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

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

Introduction and Aim

- 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, etc.
- It is the responsibility of universities to investigate how to equip engineering students with competency in computational thinking[1] and incorporate contemporary computing fundamental knowledge into their academic curriculum[2].
- Computational thinking skills are best trained in domain-specific and personal relevant contexts[3]. By explicitly integrating computational concepts into classroom teaching and problem solving of the respective disciplines, engineering graduates will enter the workforce with improved and practice-ready skills in computational thinking and problem-solving.

We aim to investigate the feasibility of integrating the computational thinking into an engineering coursework. Students' perception in terms of learning experience and level of difficulty will be also analyzed.

Methodology

Context and Participants

- The present coursework is designed for CVE2141 - Hydraulics and Hydrology in our civil engineering program. There are totally 106 participants for this course in AY2021/22.
- Python programming has been taught in CVE1113 - Civil Engineering Skills for six weeks in AY2020/21.
- Fundamentals of fluid mechanics have been taught in CVE 1241 - Fluid Mechanics in AY2020/21.

Coursework Design

Tasks in the coursework:
- Implement the numerical solver based on the given Python programming template.
- Work on code verification by ensuring there are no bugs and mistakes in their code; Solution verification by ensuring the solution is converged.
- Apply the solver in a case study on backflow surface profile, and present the results in a report.

Objectives of the coursework:
- Disciplinary objective: Explain the direct step method and apply it to gradually varied flow.
- Objective on programming: Apply Python object-oriented programming skills to solve fundamental engineering problem.
- Objective on modelling and simulation: Explain the verification and validation process for a numerical solver and understand the limitations of a numerical solver.

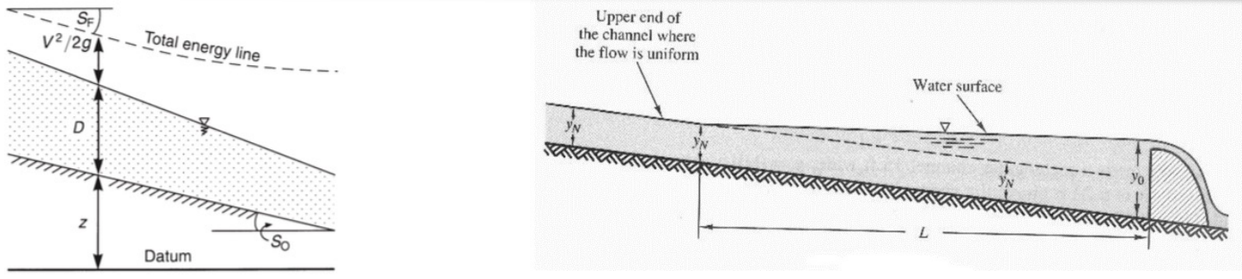


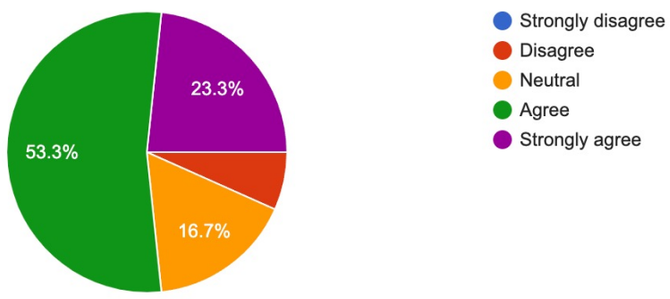
Fig. 1 (a): Definition sketch of a gradually varied open channel flow. (b): Case study in the hydraulic coursework.

Evaluation Results

Prerequisites for the Coursework

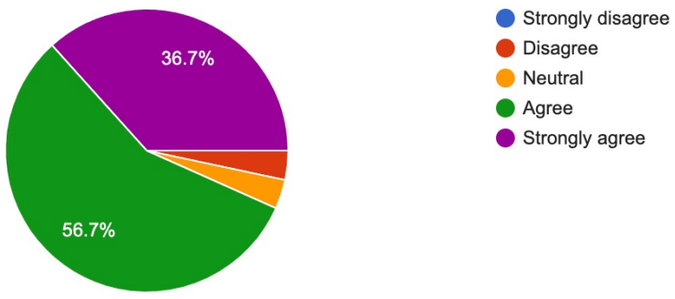
I am able to apply the programming skills I learned in Civil Engineering Skills into this project.

30 responses



I understand how to do hand calculations on direct step method before I started to work on the project.

30 responses



Student Learning Experience

- Students generally enjoy working on the coursework.
- The Python programming part can be challenging to some of the students.
- More consultation sessions are needed especially in face-to-face to help students debug the code.

"The coursework was fun and helpful for understanding of direct step method."

"Would be beneficial if students are able to interact closely with the lecturer (F2F) with the coursework and more assistance and guidance would be better."

"I really appreciate the idea of combining 2 modules together as it helps to think beyond the study. It also helped me to recall the module learnt the year before to be used rather than left behind. It was pleasant learning experience despite the hiccups here and there while coding."

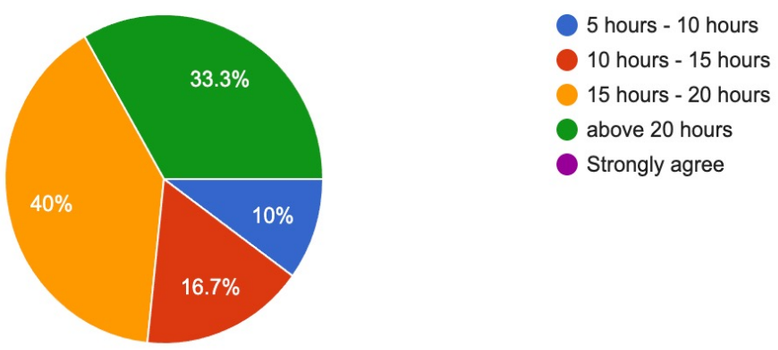
"Personally, it would be great for coursework that is non-examinable but the duration to submit the report is rather short, in line with other module submission, makes it hard to do it a proper and formal report. At the same time, it would be great for increased consultation in the coding aspect rather than the case study part. Although the consultations are great to ask module related tutorial uncertainties, limited coverage project makes it hard to understand what were the requirements initially. At the same time, if everyone codes will turn out to be the same, it would be better as a group coursework rather than individual. It is in fact not easy to code a whole programme individually."

"The coursework was fun and helpful for the understanding of direct step method"

Workload of the Coursework

How many hours do you need totally (off class) for this coursework?

30 responses



Conclusions

- This case study demonstrates the feasibility of integrating computational thinking into a disciplinary student coursework via Python programming.
- Most of the students are able to implement the solver, with the template and the guidance provided.
- The workload depends on the students' background and prerequisite skills on programming.
- In the future, more consultation sessions will be provided to improve student learning experience.

References

1. Wing, 2006, "Computational thinking". Communications of the ACM, 49(3), 33-35.

2. Vergara et al., 2009, "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, IEEE.

3. Magana et al, 2017, "Modelling and simulation practices for a computational thinking-enabled engineering workforce", Computer Applications in Engineering Education, 25(1), 62-78.