Integration of Computational Thinking into Undergraduate Engineering Education: A Case Study on Design of a Hydraulic Course

Hao Chen, Qi Cao, Christian Della, Li Hong Idris Lim
University of Glasgow, Singapore Campus

Abstract

Computing techniques have been an essential component of engineering education. Rapid development of computing science (CS) has provided a powerful means of modelling real-world physical systems through computer simulations, data processing, data analytics, and data visualizations. This indicates that students must be prepared to use CS methods and applications as a part of their fundamental education. It is the responsibility of colleges and universities to investigate how to equip engineering students with competency in computational thinking (CT) and incorporate contemporary computing fundamental knowledge into their academic curriculum. It aims to improve the professional qualifications of their engineering graduates.

It is argued that computational thinking skills are best trained in the domain-specific and personal relevant contexts. By explicitly integrating computing concepts into classroom teaching and problem solving of the respective disciplines, the engineering graduates will enter the workforce with improved and practice-ready computational thinking. It will enhance their problem-solving and design skills. Majority of studies on the discipline-based computing have been focused on the use of computer simulations to improve the concept learning. In most of the cases, the simulation software is used as "black box" in classroom teaching, which is merely used to generate outputs based on the inputs provided by the students. However, students may need to have more access to the software they are using, in order to understand the detailed underlying mechanism. It can be effectively achieved by building simulation models instead of just using them. Furthermore, through building a simulation model, computational thinking components, e.g., algorithmic design and pattern recognition are also incorporated via the programming process. In this way, students are well-motivated to learn programming through coursework. Inclusion of computer programming is also helpful to improve the students' skills in logical reasoning, computational thinking, and quantitative problem-solving.

In this paper, we will present a case study on design of a hydraulic coursework, which effectively combines the assessment of programming skills, modelling and simulations and disciplinary knowledge for a real-world hydraulic problem. The specific objectives of this study are to investigate if it can help enhance the students' computational thinking skills, acquisitions of fundamental computing concepts, as well as procedures.

The Civil engineering coursework requires students to use Python programming language to develop a tool. In the freshman year, a foundation programming subject, CVE1113 - Civil Engineering Skills Python Programming has been taught serving as an introduction to the engineering profession and training in some of the skills needed for professional success. In Year 2, the Civil Engineering subject, CVE2141 - Hydraulics and Hydrology is given on the fundamentals of open channel flow and the surface hydrology. The coursework of this subject in this paper is designed with the objectives of this coursework as follows.

1) Disciplinary objective: apply direct step method to gradually-varied open channel flow.
2) Objective on programming: apply object-oriented programming style to solve fundamental engineering problems.
3) Objective on modelling and simulations: explain the verification and validation process for a numerical solver, and understand the its limitation.

This newly designed coursework is being implemented in AY2021/22 to SIT-UoG Joint Degree on Civil Engineering. Students' evaluation and survey will be conducted by the end of the trimester. The evaluation questions will be specifically designed to reflect the effectiveness of this coursework on improving their skills in programming and computational thinking, and the challenges encountered when students make attempts to complete the coursework. Through this case study, some insights can be provided into design of such coursework as alternative assessment in engineering modules, and students' altitudes towards it.

References

Magana, A. J., & Silva Coutinho, G. (2017). Modeling and simulation practices for a computational thinking-enabled engineering workforce. *Computer Applications in Engineering Education*, *25*(1), 62-78.

Khashe, S., Gerber, D. J., & Smith, I. F. (2016). Surveying the evolution of computing in architecture, engineering, and construction education since 2012. *Journal of Computing in Civil Engineering*, *30*(6), 04016017.

Vergara, C. E., Urban-Lurain, M., Dresen, C., Coxen, T., MacFarlane, T., Frazier, K., ... & Wolff, T. F. (2009, October). Aligning computing education with engineering workforce computational needs: New curricular directions to improve computational thinking in engineering graduates. In *2009 39th IEEE Frontiers in Education Conference* (pp. 1-6). IEEE.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33-35.