Engineering Educators Bringing the World Together
2025 ASEE Annual Conference & Exposition
Palais des congrès de Montréal, Montréal, QC • June 22–25, 2025
Paper ID #47781

BOARD #476: Work in Progress: Combining Python and Simulation to Offer Easy Visualization in Early Years Teaching

Dr. Susannah Cooke, ANSYS, Inc.

Susannah Cooke is a Senior Product Manager at Ansys, managing Ansys Academic software. She works with universities to ensure that Ansys tools can be deployed to best effect in teaching and research. She holds an MEng and DPhil in Mechanical Engineering from the University of Oxford, where her doctoral thesis focused on fluid flow around tidal turbine arrays. She is excited by the overlap between industry engineering and pedagogical practices, especially where these give students a springboard into their careers.

Dr. Kaitlin Tyler, ANSYS, Inc.

Kaitlin Tyler is currently a Senior Academic Program Engineer at Ansys. Her role focuses on supporting the usage of Ansys tools in academia, with an emphasis on materials teaching and pre-university engagement. She is also the lead for the Ansys Academic Content Development Program, which focuses on developing instructional content to support integration of Ansys tools in curriculum. Her background is in materials science, with a PhD in the subject from the University of Illinois Urbana-Champaign.

She is very involved in ASEE. At the publication of this paper, she is the Awards Chair (past Division Chair) for the Materials Division and Chair Elect for the Corporate Members Council.

Work in Progress: Combining Python and Simulation to Offer Easy Visualization in Early Higher Education Teaching

Abstract

The power of engineering simulation tools is well known in industry; simulation skills are listed as a key area for new graduates. Another benefit simulation can add to a curriculum is aiding students in visualizing phenomena, particularly in "real-world" scenarios. But the tools themselves can be overwhelming for early-years students who could benefit most, requiring additional instruction time in already packed curriculums for full efficacy. In this paper, we present work in progress to leverage recent developments where the Ansys simulation suite has become more accessible through APIs and Python libraries, allowing the development of teaching resources designed for the higher education classroom. This work looks to bring the benefits of simulation into the curriculum without additional student training requirements.

Two implementation approaches will be discussed here. The first utilizes the Jupyter Notebook (or equivalent) interface to engage with the software. Students and instructors can interact with either the code or the simulation tools if desired, providing opportunity to expand depending on course needs. The second approach involves a Python-based application with front-end user interface. Students in this case interact with the desired visualizations via a simple "app", leaving the more complex simulation software unseen in the background.

Details of the teaching resource creation process, implementation challenges, and example curriculum integration opportunities will be shared, as well as preliminary feedback from academics and students using the tools presented. Our hope with this work is to lower the energy barrier for including simulation in the engineering curriculum, allowing students to take advantage of the visualization capabilities and familiarize themselves with the concepts of simulation tools early in their degree journey.

1. Motivation: strengthening experiments with simulation to enhance students' understanding

The skills which engineering students need upon entering the workforce continue to change and evolve. The World Economic Forum's Future of Jobs 2025 Report [1] highlights the growing desirability of skills related to technology but also shows that employers place increasing value on resilience, flexibility and agility. Students entering the job market need not only the core knowledge of their field and the ways to apply that knowledge, but also awareness of adjacent skillsets – the ways in which they could adapt and expand their skills in future if needed. One such adjacent skillset is usage of simulation and programming to solve field-specific problems[2], [3], [4], [5]. This has become important enough to be included in ABET program criteria[6]. But including courses covering these skills in engineering curriculum has proved challenging; introductory programming courses often face difficulty due to inability to interest both major and non-major Computer Science (CS) students[7], [8], [9] and simulation topics are generally offered later in engineering curriculum, giving students less time to gain meaningful experience before entering the workforce[10], [11], [12].

Meanwhile, industry itself is moving towards greater automation, integration and customization of technology and processes. In particular, interoperability of tools and interconnectivity of data are key trends of 'Industry 4.0' [13], [14]. For Ansys simulation tools,'PyAnsys' libraries have been released in recent years [15] which are open-source Python libraries that interface with Ansys solvers such as MAPDL (Mechanical), Fluent (Fluids), AEDT (Electronics) and others. Users can then connect Ansys simulations to the variety of tools in the wider Python ecosystem. This has clear applications in research or commercial engineering environments, and research has already leveraged these tools for optimization, automation and more across multiple disciplines [16], [17], [18], [19]. These libraries could also be deployed in university teaching, providing an opportunity to teach programming skills as part of courses students are already interested in [9]. In particular, where curricula are already trying to create programming exercises to complement fundamental learning, Python libraries linked to simulation are ideal to support this goal.

Simulation tools offer a way for students to visualize fundamental physical behaviors in real-world situations. Visualization is known to support learning in many disciplines[20], [21], [22], [23], but it can be particularly valuable where the theory being taught is non-intuitive to introductory learners, or where the limitations of theory needs to be investigated in a more complex scenario. Lab experiments are traditionally employed to support this need, but simulations are already used in some courses where lab experiments are not included[24].

2. Background: development of teaching resources with simulation to support curriculum

The Ansys Academic Program aims to support students, researchers and educators to use computational modeling and simulation tools effectively in every level of academic life. The Program has several initiatives focused on teaching and learning, including: (1) Free Student Software Downloads to lower the barrier to entry for students worldwide, (2) Student Team partnerships giving industry-level software tools at no cost, (3) the Ansys Funded Curriculum Program, which gives grants to academic institutions to support integration of simulation software in the classroom, and (4) open-access Teaching and Learning Resources, designed by Ansys and academic collaborators to support both the educator and the learner.

From the point of view of the Academic Program, the emergence of scripting tools such as the PyAnsys libraries presents an opportunity: what if students in the earliest years of a degree could benefit from using industry-standard simulation tools to visualize key theoretical fundamentals, but with no need for students to be simulation experts themselves yet. Running an advanced computational simulation from Python code means that educators can pre-set the geometry, boundary conditions and other key requirements within the code, leaving students only a small number of variables to interact with and placing those within a more user-friendly interface. This idea was initially tested in parallel with more advanced student scripting activities[25], but then explored further in the form of teaching and learning resources for wider use.

3. Implementation of Jupyter environments for fluid dynamics and electronics simulations

Jupyter Notebooks provide a working environment which combines text, images, code, web links and more [26], and have become increasingly popular in education. They can complement traditional equation-focused engineering teaching by allowing students to explore and visualize aspects of theory in a programming environment [27]. As part of the Ansys Academic Program, we were inspired to explore opportunities provided by Jupyter Notebooks to expose students to simulation and visualization of physical phenomena with text, images and links to improve their understanding. A small number of examples were developed in the areas of Fluid Dynamics and Electronics to address fundamental examples from early undergraduate teaching.

3.1. Fluid Dynamics

Early-years teaching of Fluid Dynamics often involves teaching of idealized fluid behaviour where the equations are simplified from the full Navier-Stokes governing equations. One example is Potential Flow Theory[28], which can capture a wide range of simple fluid behaviors but has key limitations for real flows. Part of teaching students this theory is also teaching them about its assumptions: and, more broadly, teaching them to interrogate the assumptions of all analytical or numerical methods in fluid dynamics. Lab experiments are commonly used to strengthen students' understanding of these assumptions. An example of a lab experiment for Potential Flow Theory is a small wind tunnel with a cylinder perpendicular to the flow, with pressure measurements taken around the diameter of the cylinder.

Since simulation tools can output pressure, they can replicate some learning objectives of such a lab, without the need for wind tunnel equipment. An [exercise in Jupyter Lab] was developed, using Ansys Fluent as the backend simulation tool. Ansys Fluent is an advanced, industry-standard simulation package which would usually be considered too complex to introduce to early-years students, however, using the PyFluent library within a Jupyter environment can hide Fluent itself, with its output pressure/velocity field data shown to the student directly in the Jupyter environment (Figure 1). Similarly, without the Fluent interface there is no need for students to learn the complexities of setting up a computational fluid dynamics problem at this stage. Instead, they are offered only one variable to manage – the input flow speed, controlled using a slider (Figure 2), allowing them to explore the impact of Reynolds number.

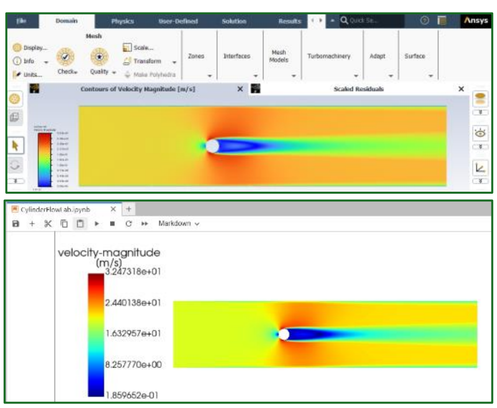

Figure 1: Velocity flow field displayed in Ansys Fluent (above) and Jupyter Lab (below)

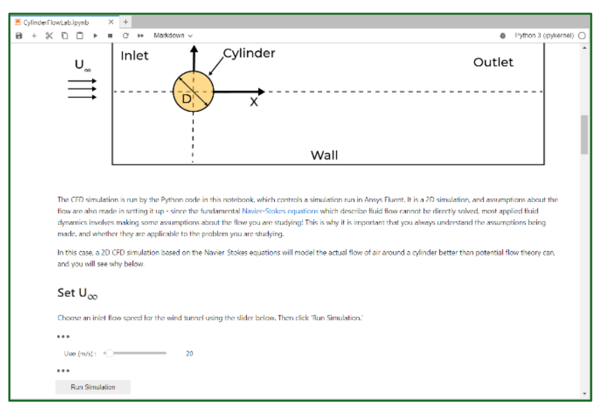

Figure 2: Jupyter Lab interface with simple button/slider controls to set values in Fluent and run the simulation

With a 2D steady-state simulation of a small problem, Ansys Fluent generates results within a minute or two on a standard laptop, reasonable for a lab session. The exercise then leads students through discussion of the outputs, as well some introductory explanation of what a computational fluid dynamics simulation entails and why there are also assumptions there to be interrogated.

3.2 Electrical/Electronics/Telecommunications

In the field of Electrical Engineering, a dipole antenna was identified as a typical fundamental problem in antenna-related courses, and an impedance matching network is another standard teaching problem. The combined exercise can also encourage deeper understanding by exploring the effects of varying dipole length, engaging critical analytical skills in electrical design. A Jupyter Notebook was again created, this time using the Python library PyAEDT to run Ansys HFSS and Circuit simulation tools.

The development of this resource was based on discussions with academics teaching this subject about emerging educational trends, particularly the use of innovative teaching methods including increased usage of programing in Electrical/Electronics engineering. Taking advantage of the Python libraries available with simulation tools allows students to gain coding skills and initial understanding of simulation software capabilities while learning engineering fundamentals.

Similar design guidelines were followed as for the Fluids resource, with explanatory images, code cells hidden by default, and plots displayed within the Jupyter Notebook (Figures 3 & 4).

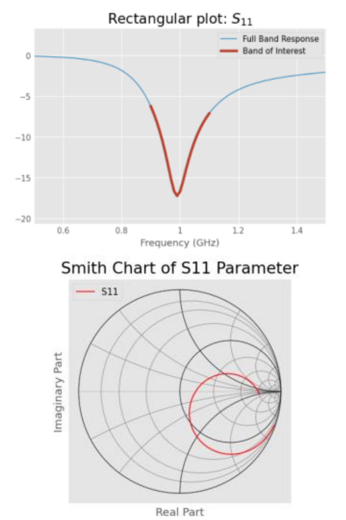

Figure 3: S11 parameter of the antenna, showing the frequency band of interest (upper) and its representation in the Smith Chart (lower), plotted using matplotlib

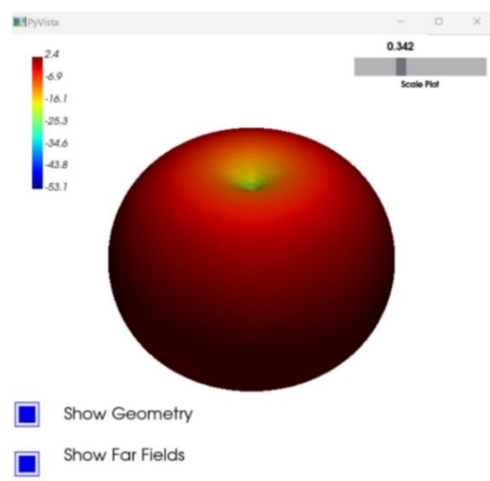

Figure 4: Gain of the dipole antenna plotted using PyVista, having the option to show/hide the dipole geometry and the radiated fields (also scalable)

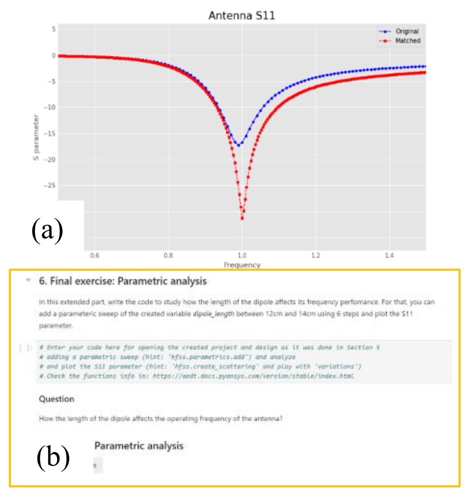

Figure 5: (a) Comparison of the S11 parameter with and without the matching network, from Ansys Circuit analysis (b) the final Notebook exercise, challenging students to write their own code, plus solution

The co-simulation with HFSS and Circuit is handled through Python to enable students to pair the antenna with a matching network and explore its response (Figure 5a). For this resource, industry readiness Ansys HFSS is opened by the code so students can observe their Python commands interacting with it, and stretch exercises are included where students engage with writing code themselves to extend the problem (Figure 5b).

4. A Python-based application for teaching in Biomedical Engineering courses:

The exercises in Section 3 are designed as simple introductions for engineering students to simulation concepts while exploring fundamental physics, with the assumption that later in their studies, students will become more familiar with simulation tools. But what about students who would benefit from the visualization aspects but are not expected or required to become simulation experts, such as biomedical engineers or medical students? For this audience, a simple view of simulation results could be sufficient to improve understanding.

To explore this concept, the Python-based Plotly Dash library was investigated for the creation of standalone applications which can drive simulation tools behind the scenes and display their results, together with additional information for students, in a self-contained interface. Combining Python tools such as this with the PyAnsys libraries gives opportunities to teach students with the power of simulation, while skipping the steep learning curve which such tools can have for students in engineering-adjacent subjects.

Recently, the Ansys Academic Program has tested this theory by creating an application with PyAnsys and the Dash library to show a simple fluid dynamics problem recontextualized for students in Healthcare-related degree programs: fluid flow through an artery (either simple or bifurcating) with various degrees of plaque buildup. The 'Ansys Academic Healthcare SimLab' application calls on the Ansys Discovery and Ansys Fluent tools in the background, for geometry definition and fluid dynamics simulation, but student users do not need to interact with the simulation tools. The application interface can include any theory or equations students may need to understand the simulations (Figure 7a), while all simulation inputs and results are contained within the app interface while the simulation tools run in the background (Figure 7b).

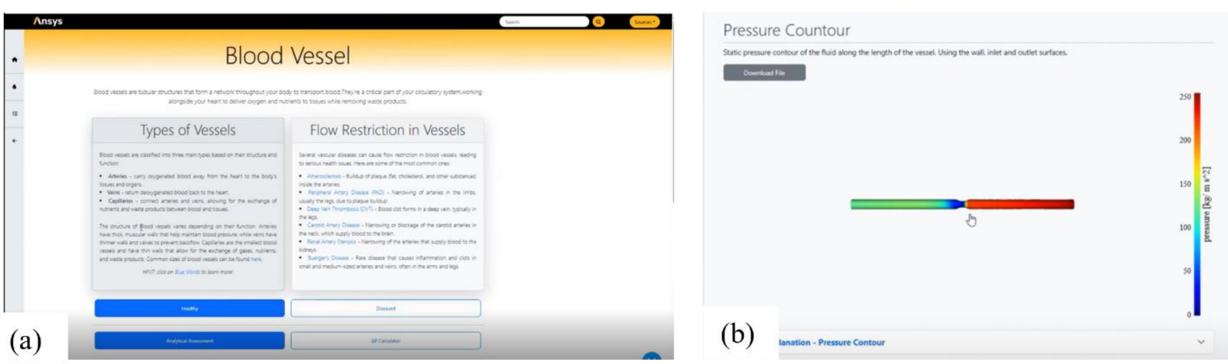

Figure 7: Screenshots of the Ansys Academic Healthcare SimLab application beta interface, showcasing (a) contextual background information and (b) simulation results with user-defined parameters

5. Initial feedback and outcomes:

Ansys Education Resources, including the Jupyter Notebook teaching resources discussed in Section 3, are available free to download from the Ansys website. As such, there is no traceability of which universities are using them or how, nor direct feedback from users on their experience. However, download analytics show that the first, Fluids-focused Notebook resource, released in February 2024, has now been downloaded over 300 times – sustaining an average of

20 downloads per month. The Electronics resource was released more recently in January 2025, so has yet to reach this level but is seeing around 10 downloads per month currently.

The Ansys Academic Healthcare SimLab application, meanwhile, is not yet publicly released but has gone through a thorough beta-testing process, with a total of ten academics at seven universities so far providing in-depth feedback on demonstrated function and usability of the app. The feedback to date has been positive, with comments such as “the information is well-balanced – not overwhelming” and “it’s an excellent tool for teaching hemodynamics”.

6. Next steps

The Jupyter Notebook education resources discussed in Section 3 seem useful and valued enough by the community that the Ansys Academic Program plans to make more available in future, possibly in other areas of fundamental physics, either created by Ansys directly or co-authored with academic users who wish to explore this area in their teaching.

The next steps for the Healthcare application are to release it to the academic community in 2025 and then get further feedback from its use in curriculum, with a view to expanding it to cover other core fundamental Healthcare topics around electronics or structural topics as well as the fluids examples. The potential for creation of further such applications, where easy access visualization could increase student understanding, are still being explored. One such area is Tensile Test labs, a common topic covered in mechanics of materials courses across multiple engineering disciplines. Currently an app is in development to allow students to simulate tensile tests, using various ASTM standards, to failure and compare simulation results to analytical data gathered during physical testing. The goal is to begin beta testing this application by end of 2025.

7. Conclusion

Based on trends from industry and literature, we believe there is a need for curriculum expansion to both strengthen students’ real-world understanding of theory and practice, and to extend their knowledge of technologies such as programming and simulation. The Ansys Education Resources presented in this paper form an exploration of combining simulation, programming and fundamental theoretical teaching in ways that could be deployed in first- and second-year undergraduate curricula in various Engineering or related courses, particularly as a complement to physical lab work.

8. References

[1] "Future of Jobs Report 2025," 2025. [Online]. Available: www.weforum.org

[2] B. Ray and R. Bhaskaran, "Integrating simulation into the engineering curriculum: a case study," *International Journal of Mechanical Engineering Education*, vol. 41, no. 3, pp. 269–280, 2013.

[3] F. Stern *et al.*, "Integration of Simulation Technology into Undergraduate Engineering Courses and Laboratories," in *2003 Annual Conference*, Citeseer, 2003, pp. 8–757.

[4] A. F. McKenna and A. R. Carberry, "Characterizing the role of modeling in innovation," *International Journal of Engineering Education*, vol. 28, no. 2, p. 263, 2012.

[5] S. Emmott and S. Rison, "Towards 2020 science," *Science in Parliament*, vol. 65, no. 4, pp. 31–33, 2008.

[6] "Criteria for Accrediting Engineering Programs, 2025 - 2026 - ABET." Accessed: Jan. 14, 2025. [Online]. Available: https://www.abet.org/accreditation/accreditation-criteria/criteria-for-accrediting-engineering-programs-2025-2026/

[7] E. Riese and S. Stenbom, "Engineering Students' Experiences of Assessment in Introductory Computer Science Courses," *IEEE Transactions on Education*, vol. 66, no. 4, pp. 350–359, 2023.

[8] A. Forte and M. Guzdial, "Motivation and nonmajors in computer science: identifying discrete audiences for introductory courses," *IEEE Transactions on Education*, vol. 48, no. 2, pp. 248–253, 2005.

[9] J. Q. Dawson, M. Allen, A. Campbell, and A. Valair, "Designing an introductory programming course to improve non-majors' experiences," in *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, 2018, pp. 26–31.

[10] L. J. De Vin and M. Jagstam, "Why we need to offer a modeling and simulation engineering curriculum," in *Proceeding of the 2001 Winter Simulation Conference (Cat. No. 01CH37304)*, IEEE, 2001, pp. 1599–1604.

[11] D. I. Spang, "Curriculum design and assessment to address the industry skills gap," in *2014 ASEE Annual Conference & Exposition*, 2014, pp. 24–345.

[12] A. J. Magana, "Modeling and simulation in engineering education: A learning progression," *Journal of Professional Issues in Engineering Education and Practice*, vol. 143, no. 4, p. 04017008, 2017.

[13] IBM, "What is Industry 4.0?" Accessed: Jan. 14, 2025. [Online]. Available: https://www.ibm.com/think/topics/industry-4-0

[14] E. Oztemel and S. Gursev, "Literature review of Industry 4.0 and related technologies," *J Intell Manuf*, vol. 31, no. 1, pp. 127–182, 2020.

[15] A. Kaszynski, "Pyansys: Python interface to MAPDL and associated binary and ASCII files," *Zenodo. doi*, vol. 10, 2020.

[16] P. Chaudhari, J. Najmon, and A. Tovar, "Efficient Design of Shell-and-Tube Heat Exchangers Using CAD Automation and Fluid flow Analysis in a Multi-Objective Bayesian Optimization Framework," SAE Technical Paper, 2024.

[17] T. Meier *et al.*, “Obtaining auxetic and isotropic metamaterials in counterintuitive design spaces: an automated optimization approach and experimental characterization,” *NPJ Comput Mater*, vol. 10, no. 1, p. 3, 2024.

[18] T. N. Phan, J. J. Aranda, B. Oelmann, and S. Bader, “Design optimization and comparison of cylindrical electromagnetic vibration energy harvesters,” *Sensors*, vol. 21, no. 23, p. 7985, 2021.

[19] Ansys, “Grad Student Applies PyFluent and Machine Learning to Advance Sustainable Materials Processing.” Accessed: Apr. 30, 2025. [Online]. Available: https://www.ansys.com/blog/student-uses-pyfluent-ml-for-sustainable-materials-processing

[20] A. Twissell, “Modelling and simulating electronics knowledge: Conceptual understanding and learning through active agency,” *J Educ Techno Soc*, vol. 21, no. 2, pp. 111–123, 2018.

[21] O. A. Fadiran, J. Van Biljon, and M. A. Schoeman, “How can visualisation principles be used to support knowledge transfer in teaching and learning?,” in *2018 Conference on Information Communications Technology and Society (ICTAS)*, IEEE, 2018, pp. 1–6.

[22] M. Olsson, P. Mozelius, and J. Collin, “Visualisation and gamification of e-Learning and programming education,” *Electronic journal of e-learning*, vol. 13, no. 6, pp. pp452-465, 2015.

[23] A. Twissell, “Visualisation in applied learning contexts: a review,” *J Educ Techno Soc*, vol. 17, no. 3, pp. 180–191, 2014.

[24] V. M. Rossi, “Simulation led optical design assessments: Emphasizing practical and computational considerations in an upper division physics lecture course,” *Am J Phys*, vol. 90, no. 4, pp. 279–285, Apr. 2022, doi: 10.1119/5.0064138.

[25] S. COOKE, S. COLEMAN, and J. DERRICK, “Exploring The Potential For Scripting With Simulation In Engineering Education–Practical Examples Using Python And Ansys,” 2023.

[26] J. W. Johnson, “Benefits and pitfalls of jupyter notebooks in the classroom,” in *Proceedings of the 21st annual conference on information technology education*, 2020, pp. 32–37.

[27] R. Castilla and M. Peña, “Jupyter Notebooks for the study of advanced topics in Fluid Mechanics,” *Computer Applications in Engineering Education*, vol. 31, no. 4, pp. 1001–1013, 2023.

[28] B. S. Massey and J. Ward-Smith, *Mechanics of Fluids, Ninth edition*. CRC Press, 2012. doi: 10.1201/9781315272542/MECHANICS-FLUIDS-JOHN-WARD-SMITH.

[29] R. Lombardo, “Python and Plotly Dash, A Quick and Convenient Way to Develop Web Apps for Teaching Physical Chemistry Models,” *J Chem Educ*, vol. 101, no. 11, pp. 4661–4670, Nov. 2024, doi: 10.1021/acs.jchemed.3c01167.