ResearchOnline@JCU

JAMES COOK UNIVERSITY AUSTRALIA

This is the author-created version of the following work:

Arjmandi, Mohammadreza, Woo, Meng Wai, Mankelow, Cody, Loho, Thomas, Shahbaz, Kaveh, Auckaili, Amar, and Thambyah, Ashvin (2023) ***Embedding computer programming into a chemical engineering course: The impact on experiential learning.*** **Education for Chemical Engineers, 43 pp. 50-57.**

Access to this file is available from:

https://researchonline.jcu.edu.au/86782/

© 2023 Institution of Chemical Engineers. Published by Elsevier Ltd. All rights reserved. AAM available under a CC BY-NC-ND license.

Please refer to the original source for the final version of this work:

https://doi.org/10.1016/j.ece.2023.01.008

Embedding computer programming into a chemical engineering course: The impact on experiential learning

Mohammadreza Arjmandi[1,*], Meng Wai Woo[1], Cody Mankelow[2], Thomas Loho[1], Kaveh Shahbaz[1], Amar Auckaili[1], Ashvin Thambyah[1]

[1] Department of Chemical and Materials Engineering, University of Auckland, Auckland, 1010, New Zealand
[2] Department of Civil and Environmental Engineering, University of Auckland, Auckland, 1010, New Zealand

* Corresponding author: M. Arjmandi, r.arjmandi@auckland.ac.nz

Abstract. The need for autonomous engineering graduates who demonstrate hands-on skills has increased in the industry. Computer programming helps engineering students solve real-world problems systematically and accurately by applying governing physical and mathematical models into a format that a computer can read and execute. This study describes the pedagogical approach of incorporating programming workshops and assessments into a second-year chemical engineering course. The impact of this intervention on experiential learning amongst the students was then evaluated by analysing the feedback provided by voluntary participants during several focus group sessions. The feedback gave further insight into teaching pedagogy with respect to Kolb`s experiential learning cycle. It was found the programming background of an individual clearly affects the phase of the learning cycle they predominantly experience during the workshops. Furthermore, programming background affected an individual's critical thinking while approaching an engineering problem. Constructive feedback provided by the student participants offered an invaluable opportunity for the teaching team to reflect on what went well and the areas for improvement in future iterations. The findings of this study can advance knowledge around design and implementation of a programming module within an engineering course.

***Keywords*:** Engineering education; qualitative study; programming with MATLAB; problem-based learning; student experience;

1. INTRODUCTION

The need to learn computer programming is well understood in some engineering disciplines, such as software engineering, electrical, mechatronics, and similar engineering disciplines. individuals need to develop computer programmes to interact with software/hardware. However, the application of programming in solving problems is less evident to those studying other disciplines such as mechanical, civil, chemical and materials engineering. Moreover, the solution to engineering problems often requires iterations to achieve the best solution [1]. In most cases, finding an analytical solution to an ill-structured engineering problem is quite expensive, if not impossible, in which numerical methods might significantly assist [2]. To implement numerical solutions, one should utilise programming to translate the mathematical and physical models into a computer language and benefit from the high processing power of modern computers. Programming is helpful in problem-solving, decision-making, implementation of numerical methods, optimisation, visualisation, and analysis of engineering problems [3, 4]; and programming skills are essential for engagement with modern engineering technologies. Furthermore, programming allows engineering graduates to communicate more effectively with other experts [5].

On the other hand, the integration of a programming component into an existing course is challenging. The learner`s perception of programming and their mindset might affect the learning process. Programming is not a single skill, and to enhance programming skills, one needs to strengthen the complex cognitive set of skills [6]. Some chemical engineering undergraduate students have indicated they choose the discipline to avoid programming, arguing that if they develop their software skills, such as Microsoft Excel, they would be more equipped to work in the industry than programming. Coronell [7] opposed this notion and argued that by programming the student applies a systematic problem-solving approach. Previously it was found the relevance

of programming to the discipline must be clarified to turn the learner`s attention towards its significance and promote a more expansive application [6, 8]. Student motivation is, therefore, a crucial stage in the learning process.

Furthermore, programming within the context of problem-solving is difficult as it involves a multistage process. It involves understanding the problem and unknown parameters, developing of a solution strategy through algorithms, knowledge of a programming language, its syntax, and tools, debugging the code, and finally, verifying the results. Therefore, if the problem-solving tool (programming) is too complex, the student must allocate more time and resources to learn it than to think about how to approach the problem. In this regard, Cazzola and Olivares [9] proposed and implemented a gradual learning programming scheme so that the students were introduced to new concepts one at a time. Feedback from students revealed that gradual development was helpful for less experienced students by allowing them to focus more on the problem than the programming, students. However, those with pre-existing programming knowledge were disappointed or annoyed by the approach as it limited their capacity.

Industry 4.0, digitisation of the manufacturing industry, has led to the advancement of computational technologies such as artificial intelligence, digital twins, and data analytics. Interdisciplinary specialities are on the rise and there is a significant demand globally for well-trained engineers. One essential literacy that we recognise is lacking in the current chemical engineering courses is that of computer programming with relevance to real-world chemical engineering applications. Previous studies found programming as one of the essential skills needed in Industry 4.0 [10-12]. Previous studies have implied a lack of programming in the chemical engineering curriculum [13], and some researchers have introduced programming to the curriculum either as a course or a module within a course [7, 10]. The traditional lecture-style pedagogy to teach programming was found ineffective, while the interactive framework which encourages students from passive learners to active learners, has resulted in increased engagement and higher satisfaction among the student cohort [14].

We aim to provide new opportunities for our undergraduates to gain more exposure to programming throughout their studies. Furthermore, solving real-world engineering problems is recognised as an essential aspect of engineering training [15, 16]. To embark on this initiative, we have introduced programming workshops and designed assignments requiring MATLAB programming. By participating in programming workshops and carrying out exercises and assignments, the students are given opportunities to enhance their programming skills and deal with real-world chemical engineering problems that require both theoretical knowledge and computer modelling.

To this end, the integration of programming and evaluating the impact on active learning amongst students are still to be explored. This study evaluates an initiative of embedding programming into a second-year chemical engineering course through qualitative analysis of student feedback collected through focus groups conducted in two consecutive iterations of the course, and with respect to Kolb`s experiential theory [17]. We will discuss the feedback and then assess our hypothesis on whether embedding programming into the course would facilitate the active experimentation element of Kolb`s experiential learning, i.e., application to a real-life situation, providing a better learning experience for the student cohort. In addition, we will also aim to assess if the interest level of the participants, their academic background and the MATLAB workshops` implementation have affected the assessment of the hypothesis above.

2. MATERIALS AND METHODS

2.1 Teaching pedagogy

A programming module was initially introduced in semester two (S2), 2019 to the course *CHEMMAT203- Process Engineering 3: Transfer Processes,* a second-year core course in the Bachelor of Chemical and Materials Engineering (Honours) programme at the University of Auckland, New Zealand. Fundamental topics of Fluid Mechanics and Heat Transfer are taught in this course, including fluid properties and hydrostatics, fluid motion, steady-state and transient heat conduction, heat convection, and the design of heat exchangers. Approximately 80 students enrolled in the course being run by three lecturers. Weekly programming workshops were introduced and was delivered by an additional lecturer to facilitate learning through hands-on experience of the taught concepts and introducing numerical solutions to deal with complex real-world and ill-structured problems within the context of the above topics. Course syllabus and topics for programming workshops are summarised in Tale 1. The learning outcome of the programming workshops is to utilise numerical techniques to solve complex

problems related to Fluid Mechanics and heat transfer. MATLAB® (MathWorks) was selected as a programming platform, due to student`s pre-existing knowledge and as it is widely used in the industry. An introductory session to programming with MATLAB and programme debugging was held, focusing on practical tips and tricks to refresh and upskill the students.

Next, some real-world problems related to heat transfer and fluid mechanics topics were outlined in the following workshops, which were either too difficult to solve using conventional analytical formulae or required a significant level of simplification affecting solution accuracy, and often a numerical approach deemed resulting in more accurate results. The students were taught how the problem could be then modelled using mathematical and numerical formulations and how to implement the model into a computer programme to obtain results. Each workshop come with some worked examples and non-compulsory take-home exercises, with the support provided for those who have attempted the exercises.

Later, in Semester 2, 2020, the content for workshops was updated to better align with the programming assignments to facilitate problem-based learning (PBL). The introductory lecture for each session was pre-recorded and was provided to the students prior to weekly workshops to allow more hands-on time for the students tackling the assignment items. In each workshop, following a short introduction to the topic and the learning objectives by the instructor, the students were supposed to carry out the assignment tasks by following the workshop manual for the session.

To assess the learning objectives, as well as engaging the students with the programming module, an individual programming assignment (Fluid Mechanics) and a summative group programming assignment (Heat Conduction) were designed and assigned to the students during S2, 2019. To ensure a fair contribution, the students evaluated their teammates through an anonymous peer-marking and those who underperformed in the team activities were penalised in their individual assignment mark. Peer-marking was compulsory and was achieved through filling an online form submitted by everyone for all their teammates. Those who were identified as underperforming individuals were given a chance to justify their lower contribution. In S2 2020, the Heat Conduction assignment was modified into an individual assignment. This was following observations in 2019 in which not all student team members might have a chance to actively participate in the coding-based activities of the assignment. The outlines for assignments and the tasks and analyses that the students should carry out are illustrated in Figure 1 and Figure 2. We assessed steady-state and distributed transient heat conduction topics in Heat Conduction assignments. Although solving the heat equation as a Partial Differential Equation seems challenging for the second-year students, one-dimensional transient heat conduction and the use of Heisler charts were covered in lectures and to achieve a cohesion between the lectures and workshops, a numerical approach was implemented in the programming workshop to solve the heat equation. Concerning the Fluid Mechanics part, the assignments were focused on using numerical methods to solve the non-linear Colebrook equation to find the Darcy friction factor for a viscous flow in a piping system considering major and minor energy losses. The assignments carried 20% of the total mark for the course. Lecturer`s feedback to individual submissions was provided prior to the following assignment. We believe that the assignments along with the take-home exercises embedded in the notes offered the students an opportunity to repeat the simulations through programming and further reflect on their hands-on observation to fulfil the Reflective Observation phase of Kolb`s model.

2.2 Focus groups

To conduct focus groups from voluntary student participants, an ethics application was submitted to and was approved by The University of Auckland Human Participants Ethics Committee for three years (Ref. #024409). Next, the researchers advertised and arranged the focus groups. Data were collected and analysed from the audio recording of the participants with the recordings transcribed by the authors in this manuscript (no external transcription). Each session lasted approximately an hour, conducted by an academic staff (as moderator) who was not involved in teaching the course. The questions (attached as supplementary information) were developed as part of ethics approval process to test our hypothesis outlined previously and to gain as much information as possible regarding the student`s learning experience and their feedback associated with the programming component of the course. The participants were randomly placed into focus groups. The participants signed a consent form where participants were reminded of their rights to withdraw from participation at any time prior to the conclusion of the session without reasoning. The moderator firstly posed the question (which will be discussed in the results section) to the group and allowed each participant to provide their response so that everyone had a chance to respond to the question without interruption from the rest of the participants. The moderator then

allowed the group to discuss the question further and respond to each other where necessary. This structure was followed for each question. To analyse the qualitative data towards answering the research questions raised in this study, as part of the analysis, we also classified the responses from the participants based on their performance as listed in Table 2. It should be noted that the small number of participants was due to the voluntary nature of focus group participation and running the sessions in the following semester to prevent a misconception that participation might affect the individual`s mark. Therefore, the authors expected a low student turn up. However, the participants with various performance levels representing the whole class attended the sessions, as shown in Table 2.

3. RESULTS

In this section, the responses provided by the participants during the focus group sessions will be presented. The complete set of focus group questions is provided as supplementary information. In the discussion here, alphabets in bracket (A, B, C or D) classify the participant corresponding to Table 2.

3.1 The effect of programming workshops on learner`s understanding of taught concepts

The responses from the majority of the 2019 cohort indicated that overall, the programming workshops had enhanced the student`s understanding of the fluid dynamics and heat transfer topics taught in lectures. For example, one student (B) replied *"It helped us solve the problem by separating it piece by piece so that we can understand it thoroughly."* and another participant (B) replied: *"we were able to use the theory that we learned to apply to different types of questions, rather than just doing it by calculations on paper, we can think about it more, and it can be used in the real world."* The relevance of programming to the Chemical Engineering specialisation was acknowledged by a participant (A) in a comment: *"We did study all coding principles in the first year, but to be able to implement it in Chemical and Materials engineering part of it, that was very useful because, in the first year, it was very generic."* However, some participants from the 2020 cohort (B) found that the workshops did not increase their knowledge of heat transfer or fluid mechanics, although they found the workshops helpful as it allowed them to enhance further their programming skills. A participant (A) found programming helped to understand heat transfer content deeper and suggested programming assignments be expanded to their 3rd-year courses: *"I liked the technical part of it… if the teaching team really wants us to learn that scope, they should include the MATLAB exercise for this semester, as well next semester [3rd-year courses]."*

3.2 Pre-existing programming skills enhanced through workshops

Participants from the 2019 cohort responded that the workshops enhanced their overall learning experience. The relevance to chemical engineering was highlighted in the comments once again. For a participant (A), workshops enhanced the programming skills as they had to repeat the process: "*In the first year when I was learning MATLAB, I wasn't very good at it, and then doing it again in the second year enhanced it. I sort of get to understand that, especially because we had a step-by-step workshop.*" Concerning the group assignment, one participant (C) replied: "*The way our group was structured, we assigned the roles to each other, so one does the programming, others write the report, and are not much involved in coding.*" Similar comments from other participants revealed that the students should have received additional guides on working together for a group programming assessment. Some students expressed the logistics issues adversely affected their learning experience in the workshops, e.g., they found it not easy to concentrate in the computer lab as other students were occasionally entering or leaving the room. Furthermore, due to the large size of the computer labs, sometimes it was hard to hear the instructor clearly. The participants from the 2020 cohort responded that the workshops were a good learning experience. A participant (B) replied: "*Actual programming part was not too hard, which is really nice to see that you can bring along some of your programming skills, even if they are not a great programmer. But you could still apply the basic knowledge to the tasks that we are looking at. So yeah, useful*!" According to feedback, it was found that in the 2020 iteration, the workshops were conducted with significantly fewer logistical issues.

3.3 Cohesion between coursework and the programming workshops

The participants responded that they felt that overall, there was a relatively good level of cohesion between the workshop learning materials and the assignments tasks. One participant highlighted the link between the lectures

and programming workshops: *"Our lecturers mentioned specifically, "this is the part where you must apply [iterative solution] in MATLAB. This is what you will learn in the following workshop." So, the lectures really knew what will be thought in the workshops..."* However, the participants did not find a clear link between the workshops and experimental work in the labs: *"I found the lectures and the workshops were useful for the assignments, but there was not any correlation between the labs and the assignment."* The same question was posed to the 2020 cohort, allowing them to reflect on their learning experiences throughout the semester. A participant shared their own experience to put it into context: *"...I found that quite useful because I sort of remembered that lecture and then, when it came actually to do it myself, I found it very useful to go over lectures."* A student (A) indicated they found programming workshops helpful in the lab report writing: *"The programming assignments helped us in calculations for the lab"* and suggested having programming lectures in the regular lecture hours of the course. However, another participant (B) opposed this idea: *"...I think MATLAB programming during the lecture might not work, because you do want to learn it yourself. So, I disagree with having MATLAB in the lectures because it is NOT as hands-on for the students."*

3.4 Contribution towards student`s performance in the final exam

Concerning the impact of the programming workshops on the individual`s performance in the final exam, almost all participants indicated they found little to no direct contribution of programming in their final exams. Nevertheless, the facilitator brought up whether the workshops might improve the participants' conceptual understanding and if that might indirectly help the individual`s performance in the exam. Upon further reflections by one participant (A): *"It actually had an excellent effect on my conceptualisation of the way heat transfer actually happens. I would say that I did follow through into the final exam, and potentially (not definitely), one of the reasons I was more comfortable with heat transfer at the final exam than the Fluid Mechanics section was because this assignment had ingrained a better understanding."* Furthermore, the discussion items in the heat conduction assignment contributed towards deeper understanding as expressed by another participant (A): *"... When I did the understanding part, where I had to explain it through... yes, that definitely did develop my understanding of it. It is somewhat similar to the exam questions, just without programming...."* Interestingly, the 2020 cohort felt the programming workshops contributed more towards the heat conduction questions in the exam, whereas there was no indication on its contribution towards the Fluid Mechanics module: *"MATLAB helped with the heat transfer part. I also enjoyed doing the coding exercises... I think having the assignment was good for the exam as practice."*

3.5 The impact of active experimentation on learning experience

The participants then shared their thoughts on whether active experimentation through programming led to a deeper understanding of the physical concepts taught in the lectures. One participant (B) replied that the code developed by their team let them change input, and attempt explaining the trends observed by referring to the theories they had learned in the class. However, the team was not confident with the results. This was due to comparison of their results with those of another team. Nevertheless, all participants from group A had different opinions and according to one: *"The interpretation part of group assignment helped me to actually understand what is going on, and what is the physical meaning behind it...I did go back and see what those numbers actually meant...."* The participants from the 2020 cohort agreed the workshops were helpful in that regard: "*You would have to do the MATLAB programming, and then you must explain your answer and the iterations, so what happened [was] we had to search for 'why'. Then, we referred to the notes, which is good, so it is programming [together with] discussion from the notes. In the end, it turned out to be useful because it helped me to understand the connection between [concepts]...."*

3.6 Self-reflection

Self-reflection on learning and the extent to which the workshops provided reflection opportunities for the participants were discussed next. While participants from group B believed the workshop did not provide many opportunities to reflect on the learning experience due to limited resources (teaching staff, time constraint, etc.), group A participants offered additional insight. The participants argued that the assignments tasks allowed learning reflection to some extent. Some workshop materials, e.g., MATLAB programming lecture slides, made them reflect on their learning experience from 1st year programming course and the extent to which they had obtained programming skills. The 2020 cohort felt the workshops themselves did not help much in terms of

reflection. Still, the feedback provided by the instructor on their submissions contributed to reflecting the extent to which an individual fulfilled the learning objectives: "*I do not think the workshop itself helped us reflect…Once we got our marks back, and by reviewing the instructor`s comments [we reflected on our learning].*"

The next set of questions tested the potential effect of student background and implementation approach on the evaluation of our hypothesis.

3.7 On increasing student engagement

Workshop activities such as real-world case studies, quizzes, or practical exercises that the students can participate in, and the impact on student engagement was discussed. Participants classified in group C suggested including more examples and more explicit instructions to help the students with the assignments, which might result in deeper learning for them. In contrast, one participant from group A believed fewer materials to be presented and more time should be allocated to the students in the workshop to experiment with code writing and tweaking the scripts by themselves, implying an extra focus on active experimentation. The participants indicated a higher student engagement would have achieved, if the instructor had interacted more frequently with the class. The response from the 2020 cohort revealed self-motivation has a significant impact on the engagement. According to a participant from group B: *"I was not interested in programming, so I'm self-motivated to [learn] that. I want to be able to do it, that is sort of a test of my own ability and understanding. I was motivated to do it because of this [challenge], so I enjoyed it in the end and [was] just self-motivated to do it from my previous history of coding."* According to another participant from group A, the relevance of the MATLAB programming platform to industries shall be communicated more clearly to motivate and engage students. The other suggestion was to allocate more teaching staff as one instructor seems insufficient for handling over 70 students in the computer lab.

3.8 Attendance rate

When asked how often the participants attended the workshops, groups C and D participants indicated that they did not participate in the workshops frequently as they had access to online learning materials: *"I did not attend the workshops after attending the first one, which was basically a revision of the first year, and everything was uploaded onto the CANVAS® [online learning platform of the University of Auckland], so I could just catch up."* On the other hand, group B participants said they attended almost all workshops as: "*[although], the content was on canvas, but the instructor explained the equations in the workshops and how we should use it for the assignment.*" As for participants from group A, the majority missed two or three workshops throughout the semester. A low attendance rate was found amongst the 2020 participants, with the majority attending a few sessions in-person.

3.9 Participant`s expectation on additional learning materials

The students from group A indicated no other materials to be added and instead suggested to put more attention towards core teaching materials on solving governing differential equations numerically. *"…I think we were given a little bit too much stuff; they ended up being only tangential to the assignments and to what I personally see as the central core of this whole programming module, which is solving governing equations. I think, if anything, they should focus on that central core."* Another participant (A) shared their experience on the 1st year programming workshops and suggested teaching materials be shifted in lecture rooms to allocate more time for the students to test their learning by engaging with the non-assessed exercises. In 2020 the instructor added more guidance and structure to the assignment for more effortless flow. The students reflected on their experience and suggested if alternative methods to solving a problem were included, it would make it easier to understand the problems. *"I think if the instructor could show us different ways of doing it, that would be good, instead of only one solution."* In other words, the students wanted flexibility in solving problems using programming.

3.10 The impact of programming workshops on critical thinking

Critical thinking and analytical skills are affected by doing programming and vice versa. While the participants from groups C and D were unsure if their critical thinking affected, the participants from groups B and A were responded positively, saying their critical thinking affected how they write a code and how they understand scripts from other programmers. A participant from group A with an advanced programming skills responded: *"I do think that learning programming is compelling for your analytical or critical thinking skills… if you are an analytical thinker in a certain way, you are going to find coding more natural."* Other participants from group A without

previous coding background implied the workshops exposed them further to programming, which in turn enhanced the way they approached other problems: "*coding has made me realise that no matter how big the problem is, it can be simplified... It is amazing how it works. For my summer internship, I was dealing with Excel. And I have never even thought about coding at all. Then suddenly [I was advised] I can transfer Excel [data] to MATLAB as well... it is so much easier with coding. Yeah, my critical thinking skills are involved way before I write a code."* Another student from group A replied: *"I was trying to find out why I`m struggling with coding. I was able to understand difficult and complex mathematical stuff very easily. I think I don't break the problem down into smaller pieces and try to understand them deeply, which I think I should and is very important in coding."* The responses from the 2020 cohort revealed that pre-existing critical skills affected the individual`s ability to do programming. This will be explored further in Discussion.

3.11 Workshop recordings

Course feedback provided by the 2019 cohort indicated the students wanted to spend more time on hands-on programming. This led to the preparation of pre-recorded lectures which were made available before each workshop in the 2020 iteration. Therefore, the 2020 participants were asked if the videos helped them absorb the knowledge beforehand. All participants agreed that the videos were helpful in their learning: *"…especially as a refresher before the workshop started, it helps advance our knowledge…."* When asked to compare this pedagogy with the conventional method implemented in 2019, the participants replied that the instructor quickly went over the workshop materials at the beginning of each workshop. The pros and cons, according to a participant from group B, are: "*For some people who understood it the first time [through recordings], it wouldn`t have been useful, and they would have felt like hands-on time was wasted. People who hadn't watched the videos beforehand would come to the workshop to ask questions at the same time [as the instructor is giving lecture] to get a more in-depth understanding."* The rest of the participants felt the same way.

3.12 Open-ended feedback

On open-ended feedback, the 2019 participants implied that the anonymous peer-review marking implemented in the heat conduction group assignment did not fairly capture the team member`s effort. In some cases, it was found that people with the little effort received almost a similar mark as those contributing to the project the most. The participants from group B suggested that the group programming assignment be an individual assessment or involve separate tasks that can be assigned more easily to different team members. It turned out the students had issues when writing the code in MATLAB as a team as the platform does not allow collaborative scripting. Another feedback from a participant from group C: *"I feel like the individual assignment was good. You get to work on it yourself and independently, but the group assignment idea really wasn't the best."* Interestingly, all participants from group A in a separate session felt almost the same way, i.e., group programming assignment to be replaced by an individual assignment. A participant from group A who admitted they were struggling with programming in their first year indicated the workshops were exceptionally helpful: *"The instructor helped me a lot when I went to him individually, and he broke everything down into smaller pieces to help me understand, which really helped me. The workshop should be definitely continued."*

According to the 2020 cohort, the choice of MALTAB as the programming platform must be better justified to the students. A participant suggested using a programming language other than MATLAB with broader applications in the industry. They referred to Phyton as a free programming language. Their peers opposed this idea as, in general, students had less exposure to programming other than those they were taught in their first-year courses. *"…I think it would be a lot more pressure on students if they have to learn a whole programming language as such."* A participant from group A suggested having more interactive tasks during the workshops, such as programming exercises, before realising these activities had already been implemented. Final feedback by a participant from group B concluded: *"I think it was good. It could have improved in various ways discussed in these questions, but they should keep it for future classes."*

4.0 DISCUSSION

CHEMMAT203 is a core course in the Chemical and Materials Engineering programme, where the students learn the fundamentals of hydrostatic, fluid flow, and thermal energy transfer in engineering systems. The programming

workshops aimed to shift the student`s understanding of the above concepts towards gaining knowledge through experience. It should be noted that embedding programming into a course demands an extra workload for the teaching team. However, this implementation creates value for enrolled students, and the workload could be managed by allocating more resources, such as additional staff or teaching assistants with relevant background. The findings from the focus group sessions conducted in this research will be discussed with respect to Kolb`s experiential learning. Kolb [17] has developed his "experiential learning" framework to describe how experience is translated into the concept. Kolb proposed critical stages of the learning cycle as experience, reflective observation, conceptualisation, and experimentation. Kolb`s experiential learning model has been extensively implemented as a pedagogical method for enhancing students` learning experience. A study over the period of ten years revealed this implementation has led to deeper learning [18].

The feedback from most of the participants suggested that the significance of learning numerical tools for solving real-world problems was well perceived by the students. The participants with little to no programming background, acknowledged the fact that writing a programme to fulfil the assignment`s tasks is challenging, but a rewarding experience as it helps break a complex problem into smaller pieces and think through it in a structured way. This was evident from workshop manuals and assignment guides provided to help approaching a problem by breaking it into smaller sections. A shorter version of programming guide exemplar is provided as supplementary material. The relevance to chemical engineering applications also appealed to the participants. Knowledge transformation could be further understood in the context of the four dimensions of Kolb`s learning cycle. The students grasp the knowledge through the theories taught in the class, accompanied by numerical methods either covered in the workshops or through recordings provided in advance, which correlate with the Concrete Experience phase. The formative programming examples and exercises provide an opportunity for Active Experimentation during the workshops and afterwards as the students could change input parameters and observe the impact on the output after running the programme. Assignment tasks encourage the students to reflect on learning materials taught in the lectures, and workshops, facilitating the Reflective Observation phase. Furthermore, the discussion part of the assignment where the students must perform parametric studies using their own code and conduct research by accessing additional resources to support discussion contributes to the Abstract conceptualisation and Active Experimentation stages of experiential learning. To this end, the students assess their learning from the concepts taught in the class when challenged by the assignment`s discussion questions, letting the student to utilise their programme to describe trends that they observe from the code`s output and justify them appropriately.

The contribution of programming assignments towards performance in the final exam was reflected by some participants, classified as A. Some participants recalled that the Heat Conduction assignment helped in a deeper understanding of the concepts, giving them more confidence when tackling the exam questions associated with the heat transfer part. This indicates the impact of programming assessments on Abstract Conceptualisation, which is the comprehension dimension of the learning process. However, similar reflections were not expressed by other participants implying more needs to be done for average students and for the Fluid Mechanics module.

The 2019 cohort was assigned a group programming assignment. The objectives involved enhancing group problem-solving skills (teamwork), communication and collaboration to foster inter-personal skills among the students. The student feedback gave significant insight into this implementation, suggesting the group assessment allowed teammates who were not confident to do coding to review the team`s programme and learn from their peers. However, the assignment could have been structured in blocks more clearly so that task allocation to teammates could be done more conveniently. In some cases, scripting was carried out by an individual or two with a higher level of programming skills rather than being a collaborative task. Lack of confidence and uncertainties around result verification prevented in-depth discussion and further exploration for the team members to understand the significance or physical meaning of the theory entirely. Moreover, the feedback revealed that despite the peer-review marking approach, where team members with less contribution must be penalised in marks, there were cases where people with minor contributions received almost similar marks as active team members. The implementation of group assignment and short submission timeline adversely affected the student learning experience. Therefore, some participants suggested the group programming assignment be improved in future iterations. Anand et al. [19] proposed a guided group programming intervention and implemented the strategy in a 4th-year electrical engineering course to enhance students` problem-solving skills.

The approach involved two scaffolds: Introductory problem solving by the instructor and teamwork towards solving a more complex problem by programming. Teamwork consisted of three stages, i.e., conceptual modelling, problem-solving (solution and optimisation), and writing and tuning the code. Therefore, the authors suggest that group programming to be conducted more systematically to achieve the outlined objectives.

Reflective learning is the other aspect of personal skills that needs our attention in future iterations. The input by some participants suggests that learning materials and assignments offered an opportunity for the students to reflect on the problem-solving and programming skills that they have obtained in the first year. Furthermore, participants found feedback on their submissions helpful in reflecting on their learning experience. However, the same aspects were not echoed by other participants, mainly from groups C and D, indicating a lack of self-reflection among some students.

Concerning the ways to encourage student engagement, the responses clearly revealed the effect of students` academic background on engagement. It turned out the students with less programming experience got motivated through increased guidance and worked examples. In contrast, the students with coding background requested less intervention from the instructor and additional hands-on time. This finding suggests programming background of an individual clearly affects the phase of the learning cycle they predominantly experience during the workshops. Students with programming skills tend to transform knowledge by extension (Active Experimentation) or intension (Reflective observation). In contrast, less experienced students try to grasp understanding through apprehension (Concrete Experience) or comprehension (Abstract Conceptualisation). Therefore, the authors suggest the workshops to maintain a balance to engage both groups. In the 2020 workshops, the students were provided with lecture recordings covering the learning materials for each session, allowing them to have more hands-on time and more interaction with the instructor. Despite the transition to remote teaching and learning due to the COVID-19 restrictions and lower attendance rate, the participants indicated the assignment tasks were engaging and allowed them to ask questions during the workshops and do self-learning.

Programming workshops were held in-person in 2019, and the introductory lectures were not recorded, resulting in a higher attendance rate. A correlation was found between the attendance rate and the individual`s performance in the course, with lower attendance resulting in lower performance. In contrast, teaching and learning materials, including pre-recorded lectures, were readily available to the 2020 cohort, which, together with the transition to online workshops, resulted in a lower attendance rate. It is worth noting that the programming workshops were additional to four-hour weekly lectures and were not included in the student's timetable. Average students perceived the workshops mainly as extra-curricular sessions to get support regarding the assignments. Therefore, replacing one timetabled lecture hour with the weekly workshop is suggested to promote a higher engagement.

The critical thinking and analytical skills are the other aspects that correlate with the individuals' background during solving problems by programming. In general, high-performing students without advanced programming knowledge found their pre-existing critical thinking and problem-solving skills positively affected their ability in doing programming. In contrast, the students with advanced programming skills found their knowledge affects the way they approach an engineering problem, and their critical thinking is affected by programming knowledge. Participants who struggled with the course found it difficult to share their experience around this question, although the facilitator elaborated further on the question. This indicates that the critical thinking skills for this cohort might not significantly affect their learning in programming or vice versa. This, in turn, might have led to lower contribution towards group programming and lower overall engagement among this cohort. Therefore, it is suggested to shift the students' focus from extensive programming towards activities which will help the students perceive programming as a tool to analyse the concepts taught in the class. This intervention along with group-based discussion activities will enhance one`s critical thinking skills and provide a higher cohesion between the lectures and workshops.

5.0 CONCLUSIONS

In this study, a programming module was embedded into a Chemical Engineering course, and the learning objectives were assessed through programming assignments relevant to the engineering discipline. The impact of this approach on experiential learning amongst the students was then evaluated through analysing the feedback provided by voluntary participants during focus group sessions. The input from the participants suggests that the

students acknowledged the significance of the numerical tools for solving real-world problems within the context of heat transfer and fluid mechanics. Workshop manuals and guides provided to the students in programming workshops helped the students break a complex problem into pieces and think through it in a structured way. The assessments and activities associated with the programming workshops were also discussed within the context of Kolb`s experiential learning framework. Feedback provided by the 2019 participants allowed identification of points of strengths and areas which needed improvement, which was implemented in the 2020 iteration. The programming background of an individual clearly affected the phase of the learning cycle one predominantly experiences during the workshops. Moreover, programming background was found to affect an individual`s critical thinking while they approach an engineering problem. Therefore, we recommend holding an introductory workshop with a few problems requiring different levels of programming skills and asking students to participate in hands-on programming during the workshop to assess the cohort initially. This assessment would then inform the necessary level of support required from the instructor in the upcoming programming sessions to maximise student engagement. Moreover, self-reflection on learning among the students with lower programming skills should be enhanced through structured guidance and additional reflective activities to be included in workshop manuals. Based on the analysed feedback, the authors suggest such programming workshops to focus more extensively on activities involving more in-depth analysis of the concepts taught in the lectures and group-based discussion, to enhance critical thinking and problem-solving skills for the undergraduate students.

ACKNOWLEDGMENT

The authors wish to thank the Chemical and Materials Engineering students at the University of Auckland (enrolled in the course in semester two 2019, and semester two 2020), who voluntarily participated in the focus group discussions sessions and provided feedback to improve the programming workshops.

DECLARATION OF INTEREST

The authors report there are no competing interests to declare.

REFERENCES

1. Whitfield C.A., et al., *A First-year Design Project Software Tool to Emphasize Problem Solving with Computer Programming in the Design Process*, in *ASEE Annual Conference and Exposition*. 2012. p. 13.
2. Chapra, S.C. and R.P. Canale, *Numerical methods for engineers*. Vol. 2. 1998: Mcgraw-hill New York.
3. Martínez, L.G., et al., *Experiences using PSP and XP to support teaching in undergraduate programming courses.* 2014. **22**(3): p. 563-569.
4. Oreta, A.W.C. and A.P.J.C.A.i.E.E. Balili, *Demonstrating students' skills on integrating knowledge of math and engineering in an applied programming course in civil engineering.* 2015. **23**(4): p. 630-637.
5. Chilana, P.K., et al. *Perceptions of non-CS majors in intro programming: The rise of the conversational programmer*. in *2015 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. 2015. IEEE.
6. Vihavainen, A., J. Airaksinen, and C. Watson. *A systematic review of approaches for teaching introductory programming and their influence on success*. in *Proceedings of the tenth annual conference on International computing education research*. 2014.
7. Coronell, D.G.J.C.E.E., *Computer science or spreadsheet engineering? An Excel/VBA-based programming and problem solving course.* 2005. **39**(2): p. 142-145.
8. Nikolic, S., M. Ros, and D.B. Hastie, *Teaching programming in common first year engineering: discipline insights applying a flipped learning problem-solving approach.* Australasian Journal of Engineering Education, 2018. **23**(1): p. 3-14.
9. Cazzola, W. and D.M.J.I.T.o.E.T.i.C. Olivares, *Gradually learning programming supported by a growable programming language.* 2015. **4**(3): p. 404-415.

10. dos Santos, M.T., A.S. Vianna Jr, and G.A. Le Roux, *Programming skills in the industry 4.0: are chemical engineering students able to face new problems?* Education for Chemical Engineers, 2018. **22**: p. 69-76.
11. Feise, H.J. and E. Schaer, *Mastering digitized chemical engineering.* Education for Chemical Engineers, 2021. **34**: p. 78-86.
12. Kamaruzaman, M., et al., *Comparison of engineering skills with IR 4.0 skills.* 2019. https://doi.org/10.3991/ijoe.v15i10.10879
13. Kassim, H. and R. Cadbury, *The place of the computer in chemical engineering education.* Computers & chemical engineering, 1996. **20**: p. S1341-S1346.
14. Isong, B., *A Methodology for Teaching Computer Programming: first year students' perspective.* International Journal of Modern Education and Computer Science, 2014. **6**(9): p. 15.
15. Jonassen, D., J. Strobel, and C.B.J.J.o.e.e. Lee, *Everyday problem solving in engineering: Lessons for engineering educators.* 2006. **95**(2): p. 139-151.
16. Walker E.A., Pettit J.M., and H. G.A., *Goals of Engineering Education*. 1968, American Society for Engineering Education. p. 373-377.
17. Kolb, D.A.J.U.S.R.P.H., *Experience as the source of learning and development.* 1984.
18. Wyrick, D. and L. Hilsen. *Using Kolb's Cycle to Round out Learning*. in *2002 Annual Conference*. 2002.
19. Anand, A., et al. *Guided Problem Solving and Group Programming: A Technology-Enhanced Teaching-Learning Strategy for Engineering Problem Solving*. in *2014 IEEE Sixth International Conference on Technology for Education*. 2014. IEEE.

TABLES

Table 1 - Course Syllabus and the alignment with the programming workshops

Course Syllabus	Corresponding Workshop Subjects
Fluid statics	Introduction to MATLAB programming: variables, arrays, functions, plotting, logical operators, *for* and *while* loops, debugging,
Hydrostatic forces on a submerged body and a curved surface	Determine hydrostatic forces acting on submerged objects by *numerical integration*
Conservation of mass and energy, Bernoulli's principle, Reynold's number	Solve fluid flow problems using Bernoulli`s equation and the law of conservation of mass, [*Solve ODEs*]
Viscous Flow in Pipes, Major and minor losses, pipe flowrate measurement techniques	Solve non-linear equations, plot the velocity profile inside a pipe and find average velocity, and flowrate of a pipe with major and minor energy losses
Heat conduction, Fourier`s law, steady-state conduction, heat resistance network	One-dimensional steady-state heat conduction: discretisation of slab system, cylinders, matrix decomposition tool in MATLAB
Transient heat conduction, lumped model (integration against time), Biot number, Numerical integration using MS Excel	Numerical solutions to transient heat conduction (ODE), lumped capacitance model, Euler`s method, Huen`s method, ODE solver in MATLAB, explicit solution methods and their stability
Distributed transient heat transfer – infinite slab vs infinite cylinder (Heisler Chart or Gurney Laurie Chart)	Numerical solutions to One-dimensional transient heat conduction (PDE), explicit finite difference method and solution stability, built-in PDE solver in MATLAB

Table 2 - Assessment-based classification of focus group participants

Participant Description (students who performed …)	Classification	Number of Participants
Well in both programming assignments and Exam (>80%)	A	4
Well in programming assignments (>80%), but averagely in Exam (>50%)	B	5
Well in programming assignments (>80%), but poorly in Exam (>30%)	C	2
Averagely in programming assignments (>50%), but poorly in Exam (<30%)	D	1

FIGURES

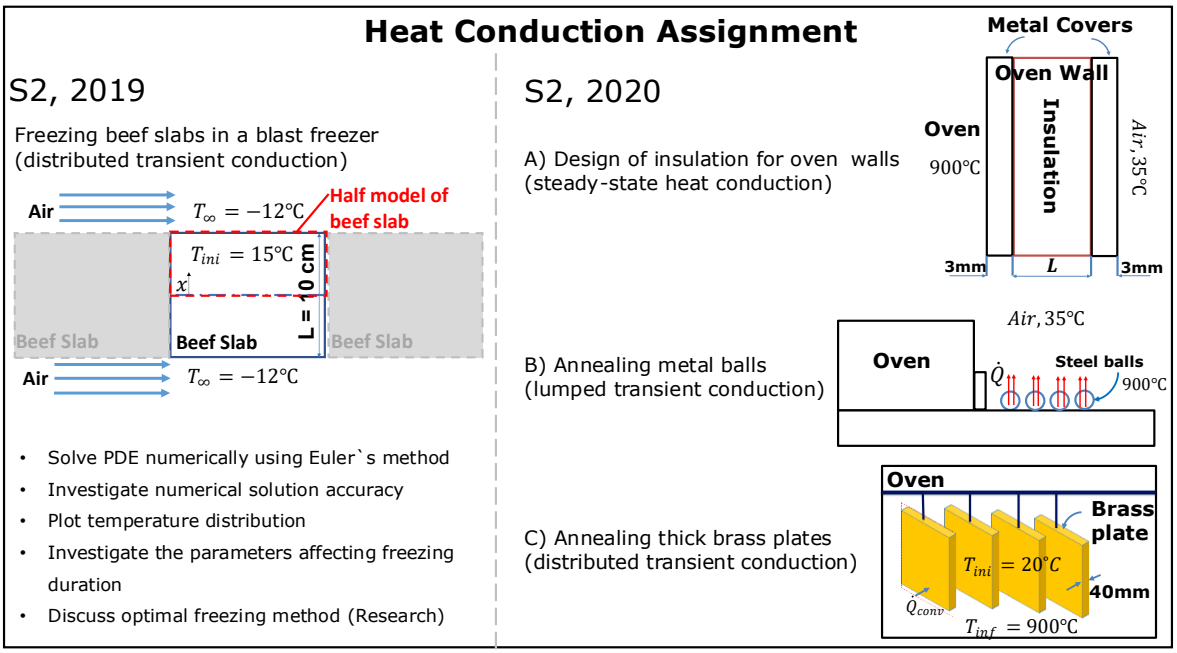


Figure 1- Heat Conduction assignments in 2019 (left), and 2020 (right)

Fluid Dynamics Assignment

S2, 2019

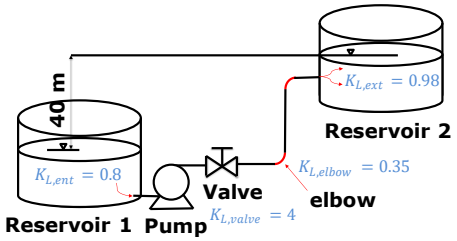
Find Darcy friction factor in an internal fluid flow (Solving the non-linear Colebrook equation)

Programming:

- Find friction factor for both laminar or turbulent flows depending on Reynolds number (Re)
- Solve the non-linear equation by an iterative method
- Programme to give proper warning if:
  - Negative Re (Re<0), or
  - Relative roughness <0.05, or
  - 2000<Re <4000 (transition flow regime)

S2, 2020

Transferring fluid from reservoir 1 to reservoir 2 (Application of extended Bernoulli's equation)



Programming:

- Find volumetric flowrate in the pipe and pump size
  - Guess a value for Darcy friction factor
  - Solve Colebrook equation
  - Compare friction factor with the guessed value and iterate
- Parametric Study
  - If pipe is made of a material with different roughness
  - Plot surface roughness against flowrate and pump

Figure 2- Fluid Mechanics assignments in 2019 (left) and 2020 (right)

SUPPLEMENTARY INFORMATION

1. Focus Group Questions

Assessment of the hypothesis:

1. The workshops focused on MATLAB programming to implement numerical methods for solving fluid mechanics and heat transfer problems. Overall, did you find it useful to enhance your learning experience of the fluid mechanics and heat transfer topics? Why do you feel that way?

2. Overall, did you find it useful to enhance your learning experience of programming which was initially taught in year one? Why do you feel that way?

3. To what extent did you find cohesion between lecture materials, experimental work in the labs, and the programming workshops?

4. How do you think the MATLAB component affected (or perhaps contributed) to your performance in the final exam?

5. Did the active numerical experimentation allowed you to reflect on the significance or physical meaning of the theory taught in class?

6. Self-reflection on learning experience is an important phase in learning by experience. Did the programming workshops give you the opportunity to reflect on your own learning experience?

Assessment on the potential effect of student`s background and implementation approach on the evaluation of the hypothesis:

7. What do you think motivates or encourages the students to be actively involved in learning process during programming workshops? (i.e. real life problem? short quizzes? short practical exercises that students can participate in? …)

8. How often did you attend the MATLAB programming workshops?

9. What learning materials was or could be added to workshops in order to give you more chance to think about the ways you could improve your learning experience on further attempts?

10. To what extent do you feel your analytical skills and critical thinking could affect your learning in programming and vice versa?

11. Some pre-learning videos have been created to help the students absorb the knowledge beforehand, allowing them more hands-on time during each workshop. How do you feel about this implementation compared to conventional running of the workshop by an individual instructor? what do you suggest for enhancing the learning experience?

12. Any general comments or feedbacks on the programming workshops…

2. Programming Workshop Exemplar (Shortened Version)

2.1 Programming Workshop: Steady-state heat transfer

Learning objectives:

- Apply numerical methods through programming in MATLAB to work out steady-state conductive heat transfer in slabs and cylindrical geometries,
- Understand how temperature is distributed as a function of position within multilayer slabs and cylinders by applying heat resistance network analogy

The process to solve heat conduction problems to find parameters such as temperature distribution through the space within the object by numerical methods is briefly outlined below:

- Divide the geometry into a finite number of control volumes
- Perform the energy balance for each control volume and find out its governing differential equation
- Apply the boundary conditions on the boundary control volumes
- Solve the set of differential equation using numerical methods to obtain the temperature distribution

Example[1]: Consider the wall of your house separating the outdoor ambient airflow and the indoor room-temperature airflow consists of a brick layer and a stucco layer, develop a programme in MATLAB to find the heat flux flowing from room to the outdoor atmosphere.

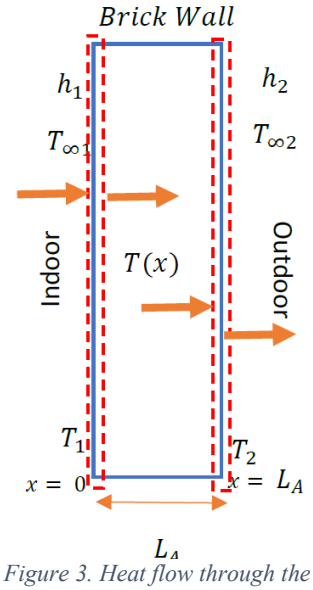


Figure 3. Heat flow through the thickness of the brick wall

Step 1: let us first consider there is no stucco layer. Refer to Fig.3
Step 2: Temperature, T, is a function of space, x : $T(x)= d_0 + d_1x$ (Eq.1) with d_0 and d_1 the constants that are to be determined.
Step 3: Using Fourier`s law, write energy balance at the two control volumes sketched at the boundaries (see Fig.3), i.e., x=0 and $x=L_A$

Solution: @ x=0:

$$h_1(T_{\infty 1} - T(x=0)) = -k\left(\frac{dT}{dx}\right)_{x=0}$$

@ $x = L_A$

$$h_2(T(x=L_A) - T_{\infty 2}) = -k\left(\frac{dT}{dx}\right)_{x=L_A}$$

Step 4: Find $T(x=0)$ and $\left(\frac{dT}{dx}\right)_{x=0}$ from Eq.1. Similarly find $T(x=L_A)$ and $\left(\frac{dT}{dx}\right)_{x=L_A}$ from Eq.1

Solution: @ x=0: $T(x=0) = d_0$,and $\left(\frac{dT}{dx}\right)_{x=0} = d_1$ (by taking the first derivative from Eq.1)

@ $x=L_A$: $T(x=L_A) = d_0 + d_1L_A$,and $\left(\frac{dT}{dx}\right)_{x=L_A} = d_1$

Step 5: These yields:
Solution:

$$h_1d_0 - kd_1 = h_1T_{\infty 1} \quad \text{(I)}$$

$$h_2d_0 - d_1(h_2L_A + k) = h_2T_{\infty 2} \quad \text{(II)}$$

Step 6: Rearrange above equations into a matrix form
Solution:

$$\underbrace{\begin{bmatrix} h_1 & -k \\ h_2 & h_2 L_A + k \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} d_0 \\ d_1 \end{bmatrix}}_{x} = \underbrace{\begin{bmatrix} h_1 T_{\infty 1} \\ h_2 T_{\infty 2} \end{bmatrix}}_{b}$$

Step 7: Write a programme in MATLAB to find the unknown constants: d_0 and d_1 (X = A\b) (You can use the following MATLAB m-file template)

```
Example 1- Steady-state Heat transfer
%% Inputs
h1 = 20;     % Internal heat transfer coefficient [W/m^2.C]
h2 = 40;     % External heat transfer coefficient [W/m^2.C]
k = 1.4;     % Thermal Conductivity of brick [W/m.C]
L = 0.2;     % thickness of brick wall [m]
Tinf1 = 25;  % Indoor ambient temperature  [degree C]
Tinf2 = 2;   % Outdoor ambient temperature  [degree C]
%% defining the matrices A and B
A(1,1) =   ;  A(1,2) =  ;

A(2,1) =  ;  A(2,2) =  ;

b(1) =   ;

b(2) =    ;
%% Solving for the unknown matrix X
X =     ;

T1 =   ;      % surface temperature of brick wall indoor [degree C]
T2 =    ; % surface temperature of brick wall outdoor [degree C]

q_LHS =       ;          % heat flux from the indoor to wall
q_RHS =       ;           % heat flux from the wall to outdoor
q_LHS =round (q_LHS,4);
q_RHS = round (q_RHS,4);

% Sanity Check
tf = isequal(q_LHS,q_RHS);
if tf==1
    disp ("Energy balance holds")
else
    disp ("Eenrgy balance does NOT hold")
end
```

Step 8: Finally:

$T_1 = d_0$

$T_2 = d_0 + d_1 L_A$

Next, we will expand our analysis by incorporating the stucco layer…

2.2 Assignment Brief (Shortened Version)

Problem statement

In a production facility, metallic products undergo an annealing process to improve their ductility, while reducing their surface hardness. This process involves a heating phase where the material is heated up and is maintained above its recrystallisation temperature for a particular amount of time, followed by a cooling phase. The products with the uniform initial temperature of $T_{ini} = 20^{\circ}C$ are placed in an industrial oven which is maintained at $T_{\infty} = 900^{\circ}C$ temperature. Once taken out of the oven, the heat-treated parts are allowed to cool slowly in ambient air at $T_{amb} = 35^{\circ}C$.

A) New insulation for the oven

Insulation plays an important role in terms of reliable functioning of the oven. The old insulation of the oven is to be replaced with a new insulation made of a high temperature insulation material (k =0.03 W/m.℃).

Tasks

A-1: As illustrated in Fig.3, the insulation is sandwiched between two layers of 3mm thick sheet metal (k = 15.1 W/m.℃). Using MATLAB:

- find out the corresponding insulation thickness values for the oven if the plant manager wants to reduce energy loss of the oven between 10%-40% (5% steps) compared to old insulation (thickness= 0.3 m, k = 0.1 W/m.℃). The oven is covered by four metal walls 0.8m (H) x 2m (L). Average heat transfer coefficient values in the inner and outer surface of oven are 140 and 15 W/m^2. ˚C, respectively.

A-2: Using MATLAB and using circuit analogy:

- Determine and plot the temperature distribution across the insulation and across wall thickness based on one of the thickness values you obtained in A-1 for new insulation.
- Justify in your report, why you selected that particular thickness value.
- Next, determine the amount of heat flux from the oven to the ambient.

2.3 The guide provided for Part A of Heat conduction assignment (S2, 2020)

In part A of the assignment, you must develop a programme in MATLAB to perform the following:

- Determine new insulation layer thickness range to meet energy saving targets of 10%-40% , respectively (task A-1) [refer to Fig.4.a]
- Plot the temperature distribution across the insulation and the metal cover [Fig.4.b] and find out heat flow rate (task A-2), and
- Cost analysis and choosing the best option for the insulation layer (task A-3).

A) **B)**

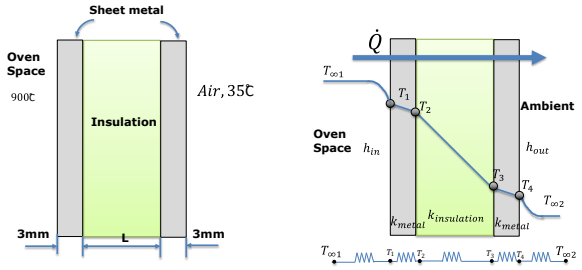


Figure 4. A) Cross-section of oven`s walls , B)Temperature distribution across the wall thickness using resistance network analogy

Here is just one approach that you can adopt to accomplish part A of the assignment (often there are multiple ways of solving problems:

A-1:

- Find out the heat flow rate from oven to ambient with the existing insulation using circuit network:

$$\dot{Q} = \frac{T_{oven} - T_{room}}{R_{total}} \quad \text{Eq (1)}$$

where R_{total} is total heat resistance of the network: convection inside and outside of oven, insulation layer sandwiched between 2 metal walls,

- Then, heat flow rate of new insulation will be a fraction of old flow rate
- Rearrange Eq (1) to get new insulation thickness
- Previous item to be repeated for finding thickness as a function of energy saving (your programme should do the iterations)

A-2:

- We have seen in this workshop that: "Under steady-state heat conduction Temperature distribution is a linear function of x".
- Choose a value for insulation layer from previous task,
- Justify in your report why you chose that one (safety, cost, etc.)
- Write down temperature distribution formulae similar to Examples 1 and 2 of the workshop
- Set up matrix of coefficient (AA) and matrix of answers (BB) in your code and solve it to get matrix of unknown constants
- Plot temperature distribution along the metal and insulation thickness
- Determine the amount of heat flux from the oven to the ambient: $h\Delta T$

References:

[1] Shih, T. M. (2013). Heat Transfer: Lessons with Examples Solved by MATLAB. Cognella, Incorporated.