Control of Black Hole Parameters for Binary Evolutions

Caltech SURF First Interim Report

Iago Mendes*

July 9, 2024

I. Introduction

In the early twentieth century, Einstein revolutionized the study of gravity by connecting spacetime geometry with physical dynamics. As John Wheeler says, "Spacetime tells matter how to move; matter tells spacetime how to curve" [1]. Being a highly complex theory, many problems of interest only have analytic solutions in special cases with symmetry. In this context, Numerical Relativity emerged as an essential field to solve these problems numerically, allowing us to explore general cases that can be found in the universe. Specifically, simulations of Binary Black Holes (BBH) became very important as gravitational wave detectors were developed, needing to use numerical results to identify and characterize signals in their data [2].

Famously, the Einstein equations relate the curvature of spacetime to the stress-energy of matter, forming a system of ten nonlinear partial differential equations (PDEs). With the 3+1 formalism, we can rearrange these equations so that spacetime is described by spacelike three-dimensional slices of constant time [3]. In doing so, we find that four out of the ten equations do not involve time derivatives, implying that they are constraints that must be satisfied at all times. The remaining six equations describe an evolution of the constraint-satisfying fields. Using this formalism, the Spectral Einstein Code (SpEC) [4] runs BBH simulations by first finding initial data and then running an evolution on them. Over time, as SpEC faced more challenging BBH with high mass ratios and spins, several improvements had to be made to the initial data techniques, which are summarized in [5].

Despite its success in BBH simulations, SpEC shows its limitations in more challenging problems, such as binary neutron star mergers and BBH with extreme configurations. In this context, SpECTRE [6] was created as a codebase that follows a better parallelism model and aims to be more scalable [7]. Previous work has already shown that SpECTRE can be faster and more accurate than SpEC when performing similar tasks due to its use of parallelism [8]. This will be especially needed for the upcoming gravitational wave detectors with higher sensitivity, such as the Cosmic Explorer, the Einstein Telescope and LISA.

As part of an effort to allow researchers to fully simulate BBH in SpECTRE, an initial data procedure similar to the one in SpEC needs to be completed. This is greatly benefitted by a scalable elliptic solver that was recently developed [8], which can now be used to solve the initial data equations. Be that as it may, before the start of the SURF program, SpECTRE did not have a way to enforce specific masses and spins for the black holes or to avoid drifts in the orbital trajectory. As described in the next section, this is the problem that this research project aims to address.

---

*imendes@caltech.edu, ibrazmen@oberlin.edu

II. Numerical Method

To find initial data, `SpEC` uses the extended conformal thin-sandwich (XCTS) decomposition [5], which transforms the constraint PDEs into a system of five elliptic PDEs. Before solving the XCTS equations, we need to specify the conformal spatial metric $\bar{\gamma}_{ij}$, the extrinsic curvature trace K, and their respective time derivatives $\partial_t \bar{\gamma}_{ij}$ and $\partial_t K$ [9]. There are many methods for specifying these quantities, but the one that has shown to be the most promising for BBH simulations is the superposed Kerr-Schild (SKS) approach [10]. It enforces quasiequilibrium conditions by setting $\partial_t \bar{\gamma}_{ij} = 0$ and $\partial_t K = 0$. Additionally, as the name suggests, it specifies $\bar{\gamma}_{ij}$ and K by superposing two analytic solutions of Kerr-Schild black holes with masses M_A^{Kerr} and M_B^{Kerr} and with dimensionless spins $\vec{\chi}_A^{\text{Kerr}}$ and $\vec{\chi}_B^{\text{Kerr}}$.

When constructing the SKS analytical expressions for $\bar{\gamma}_{ij}$ and K, we also need to choose the centers of the black holes $\vec{c}_A$ and $\vec{c}_B$. When setting up a simulation, it is more convenient to specify their relative distance $\vec{D} = \vec{c}_A - \vec{c}_B$. Because of this, for any choice of $\vec{c}_A$, we have $\vec{c}_B = \vec{c}_A + \vec{D}$. In other words, we only have to choose $\vec{c}_A$.

Having $\bar{\gamma}_{ij}$, K, $\partial_t \bar{\gamma}_{ij}$ and $\partial_t K$ specified, we can use the elliptic solver on the XCTS equations, three of which will be solved for the shift β^i. Similar to any elliptic PDE problem, we have to set boundary conditions before solving these equations. In the boundary conditions of $\vec{\beta}$, we can add a constant velocity $\vec{v}_0$, giving us more control over the initial kinematics of the binary system.

Once the XCTS equations are solved, we have all the information that we need about the zero-time slice of spacetime. With this, we can use an apparent horizon finder to get measurements of the black holes in the constructed initial data. Let M_A, M_B, $\vec{\chi}_A$ and $\vec{\chi}_B$ be the masses and spins of the actual black holes that we have been able to create. In general, these values will differ from the "target" quantities M_A^*, M_B^*, $\vec{\chi}_A^*$ and $\vec{\chi}_B^*$. For example, suppose that we wish to simulate an equal-mass non-spinning case with $M_A^* = M_B^* = 0.5$ and $\vec{\chi}_A^* = \vec{\chi}_B^* = (0,0,0)$. In order to construct the SKS analytical expressions, it is natural to specify $M_{A,B}^{\text{Kerr}} = M_{A,B}^*$ and $\vec{\chi}_{A,B}^{\text{Kerr}} = \vec{\chi}_{A,B}^*$. With this, we can solve the XCTS equations and find horizons in the resulting initial data. Once we do it in `SpECTRE`, we find that the resulting black hole parameters are actually $M_{A,B} \approx 0.6$ and $\vec{\chi}_{A,B} \approx (0, 0, -0.002)$.

If we measure other parameters of the binary, we might find that they are also not desirable. The center of mass $\vec{C}$ can be found by evaluating the integral

$$\vec{C} = \frac{3}{8\pi M_{\text{ADM}}} \oint_{S_\infty} \psi^4 \, d\vec{S}, \tag{1}$$

[5], where ψ is the conformal factor and S_∞ is a surface at infinite radius over which we evaluate the surface integral. M_{ADM} is the ADM mass, defined as

$$M_{\text{ADM}} = \frac{1}{16\pi} \int_{S_\infty} \sqrt{\gamma} \gamma^{jn} \gamma^{im} (\partial_j \gamma_{mn} - \partial_m \gamma_{jn}) \, dS_j \tag{2}$$

[9], where γ_{ij} is the spatial metric, γ^{ij} is the inverse spatial metric, and γ is the determinant of γ_{ij}. We can also use the ADM formalism to measure the total linear momentum $\vec{P}_{\text{ADM}}$ as

$$P_{\text{ADM}}^i = \frac{1}{8\pi} \oint_{S_\infty} (K^{ij} - K\gamma^{ij}) \, dS_j \tag{3}$$

[5], where K^{ij} is the inverse extrinsic curvature. Ideally, $\vec{C}$ and $\vec{P}_{\text{ADM}}$ should be equal to zero in order to minimize any drifts of the orbit, especially for long simulations. However, similar to the example above, we can only know these quantities after the initial data has been constructed.

$$\text{if not } \left(M_{A,B}, \vec{\chi}_{A,B}, \vec{C}, \vec{P}_{\text{ADM}}\right) \approx \left(M_{A,B}^*, \vec{\chi}_{A,B}^*, 0, 0\right)$$
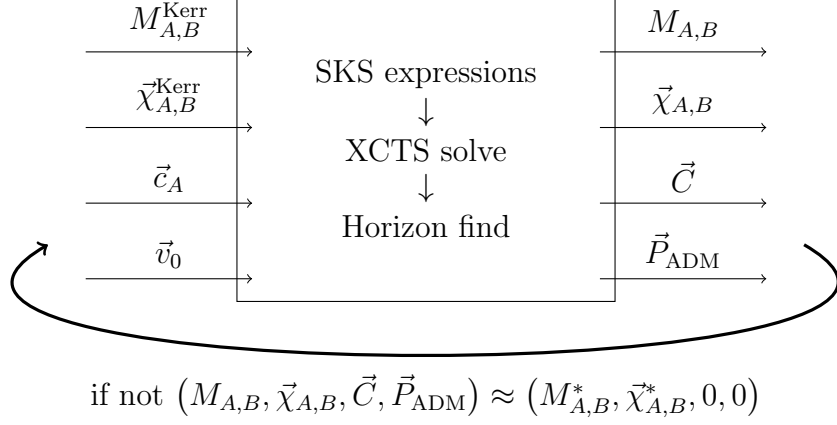
Figure 1: Schematic representation of the control loop.

Note that $M_{A,B}$, $\vec{\chi}_{A,B}$, $\vec{C}$ and $\vec{P}_{\text{ADM}}$ are the parameters that we wish to control, but we cannot directly set them to the desired values. The way that `SpEC` handles this issue is by iterating over different choices of the free data ($M_{A,B}^{\text{Kerr}}$, $\vec{\chi}_{A,B}^{\text{Kerr}}$, $\vec{c}_A$ and $\vec{v}_0$), trying to find the ones that result in the ideal physical parameters. The goal of this project is to reproduce this iterative scheme in `SpECTRE`. Figure 1 shows a schematic representation of this "control loop".

Let the choices of $M_{A,B}^{\text{Kerr}}$ and $\vec{\chi}_{A,B}^{\text{Kerr}}$ be represented as

$$\mathbf{u} = \left(M_A^{\text{Kerr}}, M_B^{\text{Kerr}}, \chi_A^{\text{Kerr},x}, \chi_A^{\text{Kerr},y}, \chi_A^{\text{Kerr},z}, \chi_B^{\text{Kerr},x}, \chi_B^{\text{Kerr},y}, \chi_B^{\text{Kerr},z}\right). \tag{4}$$

Also, let the difference between the current and target physical parameters be represented as

$$\begin{aligned}\mathbf{F}(\mathbf{u}) = \Big(&M_A - M_A^*, M_B - M_B^*, \chi_A^x - \chi_A^{*,x}, \chi_A^y - \chi_A^{*,y}, \\ &\chi_A^z - \chi_A^{*,z}, \chi_B^x - \chi_B^{*,x}, \chi_B^y - \chi_B^{*,y}, \chi_B^z - \chi_B^{*,z}\Big).\end{aligned} \tag{5}$$

As discussed in the example above, it is natural to use the target parameters as an initial guess for the Kerr masses and spins:

$$\mathbf{u}_0 = \left(M_A^*, M_B^*, \chi_A^{*,x}, \chi_A^{*,y}, \chi_A^{*,z}, \chi_B^{*,x}, \chi_B^{*,y}, \chi_B^{*,z}\right). \tag{6}$$

If we also assume that $M_{A,B}|_0 \approx M_{A,B}^{\text{Kerr}}|_0$ and $\vec{\chi}_{A,B}|_0 \approx \vec{\chi}_{A,B}^{\text{Kerr}}|_0$, then a good initial approximation for the Jacobian of $\mathbf{F}(\mathbf{u})$ is $\mathbb{J}_0 \approx \mathbb{I}$ (a 8x8 identity matrix). With this, we can update our free data at every control iteration $k \geq 1$ using a Newton-Raphson scheme [11]:

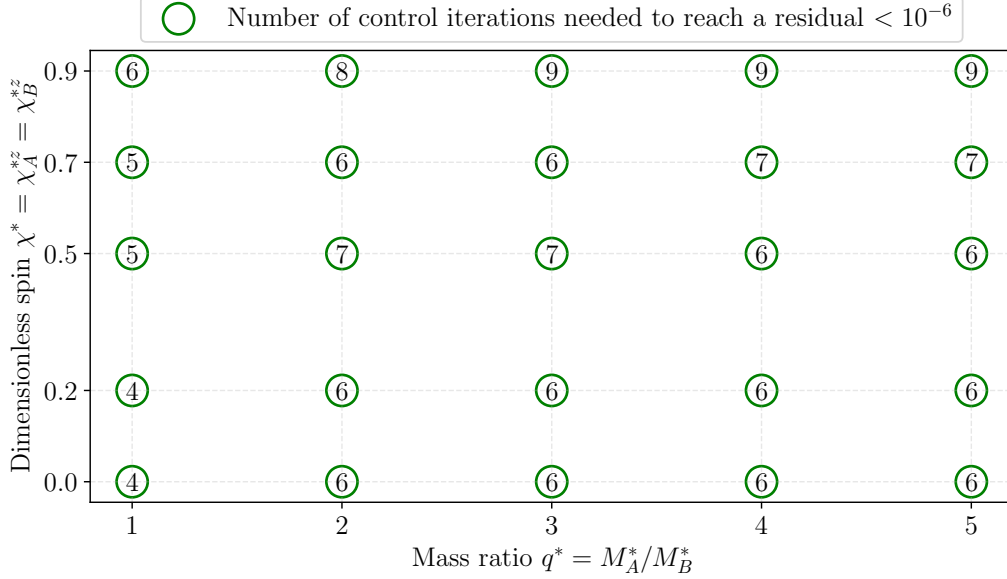$$\mathbf{u}_k = \mathbf{u}_{k-1} - \mathbb{J}_{k-1}^{-1} \cdot \mathbf{F}_{k-1}. \tag{7}$$

Then, we can update the Jacobian with Broyden's method [11]:

$$\mathbb{J}_k = \mathbb{J}_{k-1} + \frac{\mathbf{F}_k \otimes \mathbf{\Delta u}_k}{||\mathbf{\Delta u}_k||^2}, \tag{8}$$

where we use $\Delta x_k = x_k - x_{k-1}$ for any variable x from now on.

We can also come up with iterative schemes for the choices of $\vec{c}_A$ and $\vec{v}_0$. To drive the center of mass $\vec{C}$ to zero we can update $\vec{c}_A$ as

$$\vec{c}_{A,k} = \vec{c}_{A,k-1} - \vec{C}_{k-1} - \frac{M_{A,k-1}\Delta M_{B,k-1} - M_{B,k-1}\Delta M_{A,k-1}}{(M_{A,k-1} + M_{B,k-1})^2}\vec{D} \tag{9}$$

Figure 2: Tests of the current control loop over part of the parameter space of masses and spins.

[5]. Considering a small perturbation of $\vec{v}_0$, we can drive the linear momentum $\vec{P}_{\mathrm{ADM}}$ to zero with

$$\vec{v}_{0,k+1} = \vec{v}_{0,k-1} - \frac{\vec{P}_{\mathrm{ADM},k-1}}{M_{k-1}} + \Delta M_{k-1}(\vec{v}_{0,k-1} + \vec{\Omega}_0 \times \vec{c}_{A,k-1})$$
$$- \vec{\Omega}_0 \times \delta\vec{c}_{A,k-1} - \frac{\Delta M_{B,k-1}}{M_{k-1}}\vec{\Omega}_0 \times \vec{D} \tag{10}$$

[5], where $M_k = M_{A,k} + M_{B,k}$, $\vec{\Omega}_0$ is a user-specified initial orbit angle, and $\delta\vec{c}_{A,k}$ is a small perturbation of $\vec{c}_{A,k}$.

## III. Progress Update

As an initial step, my co-mentor Nils Vu[†] and I decided to implement the control of masses and spins as outlined in equations (7) and (8). This was a good way to get used with `SpECTRE` development without diving deep into its more complicated infrastructure as this was implemented in the `Python` files of the codebase. We decided to differ from the approach in [5] in some aspects, such as using $M_{A,B}^{\mathrm{Kerr}}$ and $\vec{\chi}_{A,B}^{\mathrm{Kerr}}$ instead of the black holes radii and angular frequencies. This implementation was merged into `SpECTRE`'s development branch within my first week. Since then, I have heard positive feedback from other researchers in our group who have used this feature. Figure 2 shows some initial data configurations that were used to test this control loop.

Next, I got started with the `C++` infrastructure of `SpECTRE`. With the goal of implementing a measure of the ADM total linear momentum as described in equation (3), I created pointwise functions to compute the integrands of such computation given the available tensors. These functions were tested with an analytic solution of a single boosted Kerr-Schild black hole. With a boost speed of v, we expect a linear momentum magnitude equal to

$$P_{\mathrm{expected}} = \gamma M v, \tag{11}$$

[†]nilsvu@caltech.edu
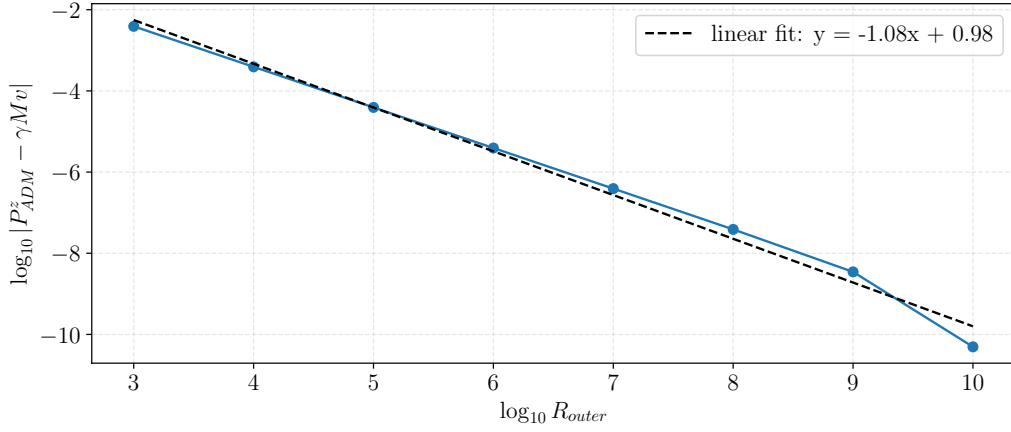
Figure 3: Converge test of the ADM total linear momentum computed as a surface integral.

where $\gamma = 1/\sqrt{1-v^2}$ is the Lorentz factor and M is the mass of the black hole. Figure 3 shows a convergence of the difference between the measured and expected linear momenta as we increase the outer boundary radius R_{outer}, over which we perform the surface integral in equation (3). From the convergence test, we see that this error decays as $1/R_{\text{outer}}$, as expected.

After that, I started working on computing the linear momentum surface integral as part of the executable that solves the XCTS equations. This required me to learn some details of the SpECTRE infrastructure for data boxes and parallelism. I am currently developing this feature, which should be merged within a week at most.

IV. Future Work

As an immediate next step, I plan on implementing the computation of the center of mass as described in equation (1), which will also require me to compute the ADM mass as in equation (2). Both of these will involve surface integrals similar to the one used in equation (3), so their implementation should benefit from my previous work. Once this is done, SpECTRE will be able to compute $\vec{C}$ and $\vec{P}_{\text{ADM}}$, allowing me to use equations (9) and (10) to control these quantities.

As described in [5], it is possible to use Gauss' law to split the surface integrals at infinity found in equations (1)-(3) into a surface integral at a smaller radius plus a volume integral. While testing the computation of $\vec{P}_{\text{ADM}}$, I briefly tried this approach, but the results were worse than evaluating the surface integral at infinity. However, this method has shown to be more numerically accurate in SpEC [5], so it is important for us to further experiment with it.

Additionally, improvements to the control loop scheme should be made. A quick one is to re-use the previous XCTS solutions as an initial guess for the following XCTS solve. This should not reduce the number of control iterations, but it will drastically decrease the time that it takes for each iteration to run. Apart from that, we must try to test extreme configurations (high mass ratios and spins), which could potentially require adding optimizations to our current method, such as line search.

Finally, we have been discussing novel improvements that we can make to account for the effects of junk radiation in the BBH physical parameters. Junk radiation is a collection of irregularities in the early evolution of numerical relativity simulations, which happens due to not accurately representing the past of related fields. Usually, researchers wait until junk radiation has stopped affecting the simulation and only consider the results thereafter. At that point, the physical parameters might have substantially deviated from the desired values due

to the energy exchanges with such radiation. This makes it especially hard to get more extreme spins.

There have been previous attempts at predicting the effects of junk radiation on the BBH masses and spins [12], using statistics to choose initial values that might result in the desired values after junk radiation has relaxed. Given the infrastructure that we have implemented for the initial data control loop, it would be relatively straightforward to attempt something similar. We are considering using a data-driven machine learning model, which could potentially perform better than a statistical model. We have also considered ways to mitigate junk radiation using perturbation theory, but this is a more uncertain idea.

References

[1] John Wheeler and Kenneth Ford. "Geons, Black Holes and Quantum Foam: A Life in Physics". In: *American Journal of Physics* 68 (June 2000). DOI: 10.1119/1.19497.

[2] B. P. Abbott et al. "Observation of Gravitational Waves from a Binary Black Hole Merger". In: *Phys. Rev. Lett.* 116 (6 Feb. 2016), p. 061102. DOI: 10.1103/PhysRevLett.116.061102.

[3] Miguel Alcubierre. *Introduction to 3+1 Numerical Relativity*. Oxford University Press, Apr. 2008. ISBN: 9780199205677. DOI: 10.1093/acprof:oso/9780199205677.001.0001.

[4] Lawrence Kidder, Harald Pfeiffer, Mark Scheel, et al. *Spectral Einstein Code (SpEC)*. URL: https://black-holes.org/code/SpEC.

[5] Serguei Ossokine, Francois Foucart, Harald P Pfeiffer, et al. "Improvements to the construction of binary black hole initial data". In: *Classical and Quantum Gravity* 32.24 (Dec. 2015), p. 245010. ISSN: 1361-6382. DOI: 10.1088/0264-9381/32/24/245010.

[6] Nils Deppe, William Throwe, Lawrence E. Kidder, et al. *SpECTRE*. Version 2024.02.05. Feb. 2024. DOI: 10.5281/zenodo.10619885. URL: https://spectre-code.org.

[7] Lawrence E. Kidder, Scott E. Field, Francois Foucart, et al. "SpECTRE: A task-based discontinuous Galerkin code for relativistic astrophysics". In: *Journal of Computational Physics* 335 (2017), pp. 84–114. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2016.12.059.

[8] Nils L. Vu, Harald P. Pfeiffer, Gabriel S. Bonilla, et al. "A scalable elliptic solver with task-based parallelism for the SpECTRE numerical relativity code". In: *Phys. Rev. D* 105 (8 Apr. 2022), p. 084027. DOI: 10.1103/PhysRevD.105.084027.

[9] Thomas W. Baumgarte and Stuart L. Shapiro. *Numerical Relativity: Solving Einstein's Equations on the Computer*. Cambridge University Press, 2010. ISBN: 9781139193344. DOI: 10.1017/CBO9781139193344.

[10] Geoffrey Lovelace et al. "Binary-black-hole initial data with nearly extremal spins". In: *Phys. Rev. D* 78 (8 Oct. 2008), p. 084017. DOI: 10.1103/PhysRevD.78.084017.

[11] William H. Press et al. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, 2007. ISBN: 0521880688. URL: https://numerical.recipes.

[12] Kenny Higginbotham et al. "Coping with spurious radiation in binary black hole simulations". In: *Phys. Rev. D* 100 (8 Oct. 2019), p. 081501. DOI: 10.1103/PhysRevD.100.081501.