

# Charlie Sowerby

[charlie.sowerby@gmail.com](mailto:charlie.sowerby@gmail.com) | [charlie.sowerby.com](http://charlie.sowerby.com) | [github.com/csowerby](https://github.com/csowerby)

## Education

June 2021 **University of California, Los Angeles** - Physics (B.S.) & Mathematics (B.S), GPA: 3.72/4.00  
**Relevant Coursework:** Intro C++ (x2), Machine Learning, Optimization, Differential Geometry, Linear Algebra (x2), Electronics, Nuclear Physics Lab.  
**Honors:** Highest Honors in Physics (Departmental Honors), College Honors Program

## Technical Skills

Languages	Python, C++, C, MATLAB, Java, HTML/CSS
Software/OS	PyTorch, Numpy/Scipy/Sympy, Git, Bash, L <sup>A</sup> T <sub>E</sub> X, GNU Make/Debugger, lldb, Linux/Unix
Hardware	Circuit design, Soldering, Raspberry Pi/Arduino Programming, Computer Aided PCB Design
Mathematical	Numerical methods for differential equations (coupled PDE's in particular)

## Technical Research and Lab Experience

All research performed at the *Basic Plasma Science Facility, UCLA*. For more information on my research visit my [website](#).

Aug-Jun 2021	<b>Eigenmode Solver</b> <ul style="list-style-type: none"><li>Reformulated a simplified electrostatic version of the Braginskii two-fluid equations to include electromagnetic corrections.</li><li>Learned how to use an existing linear eigenmode solver and modified it to incorporate the derived corrections and used the finite difference method to numerically solve PDE's.</li><li>Implemented this eigenmode solver to simulate more accurate drift waves in the Large Plasma Device at UCLA.</li></ul>
Mar-Aug 2020	<b>Plasma Imaging</b> <ul style="list-style-type: none"><li>Programmed Raspberry Pi/Arduino Camera Modules in an attempt to capture images of plasmas on timescales of less than 100ns.</li><li>Experimented with different operating systems on Raspberry Pi, (Linux vs Real Time Operating System), etc to get optimal quick image capturing.</li><li>Experimentally determined which combination of hardware and software (written mostly in C++) on these inexpensive devices was able capture images with the least latency and jitter using a pulse generator and flashing LEDs.</li></ul>
Mar-Jun 2019	<b>Relay Circuit</b> <ul style="list-style-type: none"><li>Designed and soldered my own remotely controlled relay circuit to be used for adjusting resistors in a Langmuir Probe</li><li>Programmed a Raspberry Pi to control the circuit and implemented a network socket connection to the lab computer for easier control.</li><li>Modeled a digital PCB using Altium's CircuitMaker to be printed to minimize physical space occupied by the circuit.</li></ul>

## Theoretical or Academic Experience

Mar 2020 - Now	<b>Reading Program in Graduate-Level Mathematics</b> <i>University of Chicago Mathematics / UCLA Mathematics</i> Independently studied two separate topics in advanced mathematics: (1) Fourier/Complex Function Theory (ongoing) with UChicago Professor <a href="#">Carlos Kenig</a> and (2) Smooth Manifolds/Riemannian Geometry (complete) with UCLA graduate student <a href="#">Nicholas Boschert</a> .
Winter 2020	<b>Undergraduate Grader</b> <i>UCLA Physics &amp; Astronomy</i> Graded homework assignments for upper-division class Physics 131A: Mathematical Methods in Physics. Some of the topics discussed were Linear Algebra, PDEs, Fourier Series, Green's Functions, Bessel Functions, and Cauchy's Integral Theorem for complex functions.

## Independent Projects

Aug - Now 2021	<b>Chess Engine</b> - <a href="#">github</a> <ul style="list-style-type: none"><li>Currently finishing up a UCI chess engine in C. The program is currently capable of generating and parsing a tree of 24 million chess positions per second.</li><li>The search algorithm uses Negamax with <math>\alpha/\beta</math> pruning. Currently working on implementing transposition hashing and iterative deepening.</li><li><b>During this project I significantly improved my skills in:</b> data structures, search algorithms, compiler optimization/profiling code, version control with git, organizational coding practices for large projects (<math>\geq 4000</math> lines of code).</li></ul>
-------------------	--