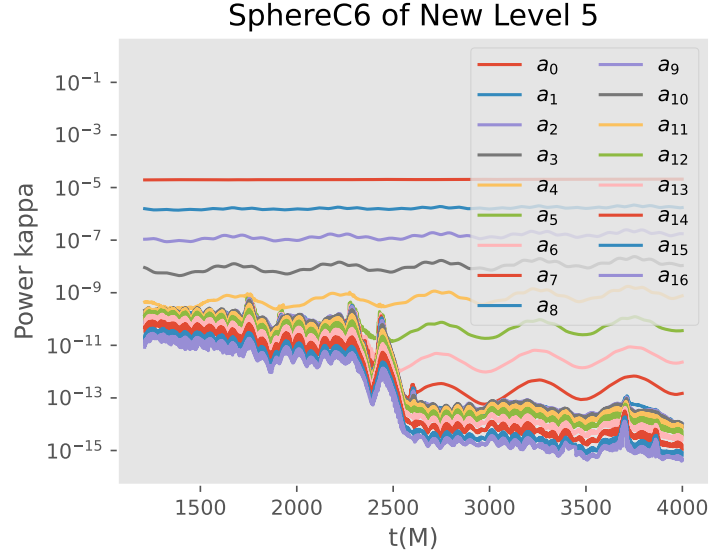


Figure 5.14: Power spectrum of κ in SphereC6 for a level 5 run with ODE fix and AMR changes.

Consider Figure 5.10, which shows the power spectrum for the variable ψ . If we evolve $\psi_{ab} - \eta_{ab}$ instead, then we basically remove the a_0 component, which could, in theory, allow us to resolve things better. Note that a_1 from Figure 5.10 is still a factor of 10 larger than the value of a_0 in Figure 5.14, so this problem does not go away fully, but it is better than the four orders of magnitude difference that we had before. We have done some single black hole simulations using the Minkowski subtracted metric as the fundamental variable and noticed improvements in the constraints for the outer subdomains. A topic of future work is to do similar simulations for a BBH system.

We have already noted that the constraint violations do not converge after a certain resolution in the outer subdomains. At the same time, they continue to converge in the inner subdomains (Figure 5.2d). We suspect the floating-point noise discussed above is one of the reasons for this. The fields in the outer domains are much smoother than those in the inner subdomains; therefore, they converge much faster, and we quickly hit the noise from calculating the constraint violations. For the simulation shown in Figure 5.6b, we reach this noise floor by level 4, and adding points beyond that does not improve the constraint, as is the case for levels 5 and 6. We should mention that the large increase in the constraint violations for levels 5 and 6 in the figure is mostly caused by the glitches in the AMR (Section 5.9) and not because we are adding more points after the spectral coefficients have saturated

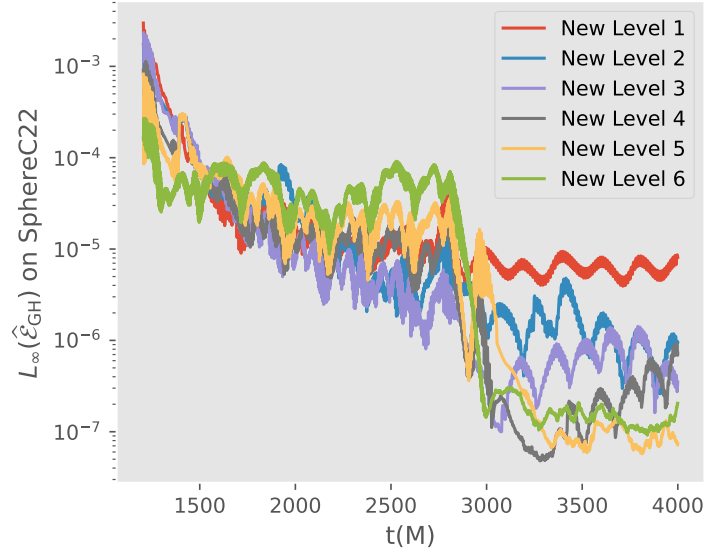


Figure 5.15: $\hat{\mathcal{E}}_{\text{GH}}$ for New runs in the SphereC22.

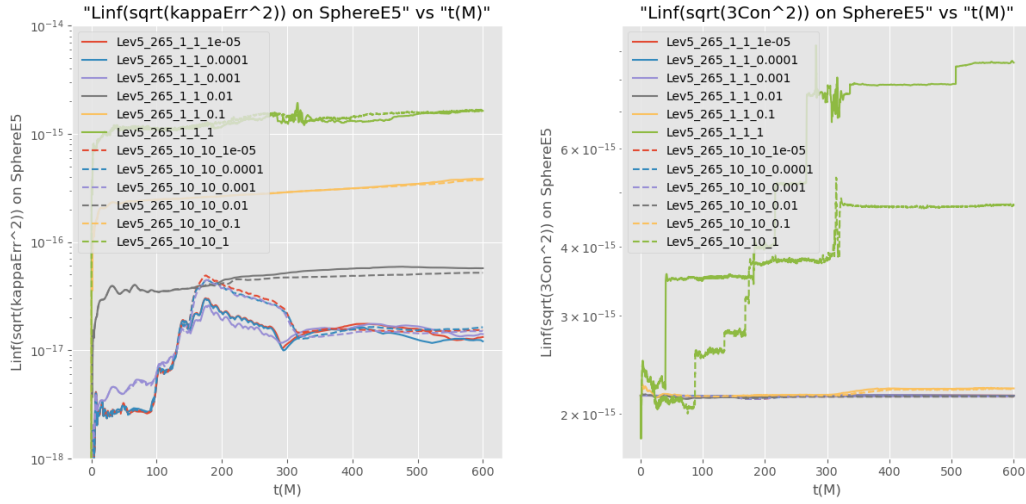
(Section 5.11). This noise floor increases as we move further and further away from the center, which is why we see even less convergence in the subdomains that are closer to the outer boundary, like SphereC22, as shown in Figure 5.15.

5.7 Conclusions

In this paper, we studied the accuracy of SpEC BBH simulations during the inspiral. We found that the constraint convergence near the black holes was limited by the ODE tolerance, while reducing the constraint violations in the wave extraction region required both more resolution and more constraint damping. We also found that increasing the resolution of the subdomains connecting the two black holes made the black-hole trajectories more stable across different levels. With these changes, the finite-radius waveform differences improved by around two orders of magnitude.

These improvements did not reduce the ODE step size in the simulations studied here because the additional grid points were added mainly in subdomains that were not limiting the time step. They do increase the work per step and, therefore, the wall time per step, but the increase is modest because much of the extra work is assigned to cores that would otherwise spend more time waiting. This means that the accuracy of SpEC inspiral simulations can be improved substantially without a comparable increase in computational cost.

We also found that convergence in the outer subdomains at higher levels is affected



(a) Error in κ for a Schwarzschild black hole in (b) 3-index constraints of a Schwarzschild black hole the outermost subdomains ending at $r=250M$. The hole in the outermost subdomains ending at truncation error for these simulations was 10^{-13} $r=250M$. The truncation error for these simulations and the ODE tolerance was 10^{-12} . The number 265 represents the outer radii, the next two numbers represent the values of γ_0 and γ_2 near the horizon. The values of γ_0 and γ_2 exponentially fall off to the last number in the label. We can see that the error in κ depends on the asymptotic value of the constraint damping parameters (γ_0 and γ_2), getting worse as we increase their value. The number 256 represents the outer radii, the next two numbers represent the values of γ_0 and γ_2 near the horizon. The values of γ_0 and γ_2 exponentially fall off to the last number in the label. We can see that the three index constraint is stationary in time at a value of 10^{-15} irrespective of the constraint damping parameters until γ_0 and γ_2 are so big that the error in the κ starts to dominate the errors in the computation of the three index constraints.

Figure 5.16: Single-black-hole tests showing errors in the outermost subdomains. The left panel shows that the error in κ increases when the asymptotic values of the constraint damping parameters are increased. The right panel shows that the three-index constraint reaches a roughly stationary floor near 10^{-15} until the error in κ becomes large enough to dominate the constraint calculation.

by floating-point errors and AMR-related glitches. This study shows that we might be able to achieve LISA-accurate simulations with the improvements we made. The next aim is to run longer end-to-end simulations and check this directly.

5.8 Generalized Harmonic System

Full evolution equations in the GH formalism are:

$$\begin{aligned}
& \partial_t \psi_{ab} - (1 + \gamma_1) N^k \partial_k \psi_{ab} = -N \Pi_{ab} - \gamma_1 N^i \Phi_{iab}, \\
& \partial_t \Pi_{ab} - N^k \partial_k \Pi_{ab} + N g^{ki} \partial_k \Phi_{iab} - \gamma_1 \gamma_2 N^k \partial_k \psi_{ab} \\
& \quad = 2N \psi^{cd} \left(g^{ij} \Phi_{ica} \Phi_{jdb} - \Pi_{ca} \Pi_{db} - \psi^{ef} \Gamma_{ace} \Gamma_{bdf} \right) \\
& \quad \quad - 2N \nabla_{(a} H_{b)} - \frac{1}{2} N t^c t^d \Pi_{cd} \Pi_{ab} - N t^c \Pi_{ci} g^{ij} \Phi_{jab} \\
& \quad \quad + N \gamma_0 \left[2\delta^c_{(a} t_{b)} - \psi_{ab} t^c \right] (H_c + \Gamma_c) - \gamma_1 \gamma_2 N^i \Phi_{iab}, \\
& \partial_t \Phi_{iab} - N^k \partial_k \Phi_{iab} + N \partial_i \Pi_{ab} - N \gamma_2 \partial_i \psi_{ab} \\
& \quad = \frac{1}{2} N t^c t^d \Phi_{icd} \Pi_{ab} + N g^{jk} t^c \Phi_{ijc} \Phi_{kab} - N \gamma_2 \Phi_{iab}.
\end{aligned} \tag{5.24}$$

Variables γ_0, γ_1 and γ_2 are free constants that are used for adjusting characteristic speeds and for constraint damping [38], Section 5.4. Note that terms involving each of γ_0, γ_1 and γ_2 sum to zero if the constraints are exactly satisfied.

5.9 Glitches due to grid changes

In addition to the junk radiation, an unresolved but understood source of error, Figure 5.17b also shows random error spikes in some outer subdomains (eg. SphereC5, SphereC29). These glitches happen randomly in the middle of the simulation and usually do not propagate outward like the junk radiation. We strongly suspect that these are caused by the AMR algorithm as it tries to change the grid resolution. The idea is that each AMR change introduces a small discontinuity in time where the dynamical variables are suddenly represented by a different number of basis functions. Ideally, these discontinuities should be below the truncation error, but because they are discontinuous and occur at unpredictable times, they introduce a small source of noise into the solution.

To test whether the glitches are indeed caused by AMR, we did some simulations where the AMR was turned off, but the grid resolution was manually set to be similar to what AMR would have chosen, and we did not observe any glitches in these simulations. In most cases, these glitches are not really relevant because they are relatively rare and only affect the outer subdomains. We mention them here because we see them in heatmap figures, for example, in the subdomain SphereC29 in Figure 5.17b and some constraints violations plots like Figure 5.8.

Table 5.2: Initial-data and relaxed parameters for the simulations used in this paper. The New and Old/ODE fix rows are measured after PBandJ.

Case	e	M_A	M_B	χ_A	χ_B
ID	–	0.4999999739	0.4999999688	$\begin{pmatrix} -9.194 \times 10^{-11} \\ -9.288 \times 10^{-12} \\ -2.138 \times 10^{-8} \end{pmatrix}$	$\begin{pmatrix} 2.648 \times 10^{-11} \\ 3.037 \times 10^{-11} \\ -2.545 \times 10^{-8} \end{pmatrix}$
New	0.0515847	0.4999998464	0.4999998413	$\begin{pmatrix} 6.140 \times 10^{-14} \\ 2.002 \times 10^{-15} \\ -2.651 \times 10^{-7} \end{pmatrix}$	$\begin{pmatrix} 1.894 \times 10^{-14} \\ -3.425 \times 10^{-16} \\ -2.612 \times 10^{-7} \end{pmatrix}$
Old/ODE fix	0.0540567	0.5000004262	0.5000004209	$\begin{pmatrix} 2.670 \times 10^{-12} \\ 4.792 \times 10^{-12} \\ -2.027 \times 10^{-6} \end{pmatrix}$	$\begin{pmatrix} -2.068 \times 10^{-12} \\ -4.602 \times 10^{-12} \\ -2.026 \times 10^{-6} \end{pmatrix}$

5.10 Single black hole tests

We have already explained in Section 5.6 why we do not expect the three-index constraint to fall below a finite-precision floor set by machine precision ($\epsilon_{\text{mach}} \simeq 2 \times 10^{-16}$) and amplified by spectral differentiation, irrespective of the resolution or simplicity of the solution. In Figure 5.16b, we show the three index-constraint in the outermost subdomain of a Schwarzschild black hole. We can see that the constraint is constant around 2×10^{-15} . In the same subdomain, one-index and two-index constraints are two orders of magnitude better.

5.11 Errors in spectral derivatives

When using spectral methods, the important error to pay attention to is the truncation error. In most situations, roundoff errors are much smaller than the spectral truncation error and can be ignored. However, when the truncation error approaches the finite-precision noise floor, derivative operations can amplify roundoff errors. Roughly speaking, the contribution from a floating-point error δ to the m^{th} derivative of a function represented by N points is δN^{2m} [64]. This means that even if we are using as few as 10 collocation points and the roundoff error is 2×10^{-16} , the moment we take a derivative, the error in the derivative will be about 2×10^{-14} .

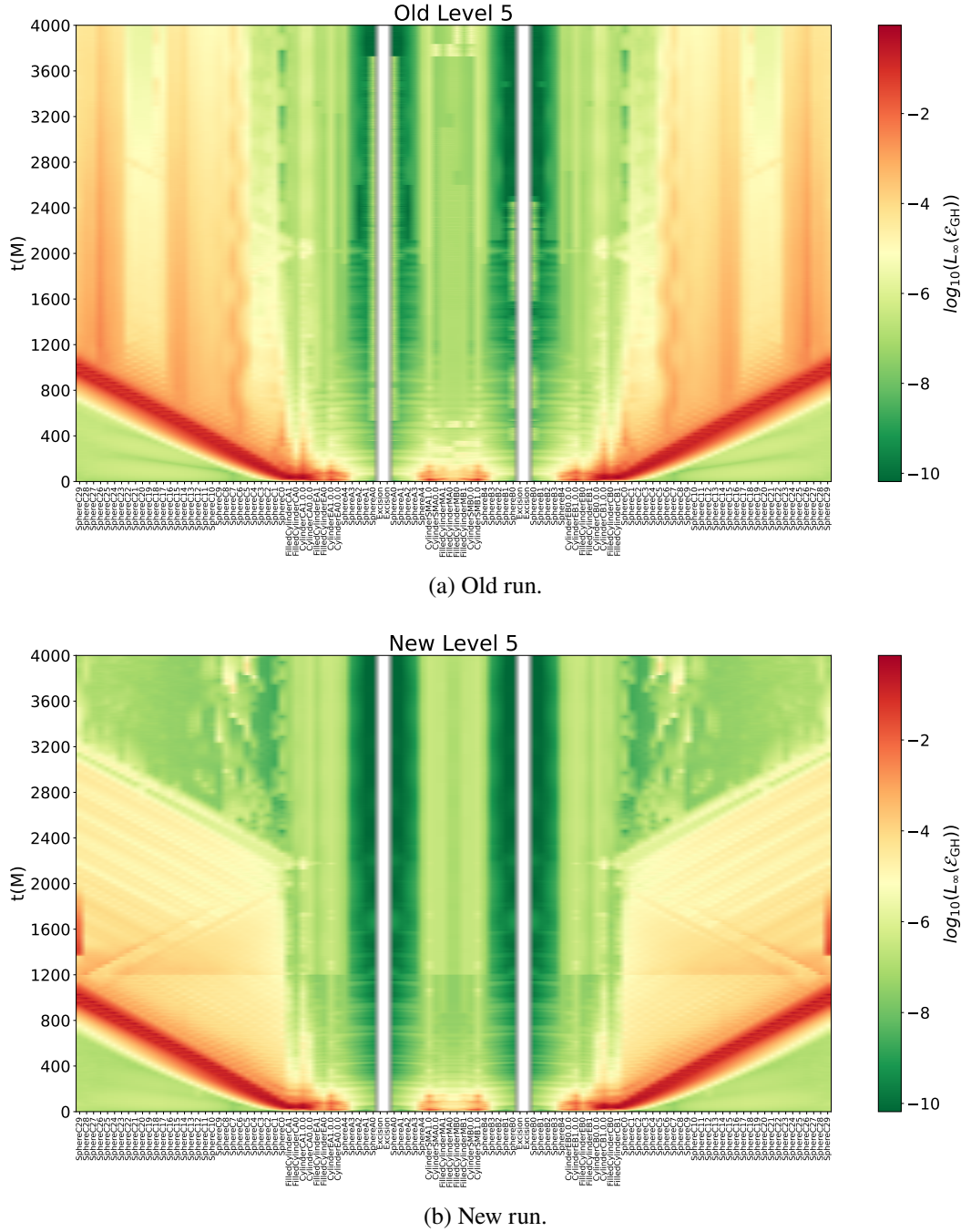


Figure 5.17: Normalized constraint-violation heatmaps for the Old simulation from the main branch and the New simulation with all the improvements. The vertical axis is simulation time, and the horizontal axis labels the various subdomains from the inner region to the outer boundary. The vertical grey areas are the location of the black hole horizons, which are excised from the simulation domain. The bright diagonal feature is the junk-radiation pulse propagating outward through the domain. Note how the constraints in the outer subdomains are much better for the New run once the junk radiation leaves the domain.

5.12 Parameters of the systems

We started the simulations with the exact same initial data for all our tests. The input parameters for the initial data solver were:

$$\text{Total mass (M)} = 1.0,$$

$$\text{Mass ratio (q)} = 1,$$

$$\text{Spin black hole A } (\chi_A) = (0, 0, 0)$$

$$\text{Spin black hole B } (\chi_B) = (0, 0, 0)$$

$$\text{Separation (d)} = 18.0M,$$

$$\text{Orbital angular velocity } (\Omega_0) = 0.012226459838/M,$$

$$\text{Radial expansion parameter } (\dot{a}_0) = -1.72797015626 \times 10^{-5}/M.$$

Relaxed parameters and initial data solver output parameters are shown in Table 5.2.

References

- [1] LIGO Scientific Collaboration, J. Aasi, B. P. Abbott, et al. “Advanced LIGO”. In: *Classical and Quantum Gravity* 32.7, 074001 (Apr. 2015), p. 074001. DOI: 10.1088/0264-9381/32/7/074001. arXiv: 1411.4547 [gr-qc].
- [2] F. Acernese, M. Agathos, K. Agatsuma, et al. “Advanced Virgo: a second-generation interferometric gravitational wave detector”. In: *Classical and Quantum Gravity* 32.2, 024001 (Jan. 2015), p. 024001. DOI: 10.1088/0264-9381/32/2/024001. arXiv: 1408.3978 [gr-qc].
- [3] B. P. Abbott, R. Abbott, T. D. Abbott, et al. “Observation of Gravitational Waves from a Binary Black Hole Merger”. In: *Phys. Rev. Lett.* 116.6, 061102 (Feb. 2016), p. 061102. DOI: 10.1103/PhysRevLett.116.061102. arXiv: 1602.03837 [gr-qc].
- [4] B. P. Abbott, R. Abbott, T. D. Abbott, et al. “GW150914: First results from the search for binary black hole coalescence with Advanced LIGO”. In: *Phys. Rev. D* 93.12, 122003 (June 2016), p. 122003. DOI: 10.1103/PhysRevD.93.122003. arXiv: 1602.03839 [gr-qc].
- [5] B. P. Abbott, R. Abbott, T. D. Abbott, et al. “GW150914: The Advanced LIGO Detectors in the Era of First Discoveries”. In: *Phys. Rev. Lett.* 116.13, 131103 (Apr. 2016), p. 131103. DOI: 10.1103/PhysRevLett.116.131103. arXiv: 1602.03838 [gr-qc].
- [6] B. P. Abbott, R. Abbott, T. D. Abbott, et al. “Properties of the Binary Black Hole Merger GW150914”. In: *Phys. Rev. Lett.* 116.24, 241102 (June 2016), p. 241102. DOI: 10.1103/PhysRevLett.116.241102. arXiv: 1602.03840 [gr-qc].

- [7] B. P. Abbott, R. Abbott, T. D. Abbott, et al. “GWTC-1: A Gravitational-Wave Transient Catalog of Compact Binary Mergers Observed by LIGO and Virgo during the First and Second Observing Runs”. In: *Physical Review X* 9.3, 031040 (July 2019), p. 031040. DOI: 10.1103/PhysRevX.9.031040. arXiv: 1811.12907 [astro-ph.HE].
- [8] R. Abbott, T. D. Abbott, S. Abraham, et al. “GWTC-2: Compact Binary Coalescences Observed by LIGO and Virgo during the First Half of the Third Observing Run”. In: *Physical Review X* 11.2, 021053 (Apr. 2021), p. 021053. DOI: 10.1103/PhysRevX.11.021053. arXiv: 2010.14527 [gr-qc].
- [9] R. Abbott, T. D. Abbott, F. Acernese, et al. “Population of Merging Compact Binaries Inferred Using Gravitational Waves through GWTC-3”. In: *Physical Review X* 13.1, 011048 (Jan. 2023), p. 011048. DOI: 10.1103/PhysRevX.13.011048. arXiv: 2111.03634 [astro-ph.HE].
- [10] R. Abbott, T. D. Abbott, F. Acernese, et al. “GWTC-3: Compact Binary Coalescences Observed by LIGO and Virgo during the Second Part of the Third Observing Run”. In: *Physical Review X* 13.4, 041039 (Oct. 2023), p. 041039. DOI: 10.1103/PhysRevX.13.041039. arXiv: 2111.03606 [gr-qc].
- [11] Alexander H. Nitz, Sumit Kumar, Yi-Fan Wang, et al. “4-OGC: Catalog of Gravitational Waves from Compact Binary Mergers”. In: *ApJ* 946.2, 59 (Apr. 2023), p. 59. DOI: 10.3847/1538-4357/aca591. arXiv: 2112.06878 [astro-ph.HE].
- [12] Nicolás Yunes, Xavier Siemens, and Kent Yagi. “Gravitational-wave tests of general relativity with ground-based detectors and pulsar-timing arrays”. In: *Living Reviews in Relativity* 28.1, 3 (Mar. 2025), p. 3. DOI: 10.1007/s41114-024-00054-9. arXiv: 2408.05240 [gr-qc].
- [13] Pau Amaro-Seoane, Heather Audley, Stanislav Babak, et al. “Laser Interferometer Space Antenna”. In: *arXiv e-prints*, arXiv:1702.00786 (Feb. 2017), arXiv:1702.00786. DOI: 10.48550/arXiv.1702.00786. arXiv: 1702.00786 [astro-ph.IM].
- [14] Matthew Evans, Alessandra Corsi, Chaitanya Afle, et al. “Cosmic Explorer: A Submission to the NSF MPSAC ngGW Subcommittee”. In: *arXiv e-prints*, arXiv:2306.13745 (June 2023), arXiv:2306.13745. DOI: 10.48550/arXiv.2306.13745. arXiv: 2306.13745 [astro-ph.IM].
- [15] Michele Maggiore, Chris Van Den Broeck, Nicola Bartolo, et al. “Science case for the Einstein telescope”. In: *J. Cosmology Astropart. Phys.* 2020.3, 050 (Mar. 2020), p. 050. DOI: 10.1088/1475-7516/2020/03/050. arXiv: 1912.02622 [astro-ph.CO].
- [16] Adrian Abac, Raul Abramo, Simone Albanesi, et al. “The Science of the Einstein Telescope”. In: *J. Cosmology Astropart. Phys.* 2026.3, 081 (Mar.

- 2026), p. 081. doi: 10.1088/1475-7516/2026/03/081. arXiv: 2503.12263 [gr-qc].
- [17] Jonathan Blackman, Scott E. Field, Mark A. Scheel, et al. “A Surrogate model of gravitational waveforms from numerical relativity simulations of precessing binary black hole mergers”. In: *Phys. Rev. D* 95.10, 104023 (May 2017), p. 104023. doi: 10.1103/PhysRevD.95.104023. arXiv: 1701.00550 [gr-qc].
 - [18] Jonathan Blackman, Scott E. Field, Chad R. Galley, et al. “Fast and Accurate Prediction of Numerical Relativity Waveforms from Binary Black Hole Coalescences Using Surrogate Models”. In: *Phys. Rev. Lett.* 115.12, 121102 (Sept. 2015), p. 121102. doi: 10.1103/PhysRevLett.115.121102. arXiv: 1502.07758 [gr-qc].
 - [19] Vijay Varma, Scott E. Field, Mark A. Scheel, et al. “Surrogate model of hybridized numerical relativity binary black hole waveforms”. In: *Phys. Rev. D* 99.6, 064045 (Mar. 2019), p. 064045. doi: 10.1103/PhysRevD.99.064045. arXiv: 1812.07865 [gr-qc].
 - [20] Jonathan Blackman, Scott E. Field, Mark A. Scheel, et al. “Numerical relativity waveform surrogate model for generically precessing binary black hole mergers”. In: *Phys. Rev. D* 96.2, 024058 (July 2017), p. 024058. doi: 10.1103/PhysRevD.96.024058. arXiv: 1705.07089 [gr-qc].
 - [21] Vijay Varma, Scott E. Field, Mark A. Scheel, et al. “Surrogate models for precessing binary black hole simulations with unequal masses”. In: *Physical Review Research* 1.3, 033015 (Oct. 2019), p. 033015. doi: 10.1103/PhysRevResearch.1.033015. arXiv: 1905.09300 [gr-qc].
 - [22] A. Buonanno and T. Damour. “Effective one-body approach to general relativistic two-body dynamics”. In: *Phys. Rev. D* 59.8, 084006 (Apr. 1999), p. 084006. doi: 10.1103/PhysRevD.59.084006. arXiv: gr-qc/9811091 [gr-qc].
 - [23] Alessandra Buonanno and Thibault Damour. “Transition from inspiral to plunge in binary black hole coalescences”. In: *Phys. Rev. D* 62.6, 064015 (Sept. 2000), p. 064015. doi: 10.1103/PhysRevD.62.064015. arXiv: gr-qc/0001013 [gr-qc].
 - [24] Alessandra Buonanno, Yanbei Chen, and Thibault Damour. “Transition from inspiral to plunge in precessing binaries of spinning black holes”. In: *Phys. Rev. D* 74.10, 104005 (Nov. 2006), p. 104005. doi: 10.1103/PhysRevD.74.104005. arXiv: gr-qc/0508067 [gr-qc].
 - [25] Thibault Damour, Piotr Jaranowski, and Gerhard Schäfer. “Determination of the last stable orbit for circular general relativistic binaries at the third post-Newtonian approximation”. In: *Phys. Rev. D* 62.8, 084011 (Oct. 2000), p. 084011. doi: 10.1103/PhysRevD.62.084011. arXiv: gr-qc/0005034 [gr-qc].

- [26] Thibault Damour. “Introductory Lectures on the Effective One Body Formalism”. In: *International Journal of Modern Physics A* 23.8 (Jan. 2008), pp. 1130–1148. doi: 10.1142/S0217751X08039992. arXiv: 0802.4047 [gr-qc].
- [27] Thibault Damour. “Introductory Lectures on the Effective One Body Formalism”. In: *International Journal of Modern Physics A* 23.8 (Jan. 2008), pp. 1130–1148. doi: 10.1142/S0217751X08039992. arXiv: 0802.4047 [gr-qc].
- [28] Antoni Ramos-Buades, Alessandra Buonanno, Héctor Estellés, et al. “Next generation of accurate and efficient multipolar precessing-spin effective-one-body waveforms for binary black holes”. In: *Phys. Rev. D* 108.12, 124037 (Dec. 2023), p. 124037. doi: 10.1103/PhysRevD.108.124037. arXiv: 2303.18046 [gr-qc].
- [29] Michael Pürrer and Carl-Johan Haster. “Gravitational waveform accuracy requirements for future ground-based detectors”. In: *Physical Review Research* 2.2, 023151 (May 2020), p. 023151. doi: 10.1103/PhysRevResearch.2.023151. arXiv: 1912.10055 [gr-qc].
- [30] LISA Consortium Waveform Working Group, Niayesh Afshordi, Sarp Akçay, et al. “Waveform modelling for the Laser Interferometer Space Antenna”. In: *Living Reviews in Relativity* 28.1, 9 (Oct. 2025), p. 9. doi: 10.1007/s41114-025-00056-1. arXiv: 2311.01300 [gr-qc].
- [31] Nils Deppe et al. *SpECTRE*. Version 2026.04.01. Apr. 2026. doi: 10.5281/zenodo.19373346. URL: <https://doi.org/10.5281/zenodo.19373346>.
- [32] Mark A. Scheel et al. “The SXS collaboration’s third catalog of binary black hole simulations”. In: *Classical and Quantum Gravity* 42.19, 195017 (Oct. 2025), p. 195017. doi: 10.1088/1361-6382/adfd34. arXiv: 2505.13378 [gr-qc].
- [33] Thomas W. Baumgarte and Stuart L. Shapiro. *Numerical Relativity*. 2010.
- [34] Masaru Shibata. *Numerical Relativity*. 2016. doi: 10.1142/9692.
- [35] Helmut Friedrich. “On the hyperbolicity of Einstein’s and other gauge field equations”. In: *Communications in Mathematical Physics* 100.4 (Dec. 1985), pp. 525–543. doi: 10.1007/BF01217728.
- [36] David Garfinkle. “Harmonic coordinate method for simulating generic singularities”. In: *Phys. Rev. D* 65.4, 044029 (Feb. 2002), p. 044029. doi: 10.1103/PhysRevD.65.044029. arXiv: gr-qc/0110013 [gr-qc].
- [37] Frans Pretorius. “Numerical relativity using a generalized harmonic decomposition”. In: *Classical and Quantum Gravity* 22.2 (Jan. 2005), pp. 425–451. doi: 10.1088/0264-9381/22/2/014. arXiv: gr-qc/0407110 [gr-qc].

- [38] Lee Lindblom et al. “A new generalized harmonic evolution system”. In: *Classical and Quantum Gravity* 23.16 (Aug. 2006), S447–S462. DOI: 10.1088/0264-9381/23/16/S09. arXiv: gr-qc/0512093 [gr-qc].
- [39] Béla Szilágyi, Lee Lindblom, and Mark A. Scheel. “Simulations of binary black hole mergers using spectral methods”. In: *Phys. Rev. D* 80.12, 124010 (Dec. 2009), p. 124010. DOI: 10.1103/PhysRevD.80.124010. arXiv: 0909.3557 [gr-qc].
- [40] Mark A. Scheel, Michael Boyle, Tony Chu, et al. “High-accuracy waveforms for binary black hole inspiral, merger, and ringdown”. In: *Phys. Rev. D* 79.2, 024003 (Jan. 2009), p. 024003. DOI: 10.1103/PhysRevD.79.024003. arXiv: 0810.1767 [gr-qc].
- [41] Daniel A. Hemberger, Mark A. Scheel, Lawrence E. Kidder, et al. “Dynamical excision boundaries in spectral evolutions of binary black hole spacetimes”. In: *Classical and Quantum Gravity* 30.11, 115001 (June 2013), p. 115001. DOI: 10.1088/0264-9381/30/11/115001. arXiv: 1211.6079 [gr-qc].
- [42] Serguei Ossokine, Lawrence E. Kidder, and Harald P. Pfeiffer. “Precession-tracking coordinates for simulations of compact-object binaries”. In: *Phys. Rev. D* 88.8, 084031 (Oct. 2013), p. 084031. DOI: 10.1103/PhysRevD.88.084031. arXiv: 1304.3067 [gr-qc].
- [43] O. Rinne. “Stable radiation-controlling boundary conditions for the generalized harmonic Einstein equations”. In: *Classical and Quantum Gravity* 23.22 (Nov. 2006), pp. 6275–6300. DOI: 10.1088/0264-9381/23/22/013. arXiv: gr-qc/0606053 [gr-qc].
- [44] Oliver Rinne, Lee Lindblom, and Mark A. Scheel. “Testing outer boundary treatments for the Einstein equations”. In: *Classical and Quantum Gravity* 24.16 (Aug. 2007), pp. 4053–4078. DOI: 10.1088/0264-9381/24/16/006. arXiv: 0704.0782 [gr-qc].
- [45] Morten Bjørhus. “The ODE Formulation of Hyperbolic PDEs Discretized by the Spectral Collocation Method”. In: *SIAM Journal on Scientific Computing* 16.3 (May 1995), pp. 542–557. DOI: 10.1137/0916035.
- [46] Béla Szilágyi. “Key elements of robustness in binary black hole evolutions using spectral methods”. In: *International Journal of Modern Physics D* 23.7, 1430014 (May 2014), p. 1430014. DOI: 10.1142/S0218271814300146. arXiv: 1405.3693 [gr-qc].
- [47] Luisa T. Buchman et al. “Simulations of unequal-mass black hole binaries with spectral methods”. In: *Physical Review D* 86.8 (Oct. 2012). ISSN: 1550-2368. DOI: 10.1103/physrevd.86.084033. URL: <http://dx.doi.org/10.1103/PhysRevD.86.084033>.

- [48] J.R. Dormand and P.J. Prince. “A family of embedded Runge-Kutta formulae”. In: *Journal of Computational and Applied Mathematics* 6.1 (1980), pp. 19–26. ISSN: 0377-0427. DOI: [https://doi.org/10.1016/0771-050X\(80\)90013-3](https://doi.org/10.1016/0771-050X(80)90013-3). URL: <https://www.sciencedirect.com/science/article/pii/0771050X80900133>.
- [49] Luisa T. Buchman et al. “Simulations of unequal-mass black hole binaries with spectral methods”. In: *Phys. Rev. D* 86.8, 084033 (Oct. 2012), p. 084033. DOI: [10.1103/PhysRevD.86.084033](https://doi.org/10.1103/PhysRevD.86.084033). arXiv: 1206.3015 [gr-qc].
- [50] Serguei Ossokine et al. “Improvements to the construction of binary black hole initial data”. In: *Classical and Quantum Gravity* 32.24, 245010 (Dec. 2015), p. 245010. DOI: [10.1088/0264-9381/32/24/245010](https://doi.org/10.1088/0264-9381/32/24/245010). arXiv: 1506.01689 [gr-qc].
- [51] Lee Lindblom et al. “A new generalized harmonic evolution system”. In: *Classical and Quantum Gravity* 23.16 (Aug. 2006), S447–S462. DOI: [10.1088/0264-9381/23/16/S09](https://doi.org/10.1088/0264-9381/23/16/S09). arXiv: gr-qc/0512093 [gr-qc].
- [52] Michael Boyle, Daniel Hemberger, Dante A. B. Izzo, et al. “The SXS collaboration catalog of binary black hole simulations”. In: *Classical and Quantum Gravity* 36.19, 195006 (Oct. 2019), p. 195006. DOI: [10.1088/1361-6382/ab34e2](https://doi.org/10.1088/1361-6382/ab34e2). arXiv: 1904.04831 [gr-qc].
- [53] Olivier Sarbach and Manuel Tiglio. “Gauge-invariant perturbations of Schwarzschild black holes in horizon-penetrating coordinates”. In: *Phys. Rev. D* 64.8 (Oct. 2001), p. 084016. DOI: [10.1103/PhysRevD.64.084016](https://doi.org/10.1103/PhysRevD.64.084016). arXiv: gr-qc/0104061 [gr-qc].
- [54] Tullio Regge and John A. Wheeler. “Stability of a Schwarzschild Singularity”. In: *Physical Review* 108.4 (Nov. 1957), pp. 1063–1069. DOI: [10.1103/PhysRev.108.1063](https://doi.org/10.1103/PhysRev.108.1063).
- [55] Frank J. Zerilli. “Effective Potential for Even-Parity Regge-Wheeler Gravitational Perturbation Equations”. In: *Phys. Rev. Lett.* 24.13 (Mar. 1970), pp. 737–738. DOI: [10.1103/PhysRevLett.24.737](https://doi.org/10.1103/PhysRevLett.24.737).
- [56] Oliver Rinne et al. “Implementation of higher-order absorbing boundary conditions for the Einstein equations”. In: *Classical and Quantum Gravity* 26.7, 075009 (Apr. 2009), p. 075009. DOI: [10.1088/0264-9381/26/7/075009](https://doi.org/10.1088/0264-9381/26/7/075009). arXiv: 0811.3593 [gr-qc].
- [57] Nigel T. Bishop et al. “Cauchy-characteristic extraction in numerical relativity”. In: *Phys. Rev. D* 54.10 (Nov. 1996), pp. 6153–6165. DOI: [10.1103/PhysRevD.54.6153](https://doi.org/10.1103/PhysRevD.54.6153).
- [58] Nigel T. Bishop and Luciano Rezzolla. “Extraction of gravitational waves in numerical relativity”. In: *Living Reviews in Relativity* 19.1, 2 (Dec. 2016), p. 2. DOI: [10.1007/s41114-016-0001-9](https://doi.org/10.1007/s41114-016-0001-9). arXiv: 1606.02532 [gr-qc].

- [59] Casey J. Handmer, Béla Szilágyi, and Jeffrey Winicour. “Spectral Cauchy characteristic extraction of strain, news and gravitational radiation flux”. In: *Classical and Quantum Gravity* 33.22, 225007 (Nov. 2016), p. 225007. DOI: 10.1088/0264-9381/33/22/225007. arXiv: 1605.04332 [gr-qc].
- [60] Kevin Barkett et al. “Spectral Cauchy-characteristic extraction of the gravitational wave news function”. In: *Phys. Rev. D* 102.2, 024004 (July 2020), p. 024004. DOI: 10.1103/PhysRevD.102.024004. arXiv: 1910.09677 [gr-qc].
- [61] Jordan Moxon, Mark A. Scheel, and Saul A. Teukolsky. “Improved Cauchy-characteristic evolution system for high-precision numerical relativity waveforms”. In: *Phys. Rev. D* 102.4, 044052 (Aug. 2020), p. 044052. DOI: 10.1103/PhysRevD.102.044052. arXiv: 2007.01339 [gr-qc].
- [62] Jordan Moxon, Mark A. Scheel, Saul A. Teukolsky, et al. “SpECTRE Cauchy-characteristic evolution system for rapid, precise waveform extraction”. In: *Phys. Rev. D* 107.6, 064013 (Mar. 2023), p. 064013. DOI: 10.1103/PhysRevD.107.064013.
- [63] Keefe Mitman et al. “Length dependence of waveform mismatch: a caveat on waveform accuracy”. In: *Classical and Quantum Gravity* 42.11 (June 2025), p. 117001. ISSN: 1361-6382. DOI: 10.1088/1361-6382/add8d9. URL: <http://dx.doi.org/10.1088/1361-6382/add8d9>.
- [64] Lloyd N. Trefethen. *Approximation Theory*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2019. DOI: 10.1137/1.9781611975949. URL: <https://epubs.siam.org/doi/abs/10.1137/1.9781611975949>.

Chapter 6

HIGH-ACCURACY BBH SIMULATIONS

Chapter 5 described several changes to SpEC that improved inspiral waveform differences by roughly two orders of magnitude. The goal of the present chapter is to apply those improvements to a small set of long, high-resolution binary black hole (BBH) simulations and test whether the resulting waveforms are accurate enough for next-generation gravitational-wave detectors such as LISA [1–3]. Section 6.1 lists the 12 BBH systems and explains why these parameters were chosen. Most simulations are still ongoing, but we present some promising results from those that have completed in Section 6.2.

6.1 Parameters of the BBH Systems

Results from Chapter 5 suggest that LISA-level accuracy should be possible for an equal-mass, non-spinning BBH system, but it is not obvious that the same improvements will work equally well for more complex systems. To test this, we chose a set of BBH systems that gradually increase in complexity. We also want to test these new simulations against those already present in the catalog [4] and against some new high-accuracy simulations being done by the GR-Athena++ group [5]. The simulations are also designed to be long, roughly 40 orbits, so that they are useful for surrogate-model construction and comparisons with analytic approximations.

The 12 simulations are labeled 01–12. Table 6.1 gives the initial mass ratio $q = m_A/m_B$, dimensionless spin vectors χ_A and χ_B , and initial coordinate separation D_0 in units of the total mass M . Runs 01–08 are SXS-catalog configurations repeated at higher accuracy and, in some cases, with longer inspirals. Runs 09–12 use the GR-Athena++ parameter choices but extend them to about 40 orbits.

Each system is evolved at five resolutions, levels 2–6. As defined in Section 5.3, a higher level corresponds to a smaller adaptive-mesh-refinement (AMR) tolerance; increasing the level by one reduces the AMR tolerance by a factor of 4.

At the time of writing, runs 09 and 10 have completed. For runs 01, 11, and 12, the level 6 evolutions have reached the ringdown, while the remaining simulations are still in progress. The discussion below, therefore, focuses on the completed runs, but we also present some results from runs 11 and 12.

Run ID	q	χ_A	χ_B	D_0
01	1.0	(0.0, 0.0, 0.0)	(0.0, 0.0, 0.0)	20.79
02	1.0	(0.0, 0.0, 0.753)	(0.0, 0.0, 0.088)	20.024
03	1.0	(0.0, 0.0, -0.8)	(0.0, 0.0, -0.8)	21.237
04	1.0	(0.0, 0.0, -0.8)	(0.0, 0.0, 0.8)	20.418
05	1.0	(0.0, 0.0, 0.8)	(0.0, 0.0, 0.8)	19.679
06	2.0	(0.0, 0.0, 0.0)	(0.0, 0.0, 0.0)	19.922
07	3.0	(0.0, 0.0, 0.0)	(0.0, 0.0, 0.0)	18.772
08	4.0	(0.0, 0.0, 0.0)	(0.0, 0.0, 0.0)	17.699
09	1.5	(0.0, 0.0, 0.1)	(0.0, 0.0, 0.1)	20.0
10	1.5	(0.0, 0.0, 0.4)	(0.0, 0.0, 0.4)	20.0
11	1.5	(0.0, 0.0, -0.1)	(0.0, 0.0, 0.1)	20.0
12	1.5	(0.0, 0.0, -0.4)	(0.0, 0.0, 0.4)	20.0

Table 6.1: Simulation parameters.

6.2 Results

Table 6.2 shows the CCE waveform difference and finite-radius waveform difference for run 09. The finite-radius waveform is extracted at a coordinate sphere of finite radius using the Regge–Wheeler–Zerilli procedure [6–10]. For run 09, we also use the CCE waveforms [11–16] to compute the waveform difference. These waveform differences are calculated using the full 40 orbits without any kind of time and phase optimization [4]. The waveform difference between the two highest levels (levels 5 and 6) is of the order 10^{-8} . Also, for the lower levels, the finite-radius and CCE waveform differences give approximately the same results. For the waveform difference between levels 5 and 6, CCE is an order of magnitude worse than the corresponding values using the finite-radius waveforms. We expect that the CCE waveforms will have higher waveform differences because CCE evolution has its own errors. But the lower levels in Table 6.2 show a different story, and we are looking into why this is the case.

While the waveform difference is the most relevant quantity for gravitational-wave data analysis, it only measures the similarity between the waveforms of two different levels. We should still look at more fundamental quantities like the constraint energy $\mathcal{E}_{\text{GH}}$, defined in Eq. (5.16), and convergence to make sure that we are actually evolving a general relativistic BBH system. Figures 6.1, 6.2, 6.3, and 6.4 show the constraint energy for each level over time for runs 09, 10, 11, and 12, respectively. We can see that the constraint energy is small and converges well for runs 09 and 10. For runs 11 and 12, the convergence is not as clean later in the simulation, after

Table 6.2: Run 09 waveform differences for both finite-radius and CCE waveforms, with each level compared to level 6. The finite-radius values are at $R = 300M$. Note that the CCE and finite-radius waveform differences agree quite well until we reach levels 5 and 6, where the CCE waveform difference is an order of magnitude larger. Although this is not the case for lower levels here, in general, we expect that the CCE waveform differences will be slightly larger because they go through CCE evolution, which introduces more errors.

Levels	CCE waveform diff.	Finite-radius waveform diff.
2 vs 6	4.095×10^{-3}	4.6027×10^{-3}
3 vs 6	1.081×10^{-4}	1.2155×10^{-4}
4 vs 6	1.571×10^{-5}	1.7658×10^{-5}
5 vs 6	3.278×10^{-8}	7.3180×10^{-9}

Table 6.3: Finite-radius waveform differences, with each level compared to level 6. Runs 09 and 10 have been fully completed, while runs 11 and 12 are still going through the ringdown. Therefore, we expect that the waveform difference values for runs 11 and 12 will increase when we have the full waveform. The full CCE evolution will introduce further errors, but we still expect that all these simulations will have a waveform difference of the order 10^{-8} .

Levels	Sim 09 ($R = 300M$)	Sim 10 ($R = 304M$)	Sim 11 ($R = 300M$)	Sim 12 ($R = 302M$)
2 vs 6	4.6027×10^{-3}	4.5726×10^{-3}	4.9078×10^{-4}	6.9546×10^{-4}
3 vs 6	1.2155×10^{-4}	2.1986×10^{-4}	5.4797×10^{-6}	7.9536×10^{-6}
4 vs 6	1.7658×10^{-5}	3.6244×10^{-5}	2.0738×10^{-6}	5.7015×10^{-6}
5 vs 6	7.3180×10^{-9}	3.0844×10^{-9}	3.2448×10^{-9}	1.4172×10^{-9}

around $t = 13000M$, and we are looking into possible causes for this.

Level 6 for runs 11 and 12 has not finished yet, which is why we do not show it in the graphs. Also, note that the constraint violations are quite large for level 6 during the ringdown, as shown in Figures 6.1 and 6.2. We suspect that this is due to the same ODE tolerance issues described in Section 5.5 and are working on rerunning the ringdown with a similar fix.

In Section 5.5, we showed that the waveform differences correlate strongly with differences in the black-hole trajectories. We show the same for run 09 in Figure 6.5, this time for both CCE and finite-radius waveforms. Note that Figure 6.5 has the merger and ringdown parts of the waveform, and we can see that there is still a very strong correlation with the difference in trajectories of the black holes between the two levels. This provides further evidence for the discussion in Chapter 5: if we want to reduce the waveform differences between levels, then we need to reduce the deviations in the black-hole trajectories; other things like waveform extraction,

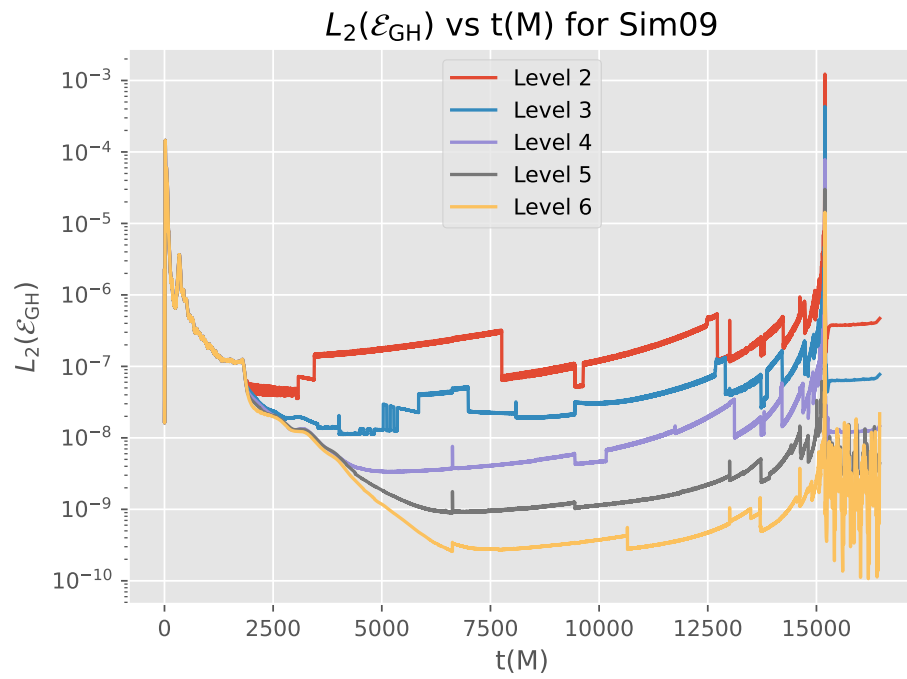


Figure 6.1: Run 09 $L_2(\mathcal{E}_{\text{GH}})$ over the simulation domain.

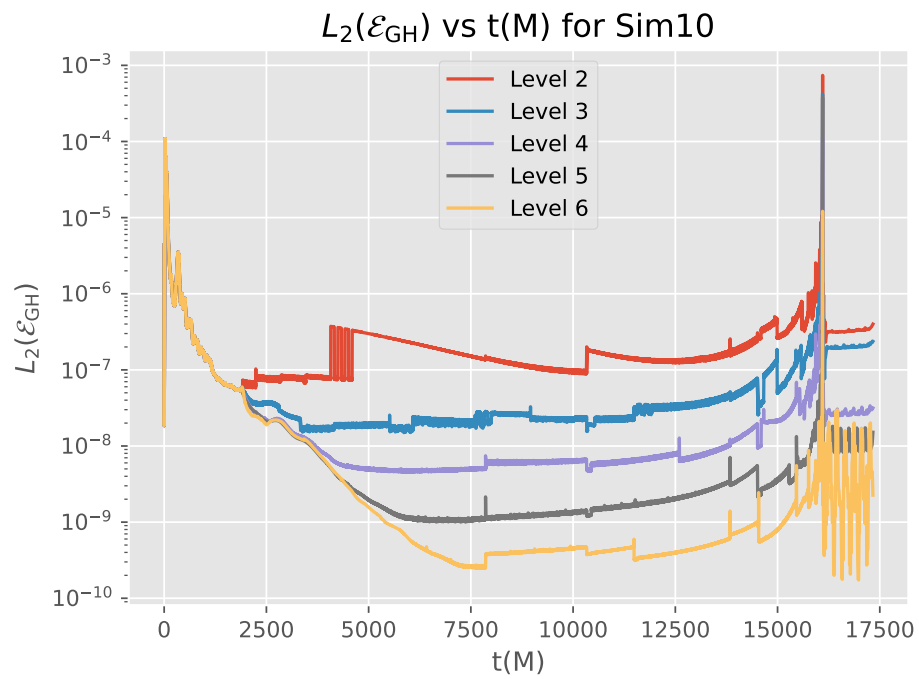


Figure 6.2: Run 10 $L_2(\mathcal{E}_{\text{GH}})$ over the simulation domain.

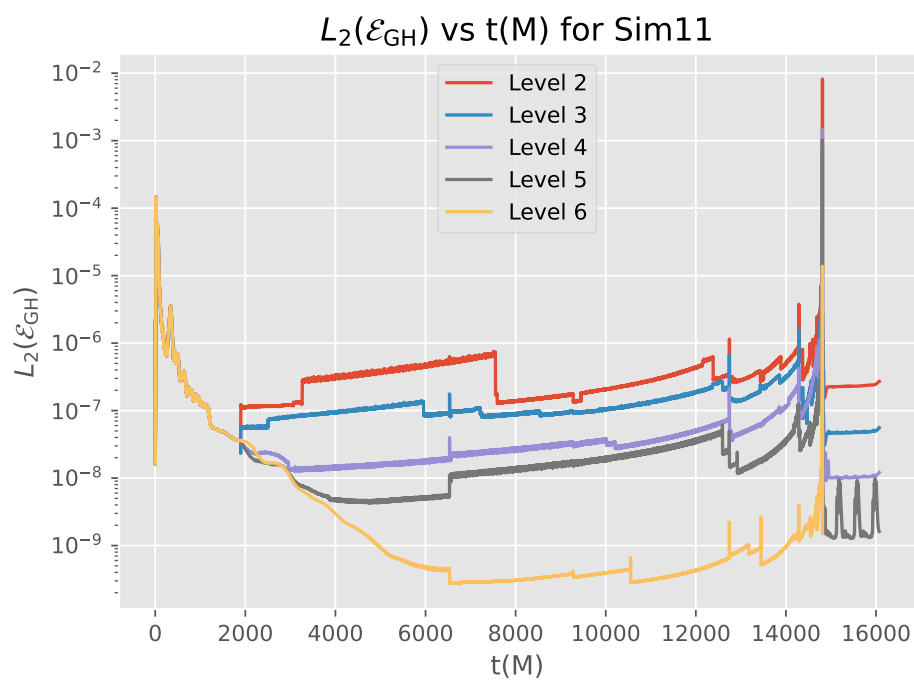


Figure 6.3: Run 11 $L_2(\mathcal{E}_{GH})$ over the simulation domain.

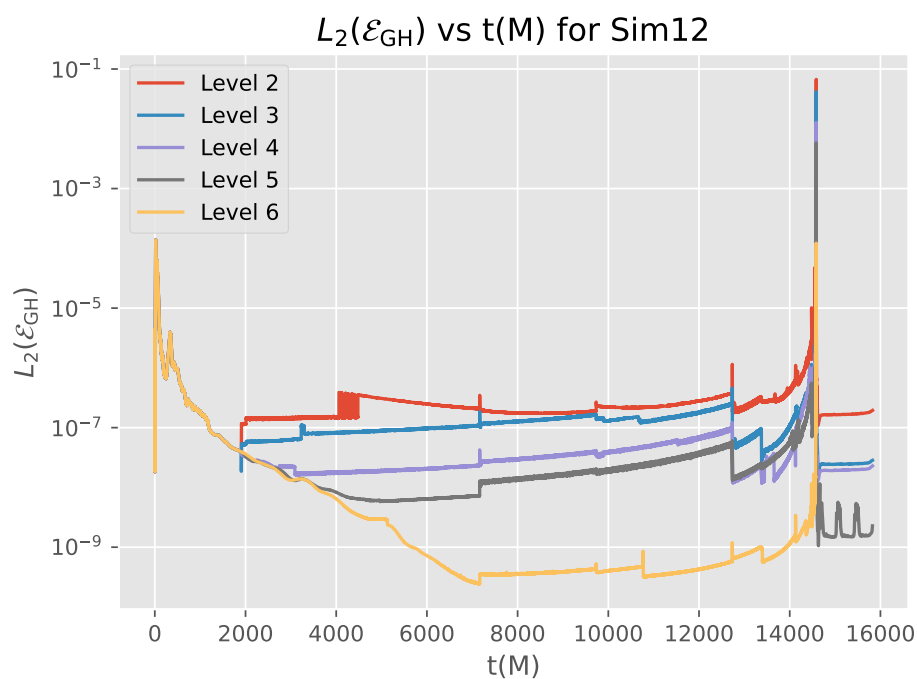


Figure 6.4: Run 12 $L_2(\mathcal{E}_{GH})$ over the simulation domain.

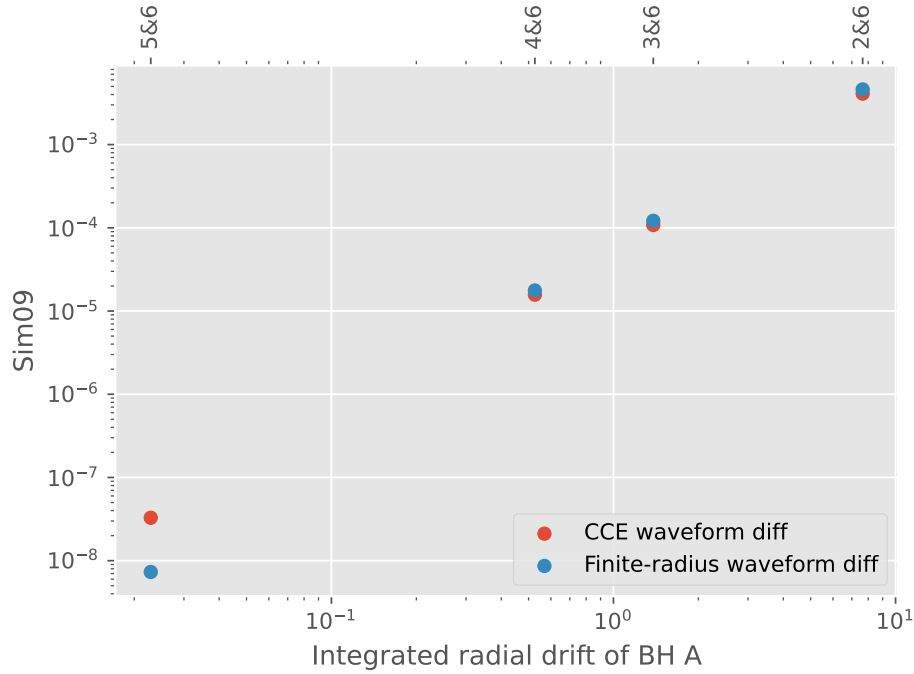


Figure 6.5: Run 09 CCE and finite-radius waveform differences versus integrated distance-from-origin difference.

constraint damping, and time-phase optimizations matter much less.

In Figure 5.11, we note that the higher modes do not correlate with the black-hole trajectories because those modes are very small for an equal-mass, non-spinning system during the early inspiral, and therefore are not resolved. We can see that when we consider unequal-mass systems with non-zero spins, those modes are resolved and we again get a very strong correlation with the black-hole trajectories, Figures 6.6, 6.7, 6.8, and 6.9.

6.3 Conclusions

The preliminary results from the simulations that have been completed are very promising. We expect that as the systems get more complex, with higher mass ratios and spins, we will not be able to achieve waveform differences of the order 10^{-8} . But even if we end up around 10^{-7} for the full 40 orbits, that should be good enough for LISA data analysis. We still have a very large parameter space that we need to cover, and these systems might require further improvements and detailed analysis. Furthermore, as we try to simulate systems with much higher mass ratios and spins, we will need to leverage the speedups offered by next-generation codes

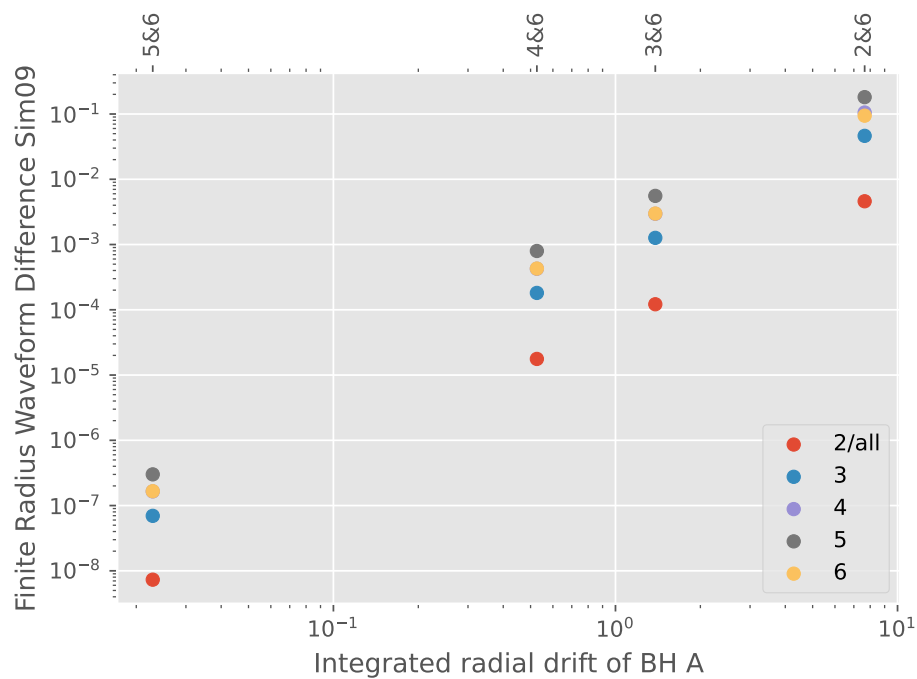


Figure 6.6: Run 09 finite-radius waveform difference versus integrated distance-from-origin difference.

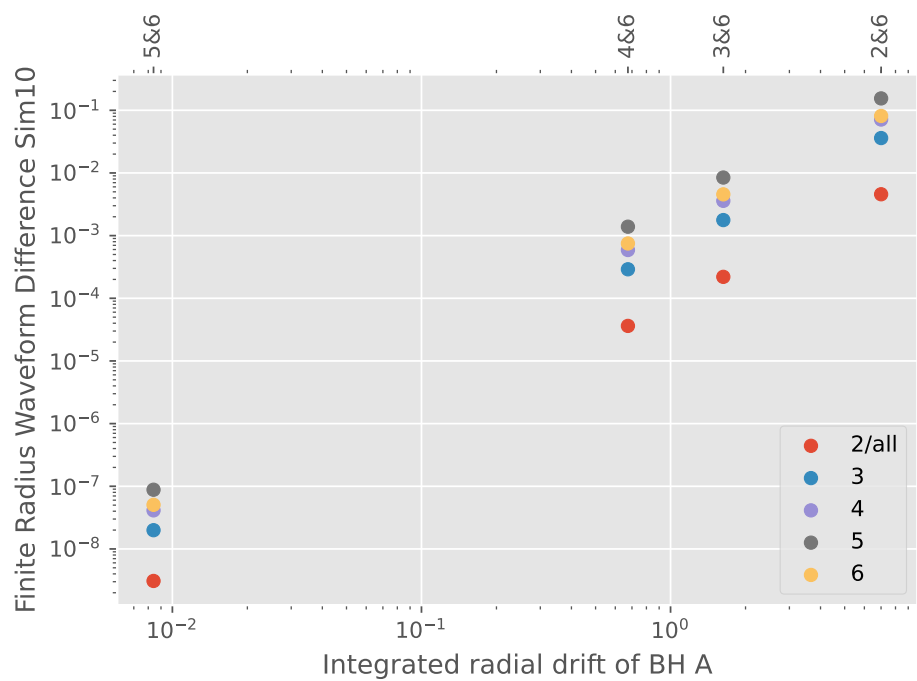


Figure 6.7: Run 10 finite-radius waveform difference versus integrated distance-from-origin difference.

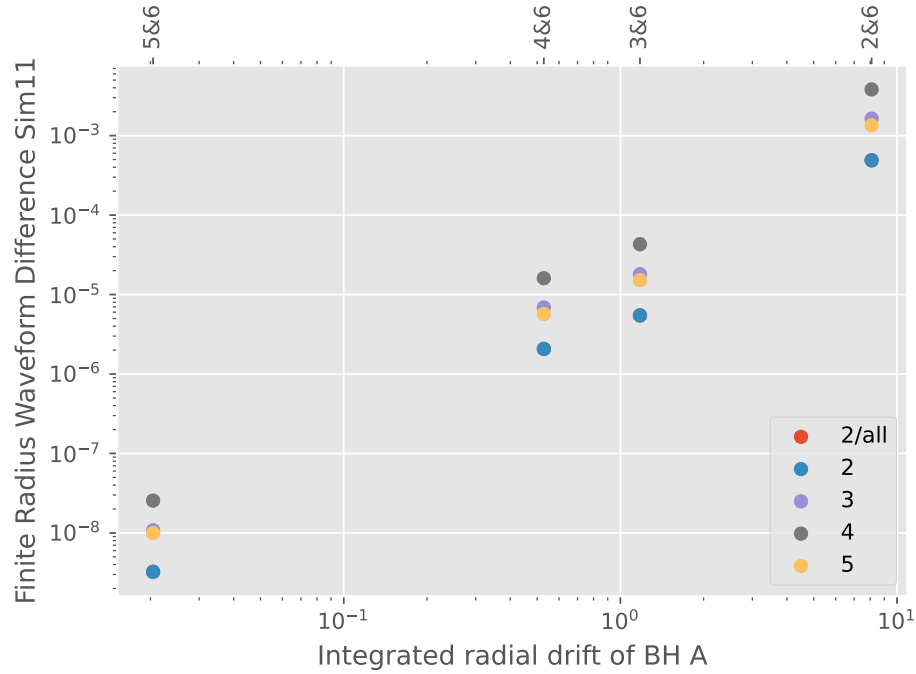


Figure 6.8: Run 11 finite-radius waveform difference versus integrated distance-from-origin difference. Note that Run 11 has not completed ringdown for level 6, so the waveform differences will be slightly lower than the final value.

like SPECTRE [17]. Having said all that, these early results are very promising and show that we might be ready with LISA-accurate waveforms by the time it launches.

References

- [1] Pau Amaro-Seoane, Heather Audley, Stanislav Babak, et al. “Laser Interferometer Space Antenna”. In: *arXiv e-prints*, arXiv:1702.00786 (Feb. 2017), arXiv:1702.00786. DOI: 10.48550/arXiv.1702.00786. arXiv: 1702.00786 [astro-ph.IM].
- [2] Michael Pürrer and Carl-Johan Haster. “Gravitational waveform accuracy requirements for future ground-based detectors”. In: *Physical Review Research* 2.2, 023151 (May 2020), p. 023151. DOI: 10.1103/PhysRevResearch.2.023151. arXiv: 1912.10055 [gr-qc].
- [3] LISA Consortium Waveform Working Group, Niayesh Afshordi, Sarp Akçay, et al. “Waveform modelling for the Laser Interferometer Space Antenna”. In: *Living Reviews in Relativity* 28.1, 9 (Oct. 2025), p. 9. DOI: 10.1007/s41114-025-00056-1. arXiv: 2311.01300 [gr-qc].
- [4] Mark A. Scheel et al. “The SXS collaboration’s third catalog of binary black hole simulations”. In: *Classical and Quantum Gravity* 42.19, 195017 (Oct.

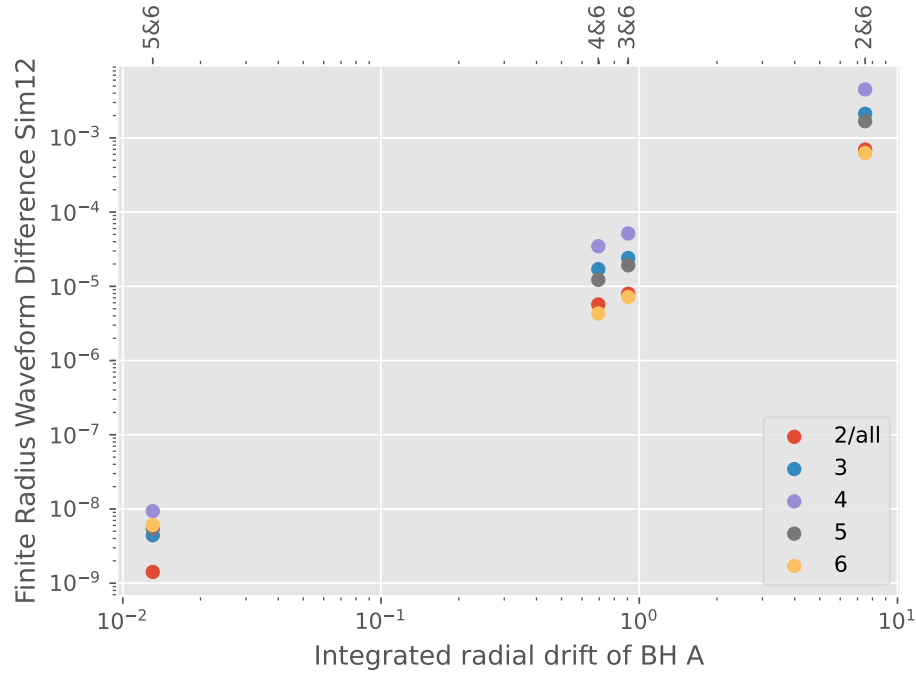


Figure 6.9: Run 12 finite-radius waveform difference versus integrated distance-from-origin difference. Note that Run 12 has not completed ringdown for level 6, so the waveform differences will be slightly lower than the final value.

2025), p. 195017. DOI: [10.1088/1361-6382/adfd34](https://doi.org/10.1088/1361-6382/adfd34). arXiv: 2505.13378 [gr-qc].

- [5] Estuti Shukla et al. “GR-Athena++ simulations of spinning binary black hole mergers”. In: *Classical and Quantum Gravity* 43.6, 065019 (Mar. 2026), p. 065019. DOI: [10.1088/1361-6382/ae4da4](https://doi.org/10.1088/1361-6382/ae4da4). arXiv: 2510.13963 [gr-qc].
- [6] Michael Boyle, Daniel Hemberger, Dante A. B. Iozzo, et al. “The SXS collaboration catalog of binary black hole simulations”. In: *Classical and Quantum Gravity* 36.19, 195006 (Oct. 2019), p. 195006. DOI: [10.1088/1361-6382/ab34e2](https://doi.org/10.1088/1361-6382/ab34e2). arXiv: 1904.04831 [gr-qc].
- [7] Olivier Sarbach and Manuel Tiglio. “Gauge-invariant perturbations of Schwarzschild black holes in horizon-penetrating coordinates”. In: *Phys. Rev. D* 64.8 (Oct. 2001), p. 084016. DOI: [10.1103/PhysRevD.64.084016](https://doi.org/10.1103/PhysRevD.64.084016). arXiv: gr-qc/0104061 [gr-qc].
- [8] Tullio Regge and John A. Wheeler. “Stability of a Schwarzschild Singularity”. In: *Physical Review* 108.4 (Nov. 1957), pp. 1063–1069. DOI: [10.1103/PhysRev.108.1063](https://doi.org/10.1103/PhysRev.108.1063).

- [9] Frank J. Zerilli. “Effective Potential for Even-Parity Regge-Wheeler Gravitational Perturbation Equations”. In: *Phys. Rev. Lett.* 24.13 (Mar. 1970), pp. 737–738. doi: 10.1103/PhysRevLett.24.737.
- [10] Oliver Rinne et al. “Implementation of higher-order absorbing boundary conditions for the Einstein equations”. In: *Classical and Quantum Gravity* 26.7, 075009 (Apr. 2009), p. 075009. doi: 10.1088/0264-9381/26/7/075009. arXiv: 0811.3593 [gr-qc].
- [11] Nigel T. Bishop et al. “Cauchy-characteristic extraction in numerical relativity”. In: *Phys. Rev. D* 54.10 (Nov. 1996), pp. 6153–6165. doi: 10.1103/PhysRevD.54.6153.
- [12] Nigel T. Bishop and Luciano Rezzolla. “Extraction of gravitational waves in numerical relativity”. In: *Living Reviews in Relativity* 19.1, 2 (Dec. 2016), p. 2. doi: 10.1007/s41114-016-0001-9. arXiv: 1606.02532 [gr-qc].
- [13] Casey J. Handmer, Béla Szilágyi, and Jeffrey Winicour. “Spectral Cauchy characteristic extraction of strain, news and gravitational radiation flux”. In: *Classical and Quantum Gravity* 33.22, 225007 (Nov. 2016), p. 225007. doi: 10.1088/0264-9381/33/22/225007. arXiv: 1605.04332 [gr-qc].
- [14] Kevin Barkett et al. “Spectral Cauchy-characteristic extraction of the gravitational wave news function”. In: *Phys. Rev. D* 102.2, 024004 (July 2020), p. 024004. doi: 10.1103/PhysRevD.102.024004. arXiv: 1910.09677 [gr-qc].
- [15] Jordan Moxon, Mark A. Scheel, and Saul A. Teukolsky. “Improved Cauchy-characteristic evolution system for high-precision numerical relativity waveforms”. In: *Phys. Rev. D* 102.4, 044052 (Aug. 2020), p. 044052. doi: 10.1103/PhysRevD.102.044052. arXiv: 2007.01339 [gr-qc].
- [16] Jordan Moxon, Mark A. Scheel, Saul A. Teukolsky, et al. “SpECTRE Cauchy-characteristic evolution system for rapid, precise waveform extraction”. In: *Phys. Rev. D* 107.6, 064013 (Mar. 2023), p. 064013. doi: 10.1103/PhysRevD.107.064013.
- [17] Nils Deppe et al. *SpECTRE*. Version 2026.04.01. Apr. 2026. doi: 10.5281/zenodo.19373346. URL: <https://doi.org/10.5281/zenodo.19373346>.

