INTEGRATING COMPUTATION THROUGH FIRST- AND SECOND-YEAR PHYSICS MAJOR UNITS

Istvan Laszlo[a], and Anna McLean Phillips[a]

Presenting Author: Istvan Laszlo (Istvan.Laszlo@monash.edu)
[a]School of Physics and Astronomy, Monash University, Clayton, VIC 3150, Australia

KEYWORDS: computation, undergraduate physics, python

COMPUTATION AS A TARGET FOR PHYSICS INSTRUCTION

Computation is increasingly recognized as a core aspect of physics practice and target for physics instruction. Literature on computation in physics often focuses on separate computational physics units or courses (see Atherton, 2023, for a review) rather than integrating computation throughout curricula. We document the process of integrating computation in the context of first- and second-year physics units at Monash University to provide a model for embedding computation throughout a three-year Australian Bachelor of Science in physics.

GOALS FOR COMPUTATIONAL PHYSICS INSTRUCTION AT MONASH

Naturally, computational physics instruction aims to develop a solid foundation of the coding skills that students will require in their careers and to develop literacy with a chosen language. However, we set out more importantly and broadly, to develop the skills needed to break down a task into its separate steps and develop an algorithm (free of syntax) and to develop student's ability to create interactive visualisations and models, in order to augment their understanding of various physical phenomena. This approach enhances engagement by showcasing the benefits of computational approaches and augments learning through linking interesting phenomena and the coding process.

IMPLEMENTATION OF COMPUTATIONAL ACTIVITIES IN APPLIED AND LABORATORY ACTIVITIES

Our vertical integration of computational skills begins in first year, where primarily we focus on developing coding skills in support of laboratory data analysis. Here, the activities begin with a basic introduction to coding in the chosen language (originally the Wolfram Language in Mathematica, and currently Python) tailored to lab analysis (plotting, fitting, error analysis). Subsequent tasks build upon this, supported by a mix of examples and exercises that ask students to follow along, modify previously used code, or fill in gaps in templates for each laboratory they complete. This approach is followed in second year with the applications broadened tremendously into fortnightly sessions that serve to create fully interactive demonstrations/visualisations of the physics discussed in other aspects of the unit. These include creating time evolutions of wavefunctions for the common quantum tunnelling problems, visualizations of random walks, manipulable ray traces, and much more. Through this, students not only learn a variety of coding techniques and principles but see very interesting and often hard to imagine aspects of physics come to life through their code!

EVALUATING THE SUCCESS OF COMPUTATIONAL INSTRUCTION

We assess our success in achieving both computational and physics learning outcomes, as well as enhance engagement, based on feedback provided annually by students in their evaluations of teaching units, on its follow through in higher year level units, and on the difference in achievement in the unit as a whole for years/students where computational applied sessions have been attended, in comparison to those cases where applied sessions were not offered or not attended.

REFERENCE

Atherton, T. J. (2023). Resource Letter CP-3: Computational physics. *American Journal of Physics*, *91*(1), 7-27.

Proceedings of the Australian Conference on Science and Mathematics Education, The University of Tasmania, 30 August – 1 September 2023, page 46, ISSN 2653-0481.