

# Eliot Heinrich

Waltham, MA  
802-310-1278 (eliot.heinrich@gmail.com)

[linkedin.com/in/eliot-heinrich-36200a67](https://linkedin.com/in/eliot-heinrich-36200a67)

[eliotheinrich.github.io](https://eliotheinrich.github.io)

## Summary

Graduating PhD candidate in computational physics. Skilled in designing and optimizing high-performance C++ software, scalable simulations in distributed environments, HPC infrastructure, and parallelized code. Seeking to apply these skills to a role in software engineering and available to start work before graduation.

## Education

<b>Boston College</b> <i>Physics (PhD, MS), GPA: 3.96</i>	Sept. 2020 – Expected May 2026
<b>University of Vermont</b> <i>Computer Science (BS), Physics (BA), Mathematics (BS), GPA: 3.91</i>	Sept. 2016 – May 2020

## Experience

<b>PhD Candidate, Quantum Simulation Research, Boston College</b> - Designed object-oriented C++ simulation library for composable quantum simulations, achieving 3-10x performance gains over standard libraries by leveraging SIMD instructions, circuit transpilation, asynchronous instruction execution, and algorithmic optimization. - Authored five first-author publications related to statistical mechanics, quantum information, large-scale numerical studies, and classical algorithms for simulating and characterizing quantum systems.	Sept. 2020 – Present
<b>HPC Research Assistant, Boston College Research Services</b> - Systems administrator for Boston College's Linux HPC cluster (284 nodes, ~500 users). - Assisted 35+ interdisciplinary research groups to design, deploy, and utilize scientific software. - Developed Python-based automated tools for job scheduling, cluster usage data aggregation/visualization, and performance monitoring, improving processes and infrastructure. - Authored documentation/best-practice guides for parallel computing, OpenMP, MPI, Linux, and Python.	Jan. 2023 – Present
<b>Quantum Theory &amp; Software Intern, MIT Lincoln Laboratory</b> - Extended a C++ quantum simulator to support higher-dimensional systems (qutrits) and added new error/noise channels, requiring modifications of core state representation and operator implementations.	June 2022 – Aug. 2022

## Projects / Publications

Full list of publications available at [eliotheinrich.github.io/publications](https://eliotheinrich.github.io/publications)  
More projects, including interactive WebGL applets, available at [eliotheinrich.github.io/projects](https://eliotheinrich.github.io/projects)

<b>qutils (C++/Python API)</b> - Modular quantum simulation library supporting simulator backends representing various simulable subtheories of quantum mechanics (i.e. stabilizer states, tensor networks, free fermion dynamics). - Flexible backend-agnostic circuit abstractions representing symbolic quantum/classical instructions, enabling multiple efficient and scalable simulation strategies under a unified API. - Authored extensive unit tests to validate numerical accuracy and benchmark performance.	<a href="https://github.com/eliotheinrich/qutils">github.com/eliotheinrich/qutils</a>
<b>Distributed filesystem (C)</b> - Wrote distributed FUSE filesystem in C using process (MPI) and thread (pthread) level parallelism. - Supports custom-built RAID 4 and RAID 5 striping and rebuild protocols, protecting against drive failures.	<a href="https://github.com/eliotheinrich/spool">github.com/eliotheinrich/spool</a>
<b>dataframe (C++/Python)</b> - Built a multiprocessing Python pipeline for efficient parameter-sweep simulations of arbitrary models. - Data stored in pandas-like frame written in native C++, supporting efficient parameter-based query of statistical properties and allowing for efficient downstream analysis.	<a href="https://github.com/eliotheinrich/dataframe">github.com/eliotheinrich/dataframe</a>

## Skills

**Tools:** C++, C, Python, Rust, CMake, Git/GitHub, Linux, LaTeX, PyTorch, SQL, Docker, CI/CD, Jira  
**Techniques:** High-performance parallel computing, object-oriented programming, advanced numerical methods, designing numerical experiments, data visualization, API design, debugging, optimization, profiling, scalable code design, technical communication, machine learning, deep learning