

Jacob B. Collins

jbcollins@csuchico.edu | (925) 207-0765 | GitHub.com/collinsjacob127 | LinkedIn.com/in/collinsjacob127 | Chico, CA

EDUCATION

M.S. Computer Science | Expected Graduation: December 2026

B.S. Computer Science Cum Laude | **Cumulative GPA: 3.74** | December 2025

Data Science Certificate | December 2025

HPC & Quantum Computing Club | President

CSCI Research Enthusiasts Club | President

California State University, Chico

SKILLS

Programming: C/C++, Python, R, MPI, OpenMP, Pthreads, CUDA-Q, Qiskit, UDS, Winsock

Tools & Libraries: OpenCV, NetworkX, PyTorch, Slurm, Docker, Singularity, SDL, Git CLI, CMake, GCE

PUBLICATIONS

Mission Planning Simulation and Design Software Scaling for Shared and Distributed Memory Computing

Integrated MPI & OpenMP modifications to NASA's General Mission Analysis Tool. Achieved 2,387% speedup on 62x Monte-Carlo simulation trials distributed across 36x 2-core nodes, compared to sequential runtime.

IEEE Aerospace Conference 2026 10.0304

Skills: MPI, OpenMP, CMake, Bash, C++

Quantum Semiprime Factorization: Leveraging Grover's Algorithm for Efficient Prime Decomposition

Designed a dynamic quantum circuit generator using CUDA-Q which can find factors of large semiprimes using asymptotically lesser qubits than existing methods. Ran simulations with Slurm using a custom Singularity container on the SDSC Expanse supercomputer.

CSCSU 2025 Research Conference

Skills: Slurm, Singularity, CUDA-Q, C++

PROJECTS

Distributed Sieve of Eratosthenes

Wrote a distributed prime number sieve in C which uses MPI to divide the search region evenly across all nodes. OpenMP is used to parallelize the local filtering through each node's region. The prime sieve itself is stored as a binary mapping to maximize memory efficiency. Tests were run using Slurm with provided Singularity containers on SDSC Expanse. I achieved 13,482% speedup (99.3% parallel) compared to the sequential runtime searching for all primes below 10 billion.

Available upon request

Skills: Slurm, MPI, OpenMP, C

Low-Latency Cross-NAT Multiplayer Game

Designed and implemented the relay server, cross-NAT P2P, physics, engine, and website for a 2-player arcade-style fighting game with cross-platform support, using only standard library functions for everything besides the GUI, which uses SDL3. Cross-platform support for Windows and UNIX systems. CMake was used for build automation and generation of installation wizard. Website and relay server both hosted on Google Compute Engine (GCE) VMs.

Source Code on GitHub

Skills: UDS, Winsock, CMake, GCE, SDL, C++

Iterative Network Simulation

Simulated an iterative adaptation of the prisoner's dilemma on a variety of networks. Automated generation of animations visualizing model statistics.

Hosted on GitHub

Skills: Python, NetworkX

Road Network Processing & Analysis

Constructed a conversion & processing pipeline for extracting road networks from Open Street Map (OSM) data. Prioritized memory efficiency by parsing file streams to generate a network with less unnecessary overhead.

Writeup on Pages | Source on GitHub

Skills: Python, NetworkX, R, Bash

EMPLOYMENT

Lead Student Research Assistant

Led research efforts to design Quantum Computing algorithms, to be compared against equivalent parallel solutions. Quantum circuits implemented with CUDA-Q in C++.

Chico State Enterprises

Dr. Jaime Raigoza, Dr. Sam Siewert

June 2024 – December 2024