Robert Nagel, mentored by Roland Haas
Department of Computer Science, College of Engineering, University of Illinois at Urbana-Champaign

INTRODUCTION

At the NCSA, the Gravity Group works with LIGO to model and simulate collisions of binary black holes and binary neutron stars, running numerical relativity simulations on the Blue Waters supercomputer.

In order to extract the gravitational waveforms produced by the simulations, there is a computational pipeline, originally a mixture of Python, C, and Mathematica code.

AIM

The multi-lingual nature of the existing pipeline has several downsides:

- C code requires compilation before execution, adding an extra step to the set-up process
- Mathematica is a very heavy dependency, and takes up lots of disk space to install
- Mathematica requires an (expensive!) license, which may be hard for a potential collaborator to obtain
- The overall system becomes harder to understand

In order to resolve these problems, my task has been to either rewrite or replace the C and Mathematica portions of the toolchain with equivalent Python code.

METHOD

Replacing Mathematica

The primary use of Mathematica was so that a library called SimulationTools could extract the gravitational waveform from the simulation outputs.

In previous years, the Gravity Group has developed a library called POWER, the Python Open-Source Waveform Extractor, that performs the task of extracting a gravitational waveform from a simulation. Once POWER was integrated into the pipeline, Mathematica was no longer a necessary component and could be removed.

Additionally, SimulationTools was used to compute a handful of pieces of metadata about the waveform. These quantities are defined with simple formulae, and thus were straightforward to implement.

Replacing C

Two of the metadata quantities are significantly more involved to compute: the orbital eccentricity and the mean anomaly.

In the existing pipeline, these computations were handled by a specialized codebase written by Sarah Habib, a previous member of the Gravity Group. The code was a Python script wrapping a C executable that did the bulk of the calculations.

In total, 60 C functions were converted, producing ~4800 lines of Python code.

RESULTS

- The simulation post-processing pipeline now only uses Python.
- Mathematica is no longer required by any part of the codebase.
- Unfortunately, parts of the pipeline have become slower. In particular, the computation-heavy C code did become noticeably slower when converted to Python.

FUTURE WORK

- Clean up the interfaces between components: Currently, components such as POWER and the eccentricity extraction code are invoked with subprocesses, a holdover from when those components were not written in Python. Instead, they preferably would be invoked as a Python library, also becoming potentially reusable by other researchers.

CONCLUSIONS

A heterogeneous computational pipeline for extracting gravitational waves, previously written in three languages, now only uses one. The pipeline is now easier to set up, because it requires neither code to be compiled before use nor an expensive Mathematica license to install. Future collaborators may find the system to be more approachable, as they only need familiarity with one language instead of three.

ACKNOWLEDGEMENTS

I would like to acknowledge Roland Haas for providing mentorship and guidance throughout the program, and the NCSA SPIN program for making this opportunity possible.