Teaching Statement

Kathryn D. Huff (katyhuff@gmail.com)

I have spent six years teaching diverse groups of learners and designing relevant curriculum. Using an example-driven teaching philosophy and an inclusive, interactive teaching style, I am able to teach both essential nuclear engineering concepts and practical computational skills.

Teaching Experience

I have taught and developed nuclear engineering curriculum for mumerous guest lectures in both graduate and undergraduate classrooms. However, the bulk of my teaching has been driven by an unmet need for software development training in nuclear engineering [1, 2]. In graduate school, I responded to that need by establishing a peer teaching organization (The Hacker Within (THW)), creating curriculum, and teaching workshops. As a postdoctoral fellow, I wrote a book [3] and chaired the steering committee of the Software Carpentry Foundation (SCF), an international non-profit devoted to scientific computing education [4, 5].

With THW and SCF, I have designed and taught scientific computing curriculum in seventeen workshops to a total of approximately six hundred students. These workshops varied from two to ten hours per day and from two days to two weeks in length, addressing topics such as software design, data analysis, verification, validation, and reproducible quality control. During these workshops, I had the opportunity to teach very heterogeneous groups of students. Typical workshop demographics included students at the undergraduate, graduate, and faculty level and traversed disciplines from physics to limnology [2].

Teaching Philosophy

Through experimentation with instructional design and teaching style as well as direct feedback from learners [2, 4], my teaching philosophy has come to emphasize inclusive student engagement as well as example-driven, interactive curriculum. Interactive exercises in a supportive classroom environment can be extremely effective at engaging students across skill levels and backgrounds [6]. Combined with targeted strategies to reduce stereotype threat and overt bias in the classroom [7, 8], including techniques such as randomized call order and name-blinded assessment, this approach can help to retain high-quality students, particularly those from underrepresented groups.

In both classrooms and workshops, I avoid lengthy soliloquy by interjecting short, relevant, interactive exercises into the lesson. In the workshops, for example, 5-10 minute conceptual explanations are followed by similarly short problem-solving sprints. During these sprints, my students actively engage the task.

However, this interactivity is only effective when concepts are also introduced in a well-ordered manner. Students must form a clear mental map of concepts to become prepared to link and categorize ideas in their future careers. Such a map can only be conveyed with instructional design that reflects the well-organized concept map of the instructor [9].

To combine student engagement with conceptual clarity, I emphasize example- and project-driven course design. By presenting concepts in the context of practical challenges, I link theory with the practice of nuclear engineering. In this way, I am also able to cover essential nuclear engineering content while also embedding practical and computational skills into relevant exercises.

Teaching Interests

My background in physics and nuclear engineering has prepared me to confidently teach a variety of classes in the department at ESU. In addition to supporting the department by teaching existing courses, I am prepared to expand the curriculum in three primary areas of my expertise: modeling and simulation, fuel cycle analysis, and waste disposal technology.

Among courses already offered in the Department of Nuclear Engineering, I am prepared to teach fundamental nuclear engineering curriculum such as NE101 and NE102. Further, my research in nuclear fuel cycle and repository analysis has amply prepared me to teach courses offered in nuclear fuel cycle analysis (i.e. NE571 and NE572) and nuclear waste disposal modeling (i.e. NE666). The highly computational nature of my current work and my experience teaching scientific computation are a good foundation on which to instruct numerical modeling courses such as NE555 and NE556. Also, my research experiences in experimental accelerator and telescope physics have prepared me to teach additional fundamental courses concerning accelerator and detector physics (i.e. NE306 and NE406).

I would also be delighted to develop an undergraduate or early graduate course on 'Simulation in Nuclear Engineering,' for example, could focus on challenges in nuclear engineering modeling and simulation such as simulation design, validation techniques, and data workflows. This project-driven semester, built around canonical benchmarking exercises in reactor physics, would motivate use of a suite of software development practices emphasized in my book, Effective Computation in Physics [3], which was designed to accompany such a course.

Additionally, I might develop a pair fuel cycle analysis and policy courses for either undergraduates or early graduate students. The first, 'Fuel Cycle Modeling and Analysis,' could cover fuel cycle metrics development and computational modeling techniques for analyzing the impacts of potential technology and policy choices. I envision using Cyclus as a learning tool for homework exercises that drive toward an analysis capstone project synthesizing course lessons into a simulation exercise or a model development project. The second, introducing 'Nuclear Energy Economics and Governance' could cover important economic considerations of nuclear energy as well as relative roles of public utility commissions on the local scale, the Department of Energy (DOE) and Nuclear Regulatory Commission (NRC) on the national scale, and the International Atomic Energy Agency (IAEA) on the global scale.

Finally, a more advanced graduate course on 'Computational Modeling of Environmental Impacts' might follow your current course on waste management NE666. I would build on that course by focusing on computational modeling methods for analyzing the environmental impact of used fuel storage and disposal, contaminants, and effluents from an array of nuclear power generation processes.

References

[1] J. E Hannay, C. MacLeod, J. Singer, H. P Langtangen, D. Pfahl, and G. Wilson. How do scientists develop and use scientific software? In *Proceedings of the 2009 ICSE Workshop on Software Engineering for Computational Science and Engineering*, pages 1–8. IEEE Computer Society, 2009.

[2] Kathryn D. Huff, A.M. Scopatz, N.D. Preston, and P.P.H. Wilson. Rapid Peer Education of a Computational Nuclear Engineering Skill Suite. In *Transactions of the American Nuclear Society*, volume 104 of *Training, Human Performance, and Work Force Development*, pages 103–104, Hollywood, FL, United States, June 2011. American Nuclear Society, La Grange Park, IL 60526, United States.

[3] Anthony Scopatz and Kathryn D. Huff. *Effective Computation in Physics*. O'Reilly Media, S.l., 1 edition edition, May 2015.

[4] Greg Wilson. Software Carpentry: lessons learned. *F1000Research*, 3, February 2014.

[5] Greg Wilson. Software carpentry: Getting scientists to write better code by making them more productive. *Computing in Science & Engineering*, 8(6):66–69, 2006.

[6] Elizabeth Green. *Building a Better Teacher: How Teaching Works (and how to teach it to everyone)*. W. W. Norton & Company, 1 edition edition, August 2014.

[7] Claude Steele. *Whistling Vivaldi: And other clues to how stereotypes affect us (issues of our time)*. WW Norton & Company, 2011.

[8] Kathryn D. Huff. Book Review - Whistling Vivaldi, May 2014.

[9] Susan A. Ambrose, Michael W. Bridges, Michele DiPietro, Marsha C. Lovett, Marie K. Norman, and Richard E. Mayer. *How Learning Works: Seven Research-Based Principles for Smart Teaching*. Jossey-Bass, 1 edition edition, April 2010.