

## Curriculum Contribution Statement

As graduate students could not hold the title of Instructor on Record at Drexel University, my official role was that of a Teaching Assistant (TA). However, I made many significant contributions to both the curriculum and teaching pedagogy to a number of advanced undergraduate courses in the Physics department. For a timeline of each course, please see the appropriate section in the my curriculum vitae.

### *Contemporary Physics I, II, III*

The Contemporary Physics arc is a special series given to the undergraduate Physics majors. The three courses span one instructional year and are designed to fully integrate students into the modern challenges of scientific study. While the students rotated professors each semester, they keep me as the TA to maintain a sense of continuity. The core curriculum was taught from a new book still in preproduction, intended to integrate hands-on computational examples to many of the traditional topics. This necessitated many changes to the assigned exercises and I wrote new problem sets designed to test the areas where students struggled. As a common fixture in the Contemporary program, I was able to guide the development of this course for future students.

### *Introduction to Scientific Computing*

I have a mini-philosophy about computing for today's modern scientist, one should be proficient in three *types* of languages: a compiled language built for speed (C, C++), a prototyping language for design and ease of use (Python, Ruby) and a symbolic algebra language (MAPLE, Mathematica). This course was an introduction to the last category. At the time, there were relatively few academic programs that introduced physics formula in a readily available visual symbolic algebra that we could template our course from. With the department head Michael Vallieres, I helped create much of the curriculum that focused on the unique strengths of symbolic algebra. Pedagogically, the interactive environment of the language itself was useful during my lectures. The examples that were live-coded in class were distributed to students for later study.

### *Computational Physics I, II*

Continuing with my computing language philosophy, this advanced undergraduate course focused on solving problems with both Python and C++. Many of the students I previously taught in Scientific Computing or Contemporary Physics were enrolled in this class. This knowledge allowed me focus the course down to the level of the individual student when designing exercises. The problems sets that I designed were borne from real-world examples that were only amenable with computer such as chaos theory, numerical integration schemes, Monte-Carlo, and time-independent quantum calculations. Many of the exercises I've written are still being used today in the course.