Flipping a Computational Modeling Class: Strategies to Engage Students and Foster Active Learning

Juan David Ortega-Alvarez
Process Engineering dept.
Universidad EAFIT
Medellín, Colombia
jortega@eafit.edu.co

Camilo Vieira
Education Studies Institute
Universidad del Norte
Barranquilla, Colombia
cvieira@uninorte.edu.co

Nicolás Guarín-Zapata
Civil Engineering dept.
Universidad EAFIT
Medellín, Colombia
nguarinz@eafit.edu.co

Juan Gómez
Civil Engineering dept.
Universidad EAFIT
Medellín, Colombia
jgomezc1@eafit.edu.co

***Abstract*— This Work in Progress, Innovative Practice paper examines the implementation and preliminary results of a flipped classroom strategy in a computational modeling undergraduate course at a [nationality] midsize university. Previous work has discussed the potential of flipped classrooms to leverage active learning through the use of videos and other computer-based tools that encourage students to explore course content autonomously. This study explores how the tools and tactics used in the course, namely out-of-class readings and Jupyter notebooks, can be effective to engage students and foster learning. To that aim, the final study will compare the changes in perceptions and performance of two classes (Fall 2019 and Spring 2020) before and after students take the course. The perceptions part of the instrument focuses on students' self-efficacy and interest in programming. On the other hand, the performance part of the instrument asks students to explain the purpose of simple programs by examining the code. Preliminary results suggest that the flipped format implemented increases students' self-efficacy regarding programming tasks, particularly within students already interested in this computational modeling. These results align well with preliminary performance results, which suggest that the strategy implemented prompts more students to try solving the programming problems and provide explanations, yet not always complete or correct.**

Keywords—flipped classroom, computational modeling, Jupyter notebooks

I. Introduction

The speed and spread of educational reform are seemingly starting to live up to the expectations of the engineering education community [1]. Undergraduate engineering instructors increasingly seek and adopt in their courses evidenced-based, learning-centered approaches to education. Nowadays, academic literature on innovative classroom practices and technological tools to support the implementation of such practices abound. One of such innovative practices, which draws heavily on computer and information technologies (CIT), is the flipped classroom [2]. The fundamental premise underlying flipped classrooms is that students and instructors should use face-to-face interactions to foster learning through discussions, peer interactions, problem-solving, and other forms of active learning. Transmission of content, therefore, is done through other means that do not require students' physical presence in the classroom [3]. There are several CIT tools that allow this transmission of content so that the instructors can select and prioritize what students should peruse on their own, and students can explore this content at their own pace. While autonomous, student exploration is guided by the implicit or explicit prompts that instructors build into the flipped materials. With several technological tools available, often the challenge for the instructors is to find tools that are effective for the characteristics of their courses and their students.

Some educational researchers posit that flipping the classroom is predicated on the use of videos to deliver the content, as opposed to simply assigning out-of-class readings [4]. Educational videos, their argument continues, are effective for out-of-class content delivery because 1) students can access videos from anywhere, provided they have internet access; and 2) students can play, pause, rewind, and explore videos at their own pace, reviewing the difficult parts as many times as necessary. It is worth mentioning that there are several types of educational videos; for instance, the instructors can record themselves giving a short lecture on a specific topic, much in the same way they would do it in a classroom full of students. On the other hand, instructors can also select already-made videos explaining the topic or presenting cases in the context of real-life scenarios. Whether self-made or not, the videos should be accompanied by prompts that guide students in their exploration.

Other authors consider that flipping the classroom is possible using different tools, being both videos and readings among them [5]. Aligned with this view, we maintain that out-of-class readings, often enhanced with interactive content, can promote the benefits of a flipped classroom. Like videos, such interactive content offers the possibility for students to explore it when convenient, at their own pace. Moreover, some of the current educational tools available allow students to practice and get real-time feedback as they peruse the content. In disciplines where mathematical modeling is key to facilitate learning, Jupyter notebooks are one of such tools.

II. Context

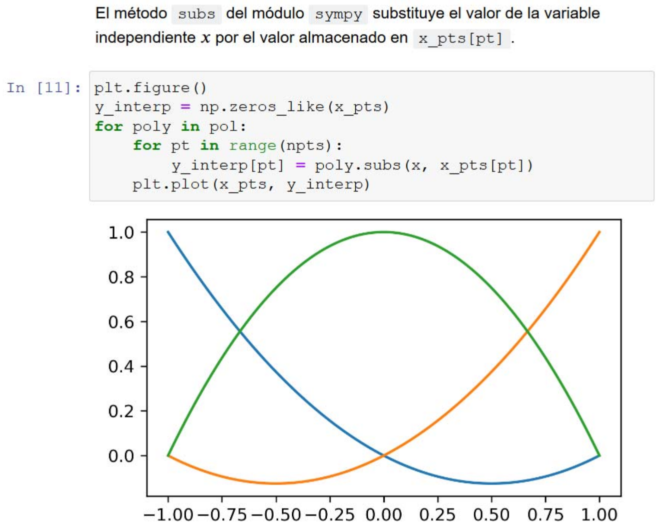
Computational Modelling is a sophomore-level, semesterly course with an average of 70 students per term. There are 16 sessions per semester with a duration of three hours with a 15 to 25 minutes break. The main purpose of the course is to provide civil engineering students with the necessary skills to tackle applied mechanics-based problems through the use of computer simulations. To achieve this goal, the course encompasses contents related to programming, numerical methods, and simulations, mainly through the finite element method. The course is more focused on learning while programming than learning to program [6], [7].

Prior to 2019, the Computational Modelling course was taught in a traditional passive environment covering topics related to algorithms and Python fundamentals; classical numerical methods (interpolation theory, root finding, and quadratures); and computational modeling, focusing mainly on the proper handling of specific computer programs. Given the characteristics of the course, namely class size, problem-solving orientation, and intensive use of computation, the instructors saw it fit to try and flip it. Three different resources were then articulated in the implementation of the flipped classroom approach: First, the different topics of the course were arranged in a set of class notes containing material adapted from classical textbooks together with some independent developments which are not typically available in the published literature. For instance, concepts specific to domain discretization within the context of the different numerical simulation techniques. Secondly, the class notes were connected with a set of resources in the form of Jupyter notebooks, some of which existed prior to the intervention as simple repositories or slides with embedded computation. The instructors also created some videos which focus on software usage rather than explanation of course concepts.

In the most recent version of the course, these notebooks were endowed with explicit pedagogical elements such as worked-examples [8], [9], specific in-class coding problems, computational challenges, peer-to-peer discussion activities [10] and other active-learning techniques [11]. A third, key element in the intervention, is the incorporation into the notebooks of the finite element code SolidsPy [12]. The code is tailored to the design and deployment of learning experiences related to both mathematical and computational aspects of numerical modeling in engineering. Students interact with these notebooks in two different ways. First, they are used as interactive lecture notes where a summary of the theoretical contents is presented interleaved with code snippets, figures, videos, and short questions. This way, the notebooks can be used to study the material before the session. The notebooks are also used as the starting point for the class activities, where some parts of the code are given as scaffolding. Figure 1 presents a screenshot from a notebook example.

III. Methods

To assess the effectiveness of the intervention, we decided to look at students' performance and perceptions as the course progressed. The performance instrument is still under development and meaningful comparisons are not yet available. The perceptions instrument focused on two constructs: self-efficacy and interest. The students completed the survey at the beginning and at the end of the semester. Students' self-efficacy and interest in computation were assessed using a set of five-level Likert-scale items from [13], [14] translated into Spanish and with a preliminary validation process by [15]. The self-efficacy and the interest items are presented in Table I. The possible answers ranged from Strongly Disagree (0) to Strongly Agree (4) on a five-level Likert-scale. The Pearson correlation among these items showed convergent and discriminant validity for the instrument (see Table II).


El método `subs` del módulo `sympy` substituye el valor de la variable independiente *x* por el valor almacenado en `x_pts[pt]`.

```
In [11]: plt.figure()
         y_interp = np.zeros_like(x_pts)
         for poly in pol:
             for pt in range(npts):
                 y_interp[pt] = poly.subs(x, x_pts[pt])
             plt.plot(x_pts, y_interp)
```

Fig. 1. Screenshot of a Jupyter Notebook showing a block of text followed by a code snippet and a graph generated by that snippet.

TABLE I. Survey items assessing participants self-efficacy and interest

Construct	***Code***	***Item with English translation***
Self-efficacy	Q1	Poseo la habilidad para resolver problemas matemáticos a través de programación / *I have the ability to solve math problems using computer programming*
	Q2	Poseo la habilidad de implementar algoritmos / *I have the ability to implement algorithms*
	Q3	Poseo la habilidad de crear un programa de computador / *I have the ability to create a computer program*
	Q4	Poseo la habilidad de crear visualizaciones de datos a través de la programación / *I have the ability to create data visualization using computer programming*
Interest	Q5	Planeo tomar cursos que me permitan incrementar mi conocimiento en programación / *I plan to take courses that allow me to increase my programming knowledge*
	Q6	Siento que la programación será útil en mi carrera como profesional / *I feel that programming will be useful for my career*
	Q7	Siento que la programación es útil en mi programa académico / *I feel that programming will be useful in my studies*
	Q8	Creo que programar es interesante / *I think programming is interesting*
	Q9	Me interesan las cosas que aprendo en los cursos de programación / *I am interested on things I learn in programming courses*

IV. Preliminary Results

The survey instrument was completed by 142 students, while the posttest was completed only by 35 students. Many students in this course tend to skip the last couple of class sessions, which could explain the large difference in sample sizes. However, we argue that the sample of students who completed the posttest is representative of the population because the first quartile, third quartile, and median are almost identical for pretest and posttest grade distribution, as shown in Table III. We also compared the distribution of grades using a violin plot [14], as depicted in Fig. 2. Although the pretest distribution has a longer right tail, they are close to each other and present similar descriptors as shown in Table III.

TABLE II. PEARSON CORRELATIONS AMONG SURVEY ITEMS

	Q1	*Q2*	*Q3*	*Q4*	*Q5*	*Q6*	*Q7*	*Q8*	*Q9*
Q1	1.00								
Q2	**0.75**	1.00							
Q3	**0.58**	**0.52**	1.00						
Q4	**0.63**	**0.60**	**0.56**	1.00					
Q5	0.22	0.23	0.14	0.28	1.00				
Q6	0.18	0.20	0.08	0.20	**0.62**	1.00			
Q7	0.15	0.19	0.03	0.16	**0.63**	**0.84**	1.00		
Q8	0.17	0.21	0.11	0.25	**0.60**	**0.71**	0.77	1.00	
Q9	0.14	0.22	0.13	0.16	**0.60**	**0.64**	**0.67**	**0.69**	1.00

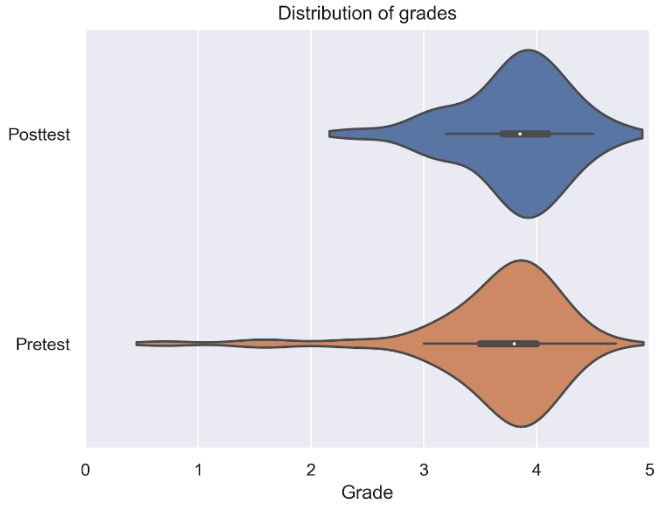


Fig. 2. Comparison of the grade distribution between pretest and posttest on a violin plot. The grades range between 0.0 and 5.0.

TABLE III. COMPARISON OF PRETEST AND POSTTEST GRADE DISTRIBUTIONS

	Pretest	*Posttest*
First quartile[a]	3.50	3.50
Median	3.80	3.85
Third quartile	4.00	4.00

[a] Grades range between 0.0 and 5.0.

Table IV shows the descriptive and inferential statistics for the measures of self-efficacy and interest in computation at the beginning and at the end of the semester. The scores for each student were computed using the factor loadings from an exploratory factor analysis. The resulting scores were normalized from $0 - 100$ for interpretation, as follows: a score between $0\% - 35\%$ corresponds to low self-efficacy/interest, $36 - 70\%$ corresponds to medium self-efficacy/interest, and $71\% - 100\%$ corresponds to high self-efficacy/interest.

TABLE IV. DESCRIPTIVE AND INFERENTIAL STATISTICS FOR THE MEASURES OF SELF-EFFICACY AND INTEREST IN COMPUTATION

	Self-efficacy		***Interest***	
	M	***SD***	***M***	***SD***
Prestest	30.19	22.17	71.48	25.24
Postest	51.81	16.08	73.40	20.74
$t(35)$	−5.98		−0.41	
p-value	< 0.01*		0.68	

On average, students started with a low self-efficacy about computation and ended up with a medium self-efficacy. This change in student self-efficacy was statistically significant ($t(35) = -5.98$, $p < 0.01$). The changes between pretest and posttest for the student interest in computation were not significant. However, the average student interest was already high ($M = 71.48$) and stayed high ($M = 73.4$) after finishing the course.

V. DISCUSSION

At the beginning of this work, we set out to assess whether the implementation of a flipped classroom strategy through the use of digital (Jupyter) notebooks with interactive code snippets would be effective in fostering student engagement and learning. As proxies for engagement, the assessment instrument we used looked at students' self-efficacy and interest regarding computation. Our preliminary results suggest that the flipped format implemented in the Computational Modeling course increased students' self-efficacy regarding programming tasks, particularly within students already interested in computational modeling. These results align well with qualitative preliminary observations of the performance part, intended to assess learning, where more students felt encouraged to tackle programming problems and provide answers to related questions. However, while the number of student responses increased, the quality of such answers is far from satisfactory, with many of them being incomplete or plainly incorrect.

Based on these preliminary results, we maintain that flipping the classroom is possible beyond the use of videos as the paramount tool for content delivery. Moreover the particularities of the course might benefit from−or even require−affordances that videos cannot provide, such as interactive coding snippets in the case of teaching computational modeling. In this vein, we believe that there is no one-size-fits-all tactic to flip a classroom. Instead, a deep understanding of the content and how this content is better learned is necessary to implement a flipped classroom that promotes student learning. Such an understanding, in our case, required that the instructors faced a learning curve regarding both the CIT tools used to develop the digital notebooks, but also the pedagogical knowledge necessary to assess the effectiveness of the implementation. Having both content and education experts proved to be invaluable to speed up this learning curve.

While getting the instructors up to speed with a flipped classroom strategy proved challenging, we believe that it is equally important to prepare students as well. Based on this preliminary experience, we expect to find that flipping the classroom works best to promote learning and positive attitudes towards the content when students are presented with the rationale and aims of the strategy from the onset.

Future work related to this project includes the collection and analysis of data regarding the evolution of student perceptions of the flipped classroom as implemented in the Computational Modeling course. To that aim, we intend to apply a perceptions survey at the first, nineth, and fifteenth week of the class. A perceptions survey designed based on literature was applied at the start and midway of the Spring 2020 class, but the contingency due to the Covid-19 forced us to switch the course to a 100% online, virtual modality on week 9 of 16. The impossibility to hold face-to-face interactions affected the implementation of the flipped strategy we had envisioned, rendering the perceptions survey unsuitable. That said, we believe that a thoroughly planned transition to virtual learning, where synchronous interactions with the students maintain the purpose of face-to-face interactions, can preserve and even heighten the benefits of flipped learning. Further research is required to explore the veracity of this argument.

In the coming semesters, it is also our intention to refine and apply the performance instrument. We recognize that the first version of this instrument, although useful to observe student willingness to engage in solving the computational problems, did not allow us to reliably observe improvement in learning.

References

[1] J. I. Lane, "Understanding the Educational and Career Pathways of Engineers," 2018.

[2] A. Karabulut‐Ilgu, N. J. Cherrez, and C. T. Jahren, "A systematic review of research on the flipped learning method in engineering education," *British Journal of Educational Technology*, vol. 49, no. 3, pp. 398‐411, 2018, doi: 10.1111/bjet.12548.

[3] E. Millard, "5 reasons flipped classrooms work," *University Business*, vol. 15, no. 11, pp. 26–29, 2012.

[4] J. Bishop and M. A. Verleger, "The Flipped Classroom: A Survey of the Research," presented at the 2013 ASEE Annual Conference & Exposition, Atlanta, GA, Jun. 2013, pp. 23.1200.1-23.1200.18, Accessed: Apr. 07, 2020. [Online]. Available: https://peer.asee.org/22585.

[5] L. A. Barba et al., *Teaching and Learning with Jupyter*. NumFOCUS, 2019.

[6] C. Guy-Gaytán, J. S. Gouvea, C. Griesemer, and C. Passmore, "Tensions Between Learning Models and Engaging in Modeling," Sci & Educ, vol. 28, no. 8, pp. 843–864, Oct. 2019, doi: 10.1007/s11191-019-00064-y.

[7] C. Vieira, A. J. Magana, R. E. García, A. Jana, and M. Krafcik, "Integrating Computational Science Tools into a Thermodynamics Course," *J Sci Educ Technol*, vol. 27, no. 4, pp. 322–333, Aug. 2018, doi: 10.1007/s10956-017-9726-9.

[8] J. Sweller, "The worked example effect and human cognition," *Learning and Instruction*, vol. 16, no. 2, pp. 165–169, 2006, doi: 10.1016/j.learninstruc.2006.02.005.

[9] O. Chen, S. Kalyuga, and J. Sweller, "The worked example effect, the generation effect, and element interactivity.," *Journal of Educational Psychology*, vol. 107, no. 3, p. 689, 2015.

[10] A. J. Magana and M. Falk, "Bringing modeling and simulation into my classroom," Faculty professional development workshop at the Materials Research Society (MRS), Boston, Massachusets, Nov. 2017. [Online]. Available: https://www.youtube.com/watch?v=CVpyqBKv9I4.

[11] A. Collins, J. S. Brown, and S. E. Newman, "Cognitive apprenticeship: Teaching the craft of reading, writing and mathematics," *Thinking: The Journal of Philosophy for Children*, vol. 8, no. 1, pp. 2–10, 1988.

[12] J. Gómez and N. Guarín-Zapata, *SolidsPy: 2D-Finite Element Analysis with Python*. 2018.

[13] A. J. Magana, M. L. Falk, C. Vieira, and M. J. Reese, "A case study of undergraduate engineering students' computational literacy and self-beliefs about computing in the context of authentic practices," *Computers in Human Behavior*, vol. 61, pp. 427–442, Aug. 2016, doi: 10.1016/j.chb.2016.03.025.

[14] M. J. Scott and G. Ghinea, "Measuring enrichment: the assembly and validation of an instrument to assess student self-beliefs in CS1," in *Proceedings of the tenth annual conference on International computing education research, Glasgow, Scotland, United Kingdom*, Jul. 2014, pp. 123–130, doi: 10.1145/2632320.2632350.

[15] Camilo Vieira and Juan Gomez, "Integrating Computation into a Civil Engineering Curriculum Instructions at a Colombian Higher Education Institution," in *Proceedings of 2019 Research in Engineering Education Symposium (REES)*, Cape Town, Jul. 2019.

[16] J. L. Hintze and R. D. Nelson, "Violin Plots: A Box Plot-Density Trace Synergism," *The American Statistician*, vol. 52, no. 2, pp. 181–184, May 1998, doi: 10.1080/00031305.1998.10480559.G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.