# Numerical Simulations of Black Hole Binaries
# LIGO SURF 2015

Lindsey Whitesides
Mentor: Mark Scheel[1]
[1] *Theoretical Astrophysics 350-17, California Institute of Technology, Pasadena, CA 91125, USA*
(Dated: November 1, 2015)

The inspiral and merger of binary black-hole systems produce gravitational waves which Advanced LIGO aims to detect. Numerical simulations of these binaries are used to compute waveform templates that help LIGO pull signals out of noise and determine binary parameters such as masses and spins. We are conducting simulations of binary black-hole mergers with high spins of 0.91 and 0.99 oriented in a "superkick" configuration, in which linear momentum is radiated anisotropically and the remnant black hole acquires a velocity of up to thousands of km/s. The final velocity depends sensitively on the initial spin orientations. Once our simulations complete, we will determine whether LIGO can distinguish between initial spin orientations for these high-spin systems; it cannot for low-spin systems. We have also developed new code which measures the eccentricity of the orbit and computes new orbital parameters. It is written in C++, and includes error bars on computed quantities. Several simulations are run iteratively, adjusting the orbital parameters each time until the target eccentricity is reached. Our new code will run during the main simulation instead of afterwards, allowing each iteration to end earlier, and thus increasing the speed of these simulations.

## INTRODUCTION

A binary black-hole system loses energy in the form of gravitational waves. This causes the two black holes to inspiral towards each other, until finally, they coalesce into a single black hole. Little is known about the final stages of these black hole mergers from observation because black holes do not emit light. However, if we can analyze the emitted gravitational waves we will be able to build a picture of the inner workings of black holes.

The first science run of Advanced LIGO has just started, so LIGO should regularly detect gravitational waves coming from some of the most fascinating events in the universe. By observing and analyzing gravitational waves, LIGO hopes to study events such as supernovae explosions and black hole mergers. However, in order to pull such a weak gravitational signals out of the noise in the data, LIGO needs expected waveform templates for possible events that it might observe. It is now possible to numerically solve Einstein's equations of general relativity for the merger of binary black holes and retrieve the resulting gravitational waves. For maximum efficiency in data analysis, independent templates are not needed for independent waveforms that LIGO cannot distinguish from one another.

If the merging black holes have opposing spin orientations, then upon merger, a significant amount of linear momentum is carried off by gravitational waves. Momentum must be conserved, so the resulting black hole will gain momentum that may be large enough for it to be kicked out of its host galaxy [1, 2]. This strange phenomena is called a superkick. Extremely high spin superkick simulations have not been attempted yet, and we are interested in them because such high energy black hole mergers may behave differently than we expect.

Ultimately, we want to know whether LIGO can detect the direction of the spins at merger. The magnitude and direction of a superkick are known to depend sensitively on how the spins are oriented at the moment of coalescence [3], yet it is unknown whether the gravitational waveforms produced by these mergers are affected by the spin orientation enough for LIGO to be able to detect. From previous work, we know that LIGO is not sensitive to the spin directions at the time of merger for low spins up to 0.5 relative to maximum. However, the superkick itself, and thus the effect on the waveform, should be much larger for higher spins so LIGO may be able to detect it.

Time limitations are a major challenge in numerical relativity: these simulations take weeks to months. Thus, it is necessary to make educated guesses for initial parameters of the black holes orbit so that we can start the simulation at the more interesting orbits closer to merger. Eccentricity of orbit is one such parameter of interest. Isolated binary black hole systems radiate gravitational waves as they orbit, causing a reduction of eccentricity over time until the orbit is essentially circular [4, 5]. Most of the black hole binaries that LIGO will observe are expected to be of this type, so in our simulations the eccentricity produced by the initial conditions must be sufficiently close to zero before we can proceed to completely evolve the merger.

This project was split into two distinct parts. The main, overall focus was on conducting high-spin superkick simulations. While the simulations were running, I improved the eccentricity reduction process.

I ran high spin black-hole superkick simulations for two different spin magnitudes. One simulation has a relative spin of 0.91, where 1 is the maximum allowed by the theory of general relativity. This spin is quite high, but remains safely within the range that the supercomputer is capable of handling. The other has a spin of 0.99 which pushes the boundary of what our current code is capable of handling. The spin 0.91 simulation is nearing completion of the eccentricity-reduction phase, where the initial guesses for the orbital parameters, angular and radial velocity, are iteratively updated until the resulting eccentricity is close enough to zero. The initial angular and radial velocities chosen for our spin 0.99 simulation produced a highly eccentric orbit and the black holes merged incredibly quickly, as a result of extreme precession. We have created a spin 0.99 run with better guesses for the orbital parameters, resulting in a much more circular orbit, which is more useful. It is currently beginning the eccentricity reduction process.

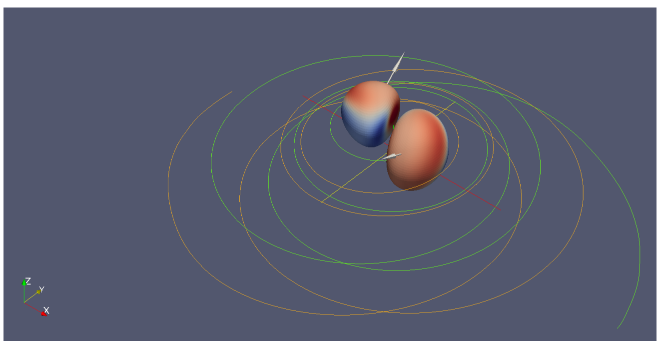

FIG. 1. Visual animation of the two black holes in the Spin 0.99 simulation at merger. The trajectories of the black holes are shown by the lines and the spin directions are indicated by the arrows. The surfaces are the apparent horizons, and the colors indicate the vorticity [6], a measure of spin.

As the simulations ran, I improved the eccentricity measurement code. The code was rewritten in C++ and in a more efficient, organized way. This code calculates the eccentricity of the orbit and updated guesses for what the orbital parameters should be to reduce the eccentricity. Our new code also calculates error bars for the eccentricity and the updated orbital parameters, and it allows us to impose constraints on the eccentricity-fit parameters. We are in the final stages of implementing this eccentricity code into the main simulation code. Measurements of the eccentricity and associated uncertainties will be reported as the run progresses, rather than being computed after the run has completed. This will significantly improve the time efficiency of our numerical simulations.

## METHODS AND RESULTS

### Spin 0.91 Simulation

Our 0.91 magnitude spin simulation is going through the eccentricity reduction phase. After a set time, the simulation stops, the eccentricity is calculated and the initial orbital parameter guesses are updated. Then the simulation is repeated with these updated orbital parameters, in order to obtain an eccentricity closer to zero, which resembles a natural binary black hole merger. When the eccentricity does not decrease after a change in orbital parameters, the resolution is increased to produce better results. Higher resolution simulations take longer to run, so we start at lower resolutions, but they become necessary when we need finer adjustments. We started this run at the lowest resolution setting, Level 0, and re-ran the simulation on higher resolution settings. It is currently on Level 4, but has not made extremely significant progress.

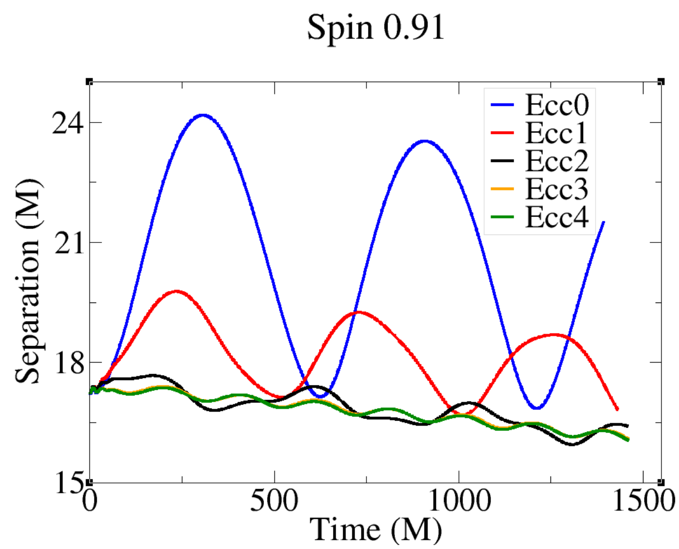


FIG. 2. Proper separation of two black holes in the Spin 0.91 superkick configuration over several eccentricity reduction iterations. The amplitudes of the oscillations decrease after each iteration, showing that the orbit is becoming more circular.

### Spin 0.99 Simulation

For our 0.99 magnitude spin simulation, our initial guess for the orbital parameters was not nearly as good. As a result, the orbit was extremely eccentric, and the black holes actually inspiraled and merged in about a fourth of the time it normally takes for an eccentricity reduction run to stop and guess improved initial parameters. Also, the spin orientation completely flipped sign during this simulation, which appears to be an interesting signature of extreme precession.

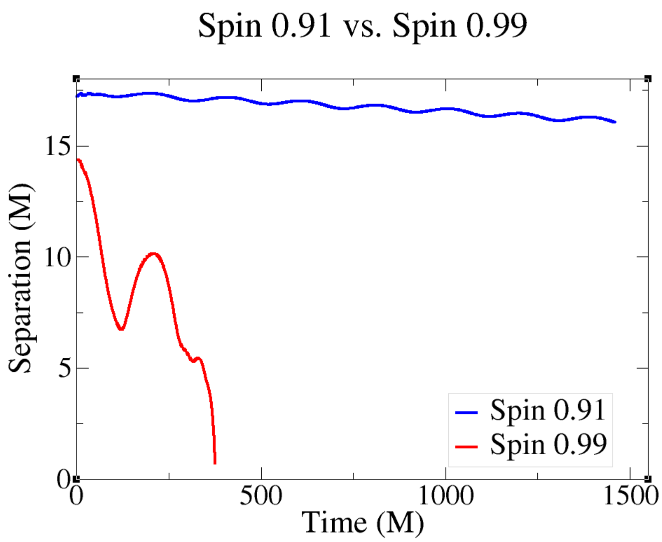


FIG. 3. Comparison between the proper separation of two black holes in a superkick configuration for our Spin 0.91 and Spin 0.99 simulations. The small oscillations in the Spin 0.91 separation indicates the orbit is slightly eccentric. The Spin 0.99 separation shows extremely large eccentricity and an unusually quick merger.

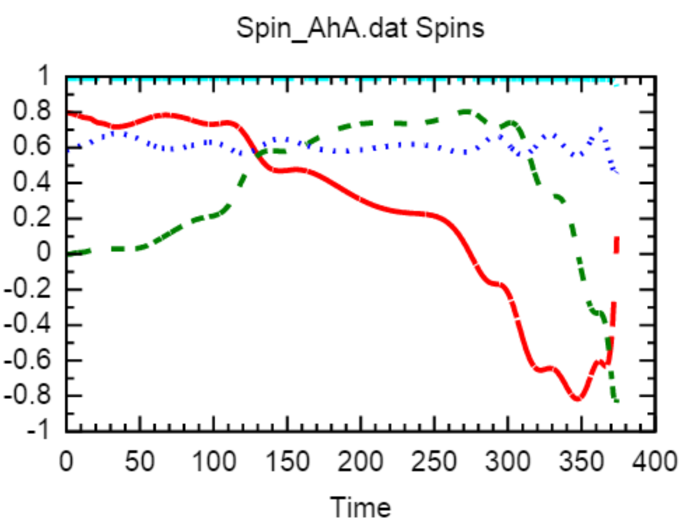


FIG. 4. Spin of one of the black holes from the Spin 0.99 superkick configuration. The red line indicates the spin magnitude in the x direction; the green line, the y direction; and the blue line, the z direction. The flip of the x-spin from 0.8 to -0.8 appears to be a sign of extreme precession.

We conducted more simulations to determine whether the rapid merger and the extreme precession were physical effects due to high eccentricity, or whether they were due to insufficient numerical resolution. In particular, we ran the simulation with the same initial conditions but at higher resolutions. We started the run at Level 1 and we went to Level 3. These higher resolution runs agreed with the first run, and we conclude that the results were the effect of extreme procession.

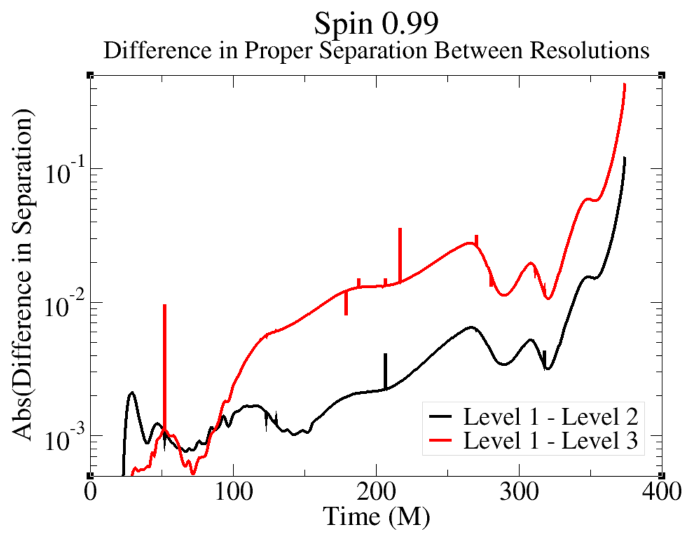


FIG. 5. Absolute value of the difference in the proper separation of two black holes in the Spin 0.99 superkick configuration between resolution levels. The differences have been shown on a log scale. The plot shows that differences between resolutions is converging, and thus we conclude that the effects we see in this simulation are due to extreme resolution and not numerical error.

Although the initial spin-0.99 run is interesting in terms of precession, for LIGO purposes we still wish to produce a superkick simulation with spins of 0.99 and an initially circular orbit. The previous run did not complete enough orbits for our current eccentricity reduction and orbital parameter updating code to be useful, so we created additional 0.99 spin simulations where we manually increased the initial angular frequency slightly. A simulation with a more circular orbit is currently running.

### Eccentricity Reduction Code

While these simulations ran, I improved the code that we use to calculate the eccentricity and update the initial orbital parameters, which are the angular and radial velocities. This is done by measuring the derivative of the angular velocity with respect to time, or the angular acceleration, from the simulation and then fitting it to a function with undetermined coefficients. For example, one of the functions we fit to, which we call F1cos1, has the form

$$\frac{d\omega}{dt}(t) = a(b-t)^{-11/8} + c\cos(dt+e), \tag{1}$$

where the first term represents the leading-order post-Newtonian increase of $d\omega/dt$, and the second term represents oscillations caused by eccentricity. Here $a$ is the amplitude of the power term, $b$ is the maximum time of the simulation, $c$ is the amplitude of the oscillatory term, $d$ is the frequency of the oscillations, and $e$ is the phase of the oscillations. After these coefficients are determined by fitting to the angular acceleration, they can be used to compute the eccentricity of the orbit and the changes in the orbital parameters which will cause the eccentricity to be smaller in the next iteration.

I recoded the fitting functions in a way that is now more organized, efficient, and understandable. I then updated a test code and used it to make sure that the new functions work properly and fit to the data with small errors. We also added the ability to impose minimum and maximum bounds on the fitting coefficients. Now we can ensure that physically positive coefficients, such as frequency, remain positive while fitting to the data and we can prevent errors in coefficients being much higher or lower than they should be.

We tested that given a good initial guess for the fit coefficients, the fitting code can perform its fits and recover the correct parameters, within a small error. We wrote a new version of the eccentricity-reduction code in C++, which is several times faster than the old, Python version, and we fixed many bugs in the code along the way. It now gives reliable calculations for the eccentricity which agree with the old version and succeed in cases where the old version failed.

Furthermore, this code now gives error bars for the eccentricity and the updated orbital parameters. This will allow us to see how well the code worked. If it returns a huge error bar, then we will know that the results may be unreliable and we will be able to proceed with more caution if we choose to continue to use the results or we may just abort the process early. For instance, if the simulation has only computed a few orbits, the eccentricity should have a fairly large error bar to reflect that.

This code still does not work on the extremely eccentric spin 0.99 simulation, but the old version did not either. This is because the number of orbits for this simulation is too small, and the eccentricity too large, for a reliable measurement. We may need a separate way to deal with highly eccentric or anomalous cases like this. At the moment, we have implemented a case where the code will simply stop early and not return any orbital parameter updates if the initial eccentricity is too high.

We are currently in the final stages of implementing our new version of the eccentricity code into the main simulation. We have written code which will read current files from the simulation, and call the eccentricity reduction code. Checkpoints and restarts are called to ensure that the program can continue where it left off. Thus, our new code will be able to run *during* the simulation. We will be able to use the information it gives us to understand how our simulations are progressing and fix or abort them if necessary. If we see that the orbital parameter guesses are not improving as the simulation continues to run, it can be stopped and restarted with the new parameters. If we see that the new orbital parameters are already known accurately enough, we can stop the simulation early. This should save a significant amount of time, as we will no longer have to wait for a set amount of time to calculate quantities and observe how the run is progressing.

## CONCLUSIONS AND FUTURE WORK

Both our spin 0.91 and 0.99 simulations are still running and have not yet been completed. In the future, once the simulations attain an eccentricity sufficiently close to zero, we will start three more superkick simulations identical to the original except with the spins oriented differently. After the simulations have been completely evolved, we will fit the strength of the superkick versus the orientation of the spins to determine the orientation that would give the maximum and minimum superkicks for these setups. Finally, simulations for these maximum and minimum configurations will be run and the waveforms will be analyzed to determine if LIGO can distinguish the difference. The resulting gravitational waveforms from these high spin configurations will then be analyzed and compared to those produced in a previous low, 0.50, spin simulation where it was determined that LIGO could not detect the difference between spin orientations.

The eccentricity reduction code has been improved to calculate error bars, constrain fitting coefficients, execute faster, and run during the simulation. The new code must be tested on an actual simulation to ensure everything works properly and any bugs must be fixed before it is implemented into the main simulation framework.

## ACKNOWLEDGMENTS

A huge thanks to my mentor, Dr. Mark Scheel, for his patience teaching me and answering all of my questions. I thank Dr. Dan Hemberger for many helpful interactions. I also thank LIGO for providing this opportunity and arranging lectures and trips, and I thank Caltech for hosting this program. Our binary black-hole evolutions were performed using the Spectral Einstein Code SpEC [7]. Computations were performed by the supercomputers Zwicky and Comet, at Caltech and the San Diego Supercomputer Center, respectively. Funding for this project was provided by LIGO and the NSF.

**REFERENCES**


[1] Manuela Campanelli, Carlos O. Lousto, Yosef Zlochower, and David Merritt. Maximum gravitational recoil. *Phys. Rev. Lett.*, 98:231102, 2007.

[2] J. A. Gonzalez, M. D. Hannam, U. Sperhake, B. Brügmann, and S. Husa. Supermassive recoil velocities for binary black-hole mergers with antialigned spins. *Phys. Rev. Lett.*, 98:231101, 2007.

[3] Carlos O. Lousto and Yosef Zlochower. Hangup Kicks: Still Larger Recoils by Partial Spin/Orbit Alignment of Black-Hole Binaries. *Phys. Rev. Lett.*, 107:231102, 2011.

[4] Alessandra Buonanno, Lawrence E. Kidder, Abdul H. Mroué, Harald P. Pfeiffer, and Andrea Taracchini. Reducing orbital eccentricity of precessing black-hole binaries. *Phys. Rev. D*, 83:104034, 2011.

[5] Harald P. Pfeiffer, Duncan A. Brown, Lawrence E. Kidder, Lee Lindblom, Geoffrey Lovelace, and Mark A. Scheel. Reducing orbital eccentricity in binary black hole simulations. *Class. Quantum Grav.*, 24(12):S59–S81, 2007.

[6] Robert Owen, Jeandrew Brink, Yanbei Chen, Jeffrey D Kaplan, Geoffrey Lovelace, Keith D Matthews, David A Nichols, Mark A Scheel, Fan Zhang, Aaron Zimmerman, and Kip S Thorne. Frame-Dragging Vortexes and Tidal Tendexes Attached to Colliding Black Holes: Visualizing the Curvature of Spacetime. *Phys. Rev. Lett.*, 106:151101, 2011.

[7] http://www.black-holes.org/SpEC.html.