

My most effective professors had practical experience in the subjects they taught, enabling them to design challenging and realistic projects, and facilitate and answer questions that tied together multiple subjects and practice. From the mathematician who used numerical analysis in simulations, to the physicist who sneakily taught the basics of computer simulation in an intro to data structures course, to the computer scientist who invented a programming language to solve specific problems and stood up a language committee across industries and continents to help them solve theirs. I believe it is paramount to maintain a collegial yet challenging classroom environment that resembles the multidisciplinary teams across the nation's finest industrial and research institutions, fully engaging the students' imagination and ability to solve problems as a team.

My educational goal at Texas A&M University is to expose students to the "full stack" or "codesign" approach to using computation to advance science. From hardware to middleware to software, the components must work together to achieve unprecedented performance and therefore enable the nuanced, expansive computation necessary to make scientific progress possible. After a lengthy steady state in computer architectures in mainstream and high performance computing, the past decade has proven to be disruptive, forcing the HPC community to rethink much of the approach to hardware and software while delivering breakthrough science results through simulation.

Capitalizing on my solid background in programming languages, compilers, and algorithms, along with research in performance tools, interprocess communication, programming models, and simulation optimization, I landed at the intersection of these often disjoint fields in my first job after graduation. Working with multidisciplinary teams designing simulations, researchers in tools and programming models, and hardware vendors proved extremely rewarding. I am contributing to the bleeding edge of programming models, interprocess communication, and tools while improving performance of million-line simulation codes by an order of magnitude. Committed to learning fine tuning my craft, I recently took a graduate course in physics to further my understanding of the physics I am simulating.

I have been teaching informally as long as I remember. Tutoring in school, serving as a teaching assistant and tutoring as an undergrad, I was a research mentor for five undergraduate students as a graduate student, and proposed projects for and supervised ten graduate students as staff at LLNL.

My training and work experience have provided me a strong foundation in Mathematics and Computer Science, and I am thus prepared to teach all

core undergraduate courses in Computer Science. I also envision teaching and developing electives based on my research background and on the needs and wants of the department and students. These could include courses on parallel computing, performance tools, computer simulations, and performance analysis. As these are rapidly evolving topics, these courses will be combinations of lectures and reviews of classic and recent literature, as well as projects that would expose the students to conducting modern science.

To conclude, if I am given an opportunity to teach students at TAMU, I would:

1. **guide research:** provide leadership and strategic vision for my students as well as the HPC research community;
2. **build relationships:** establish collaborations with leading research groups;
3. **deliver results:** produce fundamental and practical advancements in the field of HPC.