ApplNumComp: An Open Access Introductory Course for Applied Numerical Computing

Ashlee N. Ford Versypt ⬤ [1,2,3,4]¶ **and Duncan H. Mullins** ⬤ [1,2,3]

1 School of Chemical Engineering, Oklahoma State University **2** Department of Chemical and Biological Engineering, University at Buffalo, The State University of New York **3** Department of Engineering Education, University at Buffalo, The State University of New York **4** Institute for Artificial Intelligence and Data Science, University at Buffalo, The State University of New York ¶ Corresponding author

Summary

ApplNumComp is a repository of open educational resources supporting an introductory course on Applied Numerical Computing. Topics include Git for version control, LaTeX for typesetting, reproducible research computing, and MATLAB and Python for high-level programming and scientific computing applications of solving systems of differential equations, estimating parameters for models using regression, creating publication quality figures, and developing graphical user interfaces. The lessons can be used collectively for a similar course at another institution, for independent study by an online learner, or for incorporation of individual lessons (modules or examples) into other courses that involve related topics.

Statement of Need

An introductory course on Applied Numerical Computing was developed for upper division undergraduate and graduate students interested in research and industrial projects involving mathematical modeling using computational tools. The class was designed to fill a gap distinct from introductory programming and numerical methods courses and was intended to be very practical. Graduate students new to computational research were the primary student group targeted for the course topics. We have previously detailed the course (Ford Versypt, 2019) and the final course project (Ruggiero et al., 2018).

No single textbook covers all of the topics in the Applied Numerical Computing course. As the course emphasized best practices for reproducible research computing, we sought to apply the concept to the course materials for a reproducible computing educational experience. Here, we have curated a set of open educational resources for others to engage with the course lessons outside of the institution's learning management system and outside of the formal course. The main repository is organized into a set of lessons as Markdown-formatted text-based open-source webpages that include code snippets, explanatory text, reflection questions, sample solution software files, links to supplemental videos, related references, and other instructional materials. Each lesson can be considered as a stand alone module on a particular computational topic; however, there are conceptual links between some of the lessons. Many of the lessons include detailed MATLAB or Python coding examples, or both as in the case for Lesson 10: Python and MATLAB Plotting. For instance, Lesson 13: Parameter Estimation in Python provides a worked example for fitting parameters to one differential equation and the solution .py file followed by a second example for fitting parameters to multiple differential equations. Related Lessons 11 and 12 provide supportive general information and complimentary content with examples in MATLAB. Documentation associated with MATLAB or Python often only shows much

simpler examples. Open-source worked examples are limited for even just solving multiple differential equations simultaneously independently from the parameter estimation. The full list of lessons is available here. Other supporting materials include a list of suggested readings with links to the sources, the full set of computational assignments, and the course learning objectives.

Our experiences with the course and materials have been positive. Over the 5 course offerings, 63 graduate students and 27 undergraduate students took the course. The students were majoring in the following 12 fields: chemical engineering, mechanical engineering, aerospace engineering, civil engineering, environmental engineering, chemistry, plant and soil sciences, mathematics, electrical and computer engineering, petroleum engineering, physics, and industrial engineering. Most similar electives for cross-listed undergradaute/graduate credit averaged 10 students in a term, while this course averaged 18 per term. Typically, similar electives were offered only once in four semesters. This course was popular enough to offer each fall semester. There was a drop in graduate enrollment in the course in 2020, likely due to COVID-19 induced reductions in the overall number of new graduate students in science and engineering fields at the university. The course became highly recommended for first year graduate students in computational research groups across campus and for all first year mechanical and aerospace engineering graduate students. Additionally, it was included in the interdisciplinary Data Science minor available to undergraduate students. The conference proceedings paper detailing the course (Ford Versypt, 2019) has been dowloaded from the American Society for Engineering Educaion Papers on Engineering Education Repository 698 times as of May 1, 2025. In January 2021, the authors left the university where the course was developed, and multiple faculty reached out to request access to the course materials either for offering the course themselves in the future or for sharing the materials with their students for self-paced learning. The ApplNumComp GitHub repository is already filling that direct need. Trainees in the Ford Versypt Lab continue to use the course materials for training and examples. The repository has also been disseminated through the Computer Aids for Chemical Engineering website. The authors and the Computer Aids for Chemical Engineering Corporation trustees have discussed how to further disseminate lessons in modular form from ApplNumComp into a variety of chemical engineering courses across other universities. The authors have already presented these materials at conferences for the American Society for Engineering Education and the American Institute for Chemical Engineers (AIChE) and a webinar for the AIChE. A workshop was developed for sharing the materials in other venues including the 2022 Chemical Engineering Summer School, held once every five year for junior faculty. The corresponding conference proceedings paper on the workshop (Johns et al., 2023) has been downloaded 581 times as of May 1, 2025.

Author Contributions

Ashlee N. Ford Versypt developed the Applied Numerical Computing course at Oklahoma State University and taught it annually 2016-2020. The materials presented here were developed and delivered in a hybrid of in-person, online synchronous, and online asynchronous formats in Fall 2020. Duncan H. Mullins was a PhD student from Dr. Ford Versypt's lab, took the course in Fall 2019, served as a teaching assistant for the course in 2020, and aided in conversion of the course to this online open-access version.

Acknowledgments

References

Ford Versypt, A. N. (2019). An interdisciplinary elective course to build computational skills for mathematical modeling in science and engineering. *Proceedings of the ASEE Annual Conference.* https://doi.org/10.18260/1-2--32072

Johns, A. N., Hesketh, R. P., Stuber, M. D., & Ford Versypt, A. N. (2023). Numerical problem solving across the curriculum with python and MATLAB using interactive coding templates: A workshop for chemical engineering faculty. *Proceedings of the ASEE Annual Conference.* https://doi.org/10.18260/1-2--43749

Ruggiero, S. M., Zhao, J., & Ford Versypt, A. N. (2018). Building a MATLAB graphical user interface to solve ordinary differential equations as a final project for an interdisciplinary elective course on numerical computing. *Journal of Computational Science Education*, *9*(1), 19–28. https://doi.org/10.22369/issn.2153-4136/9/1/3