Increasing student and teachers’ satisfaction in computationally intensive exercise courses

Christa Gall

DARK, Niels Bohr Institute
University of Copenhagen

Abstract

Exercise-dominated courses in the natural sciences can be computationally heavy and require students to be able to understand their computers, deal with computational issues, install programs and to code. However, students often do not have the necessary background. This can lead to frustration for both teachers and students. Therefore, we aimed at increasing teacher and student satisfaction by reducing computer and programming challenges which are not part of the actual Intended Learning Outcomes of the specific course. The goal was to free more time for students to focus on what is essential to the learning and to concentrate better on understanding the subject of the tasks. We hypothesized that students will be less frustrated and more interested in the content of the course as well as stay more engaged in the classroom. To achieve this, we implemented and emphasized different teaching formats so, students could spend more time on understanding their results. Furthermore, we provided codes and programs the students could use to answer questions related to the course subject. We put stronger emphasis on group work (learning and problem solving) during the class than previously and implemented more dedicated and effective class and group discussions. Subjective measures of the success are based on student feedback. This project is the first phase of improving the satisfaction of students and teachers in a classroom and, hence, increase the efficiency of learning. We find that students responded positively to our dedicated efforts in diminishing the efforts spent on programming and computational issues. Indeed, the classroom atmosphere ended up being less stressful and students engaged more in reflections on their results.

Introduction

At Universities, teaching is conducted by researchers that hold a position as lecturers. Per definition, a lecturer is a faculty member at a university or college and is an expert in a specific subject matter who lectures on this subject. Thus, a lecturer will hold a lecture, which is per definition an educational talk or serious speech to an audience. This implies that a lecturer will speak to the audience rather than interact with individuals in the audience. However, such rather passive way of knowledge transfer, after which the students in self-responsibility must acquire further knowledge to deepen their understanding or to complete tasks, seems to be no longer an appealing way of education at universities. Nowadays, lecturers must educate more like teachers, albeit both professions are not interchangeable. The teacher's focus, per definition, is set on ‘helping the students to acquire knowledge, competencies, or virtue via the practice of teaching’. Hence, more responsibility on student learning rests on a teacher. From a societal perspective, these two different roles of ‘educators’ make perfect sense. Teachers hired in schools teach students that are children, who are not expected to take responsibility of their own learning. At Universities, lecturers educate students that formally are adults, and thus, are expected to take responsibility of their own learning. However, research shows that passive knowledge transfer (or passive learning) is not an efficient way of learning (Barry Issenberg et al. 2005). This likely is independent of the seniority of the students, or the subject. Hence, lecturers at universities are expected to teach more and lecture less, prompting them to test new ways to drive engagement, passion, and performance of students in the classroom.

In natural science education such as physics, but also all other science, technology, engineering, and mathematics (STEM) fields, several essential aspects of student training need to be considered. Students need to develop problem- solving skills next to acquiring theoretical and technical knowledge so that they can tackle anything from real-world issues in e.g., industry to abstract problems and fundamental research in academia. However, teaching challenges may arise as lecturers often need to cover large theoretical or fundamental knowledge grounds which can amount to an extensive teaching material presented in class within short amounts of time. On top of that, lecturers need to provide opportunities for students to actively engage with the material and eventually apply the knowledge to different sub-fields. Furthermore, lecturers need to balance the large amount of fundamental knowledge with the fast-paced growing research and technological developments.

Various concepts and theories have been developed that throughout, seem to lead to higher learning successes and outcomes of students at any level. Particularly, interesting are the concepts of flipped classroom learning and the self-determination theory. The former is one that has become attractive since the 1990s, first introduced by Mazur (1996). While at large the concept has been developed for students at ground-school or high-school levels, it works for student learning at any level (Strelan, P. et al. 2020). In such a setup, students are requested to acquire knowledge (that used to be taught by the lecturers in class) by themselves prior to the lesson. The lesson is then used to assimilate this knowledge. This is a task the students otherwise should be doing by themselves outside the lesson.

Indeed, research shows that students are more likely to engage in the active learning process in class (Herreid & Schiller 2013; Velegol et al. 2015) and are more motivated, which positively impacts their learning experience, outcome, and academic performance (Bishop & Verleger 2013; Thai et al. 2017, Lax et al. 2017; Strelan et al. 2020). Students seem to do less well on these aspects in traditional 'one way' lecturing (Albert & Beatty 2014; Roach 2014). However, it remains elusive if indeed this setup is also an effective way of teaching and learning (Kim et al. 2014; Yough et al. 2019). Indeed, better student learning maybe achieved by varying the concepts of learning throughout a course. It has been shown that incorporating four different pedagogical feedback strategies in the course teaching can contribute to enhance student learning and foster a better classroom climate (Fluckiger et al. 2010).

One essential aspect of learning, developing skills and acquiring knowledge in an educational context is feedback. However, for a positive learning experience and outcome, feedback needs to be delivered correctly. Harsh, negative, and non-constructive feedback delivered either intentionally or unintentionally can be detrimental to any human being. One form of feedback, defined as 'information communicated to the learner that is intended to modify their thinking or behavior for the purpose of improving learning', is also termed formative feedback (see for a review Shute 2008).

Formative feedback is a strategy to engage learners to continuously reflect on the learning content and rethink their choices of approach of executing given tasks as well as ways of learning. It helps students to identify what they do not know, ask the right questions, find appropriate resources, and support and evaluate their learning to ensure a successful outcome. Typically, this process is initiated by the teacher, although nothing should prevent students from initiating the process independently or in self-organized study groups. According to Buczynski (2009), frequent dialogue incorporated into learning can be informative, corrective, and motivational. Particularly attractive is the aspect of considering the students as partners in the feedback loop, inducing in students some sense of ownership of their learning.

On the flip-site, too much feedback may lead to students relying on lecturers' or assistants' immediate response (help, feedback) without them making enough of an effort to try to understand the tasks or to find answers to questions on their own. This may can lead to frustration on part of the students if the 'feedback needs' of the students are not met. However, lecturers may also tend to over-explain things in good will and kindness, serving more feedback to the student than needed. This can induce some resentment of the lecturers who may not feel acknowledged for the effort they put in the feedback.

Depending on what the student and lecturers are used to, the perception of what is considered 'good feedback' can vary significantly. This problem has been addressed in the literature, finding that students (learners) need to be actively involved in seeking, generating, and using feedback rather than just being subjected to it (Boud & Molloy 2012; Molloy & Boud 2012; Boud & Molloy 2013). A good pedagogical strategy of feedback that still challenges the students as well as student education in providing constructive feedback may therefore be vital.

Next to the learning strategies and concepts discussed above, to ensure well-being and satisfaction, students need to be able to learn autonomously. This is described in the self-determination theory (SDT) (e.g., Ryan et al. 2002; Deci & Ryan 2002b; Ryan & Deci 2017). It is a theoretical framework that supports a learning environment in which the autonomous motivation of students increases and human basic needs of autonomous decision taking and acting are met. According to research and theory, empowerment of personal development, growth, and well-being as well as intrinsic motivation requires satisfaction of autonomy, competence, and relatedness (Ryan et al. 2002; Deci & Ryan 2002b, a; Ryan & Deci 2017).

In a classroom, lecturers can foster self-determination of students and strengthen their autonomy by giving students optimal challenges and providing greater freedom of choices in the way of learning (Deci & Ryan 2002a; Ryan & Deci 2017). Students that can live up to expectations and believe to be able to deal with challenges can feel competent. A sense of relatedness can be induced by acknowledging feelings, giving a meaningful rationale for requested behavior, and providing task oriented non-personal feedback (Deci & Ryan 2002a). Students, but also lecturers that are embedded in such an environment are more likely to flourish and thrive, are less stressed, likely retain a higher mental health and thus, can achieve higher learning outcomes (Levesque et al. 2004).

In practice, to achieve autonomous motivation in students, lecturers should promote students' intrinsic motivational aspects (Reeve 2006; Reeve & Jang 2006). Indeed, studies show that when an autonomy supportive environment is in place (Haerens et al. 2015) students overall perform better. It has also been shown that autonomy and competence is more impacted by the lecturers, while relatedness is influenced by lecturers and peers (Vasconcellos et al. 2020). However, to create an environment that supports autonomy, teachers need to structure the course such that students understand their tasks and know what is expected of them (Reeve 2006). Students need to be given the opportunity to pace their own learning process and to request help when needed (Reeve 2006; Reeve & Jang 2006).

In this project the hypothesis shall be tested that a well thought strategy balancing different teaching methods (flipped classroom and ordinary static lectures) within the theoretical frameworks of SDT and formative feedback in a computing heavy exercise-oriented astrophysics course can lead to a higher student and lecturers' satisfaction as well as learning achievements.

The Course: Astronomical Data Processing

The course on Astrophysical Data Processing is a master level pass / no pass university course. Students are required to be present in class and hand in a written report about their results of their exercises in groups of typically 3 – 4 students. The course is exercise heavy, involving computer programming, calculating, and applying programs. The course is build up as a step-by-step chain to achieve a certain outcome. This means, students are tasked with describing and discussing the process as well as their assumptions made that impact their results. During exercise, most of the time all lecturers and assistant lecturers are present to give constant feedback and help the students to achieve their tasks, applying a formative feedback strategy.

In the teaching methodology context, the course combines elements of a classical lecture, flipped classroom model, formative feedback and adheres to the philosophy as described in the SDT. Such a combination of different methods is possible as the course is in a block structure. There are two course sessions per week for 8 weeks, each session of either 4 or 8 hours of length. A typical session starts with a short one-hour lecture followed by a half an hour-to-hour long review and discussion of results from previous exercise. Thereafter, a short briefing of the new exercises is given. Typically, there are several short breaks between lectures, review, and briefing sessions. During the remaining class time, the students work independently and in groups through the exercises. This gives students a sense of autonomy, competence, and relatedness. A description of the exercises including background reading material and relevant explanations are always put online, typically about a week prior to the exercise and lecture. Students are expected, and thus repeatedly encouraged, to prepare prior to the next exercise. The minimum expectation is that students have read through the most relevant information of each exercise, so they are able to discuss the tasks in class / in their groups.

The study

The field of astrophysics is constantly evolving. New discoveries, insights on the nature of astrophysical events and the evolutionary history of the universe are made at a fast pace. This is largely made possible thanks to modern observing facilities and astrophysical wide-field surveys, producing large amounts of data every night. To process these data quickly, for each telescope and instrument, dedicated data reduction pipelines are developed that automatically provide 'ready-to-analyze' data products for the end-user astronomer. Only for smaller telescopes, data reduction often remains in a step-by-step manual manner. However, the community is making much of an effort to provide pipelines for data processing for any telescope and instrument, so that only some educated decisions and choices of parameters are required to be made by the astronomer who otherwise can run a 'black box' reduction software on any personal computer.

Naturally, these developments have pros and cons. For the teaching aspect, the pros can be summarized quickly: lecturers have plenty of data available for students. However, the cons pile up to a longer list and are largely related to computer and programming aspects. For lecturers, it is important to teach the fundamentals and core principles of data processing. This is essential knowledge to evaluate the quality of the data one does research with. To do so, students need to go through each of the many steps of data processing and understand why each step is important and how it affects the data. Most importantly, it is important to understand what can go wrong at each step and what are the consequences of choices made for every calculation on the results. The simple reason for careful data processing is that it can decide between a breakthrough discovery, a false claim, or a missed opportunity. However, doing hands-on steps of data reduction can be perceived as either tedious or boring or cumbersome for some students, especially when attractive 'one-button-black-box-pipelines' are on the market.

To process data, computational tools are essential as some calculations are complex. Typically, to ascertain such calculations is not part of the ILOs. Rather, it is important to understand the concepts and outcome as mentioned above. However, here start the challenges. Astronomical software that allows for such a step-by-step processing exists, but some of it has been developed decades ago and does no longer run on modern computers. Newer software is largely `python` programming language based. Custom or homemade pieces (or modules) of software also exists, but those are rather sensitive to the computer setup, rapidly changing versions of `python` and `python` modules. This requires students to be acquainted with their computers, operating systems, and programming in general, which is not often the case.

In recent courses, due to a revision of standard computational tools that were used in previous courses but are outdated, new computational related issues have been encountered. Some of those are related to what has been described above and may be traced back to shortcomings in the students' curricula and preceding education. This has led to frustration on both parts: students and lecturers. Furthermore, this has also shown the limitations, to what degree a course structure, provided learning and exercise material can be kept, while computational work-tools are changed. Moreover, it also shows that more emphasis needs to be put on the computational education of students, particularly in STEM fields. The latter for the simple reason as technological and computational developments in these fields progress fast.

Project goal and challenges

The goal of this project is to reduce computer and programming challenges which are not part of the ILOs and hence, increase student and lecturer satisfaction. The intended outcome is to free more time for the students and lecturers to focus on the core aspects of the course, such as understanding the problems, challenges and tasks related to data processing, rather than solving computer issues. Two problems that have been encountered with the computational work-tool revision in previous courses have specifically been targeted.

The first problem revolves around software installation challenges. As mentioned, astronomical software is not part of the 'from-the-shelf' mass produced products that can easily be downloaded and installed from a major software store. Albeit astronomical software developers increasingly pay attention to user friendliness, the fast-paced developments and community efforts to develop and improve software packages that are commonly used can lead to software clashes, incompatibility, or other issues. Another concern is that 'one-button-click' pipelines take the approach of 'one-size-fits-all'. This often does not necessarily match all research goals.

Such issues are not easily fixed and require a good understanding of one owns computer operating system, installed versions, shell scripting etc. These are competences with which a large fraction of students in a course may not be equipped with prior to the course. Another issue is that for some software, different installation methods and sequences are available. Making the right choice of the installation software and methods as well as of the sequences of installing programs can decide about whether a program will work as expected or not. While one could say that such problems by now are 'standard' and faced by everyone working in STEM fields, for students that have at this point not been in touch with software products not coming from the shelf, such problems can be insurmountable obstacles. Moreover, as observed in previous courses, it can shift the focus of the students to solving software issues rather than solving problems that are part of the ILOS. In worst cases, it can demotivate and add to the stress level of students spending more time on getting the computational tools to work than doing the calculations needed to perform the exercises. On the lecturer's side, such computational issues also shift the focus away from well thought through didactical concepts and teaching methods, allowing less time for discussions and formative feedback on the exercises and course content, as more time is spent on debugging students' computers. While of course the computational tools are well tested, the tests are often limited to certain computer setups that deviate from those of some students. This is simply because there is a too large multitude of computer systems and setups that can be tested.

The second problems also relate to the computational issues, but here more to the use and application of software packages. Assuming that students together with lecturers can install the necessary software packages, students now need to apply the tools to make the calculations. However, as mentioned above, astronomical data processing in a step-by-step manner including fostering an understanding of the data products, sometimes requires doing individual calculations. This means in short, low-level programming in `python` is unavoidable. While major computations can be done with the provided software, students still need to write snippets of code that connect, call, initialize, etc., the major calculation packages. Furthermore, students need to be able to program simple calculations, manipulate arrays or read and write in-and output files. In previous courses, it has been observed that some fractions of students do not have such competences albeit the course is a PhD/master level course. The main problem with this is not only the extra time students need to spend to make the calculations, but also that students cannot make some calculations at all, thus missing to reach some of the ILOs. Other aspects such as loss of motivation, increased stress, a feeling of incompetency and frustration increases. Independent of the origin of the shortcomings, an adequate addressing of this problem needs to be implemented in the course.

Course details

The course that has been subjected to the project had twenty students. Out of those twenty students, six groups formed of which four groups were composed of four students, one group had three students and one student insisted to remain a one-person group. The course was held at two different locations across the campus area, in three-to-four-hour blocks at each location. The course location had to be changed between the morning and afternoon session on the same course day.

Project strategy

To address the two problems and to reach the goal of more student satisfaction, the following improvements of, and changes to the three major teaching methodologies have been implemented and explored. First, a more detailed software installation guide has been provided to the students before course begin. Students have also been encouraged to install required software and most packages prior to the course. Furthermore, the first course week has been dedicated entirely to `getting the computers ready', helping with installations and a brief introduction to basics of `python` programming. Second, a larger set of pre-made codes has been developed and provided to the students prior to the exercises. Furthermore, a set of `python jupyter notebooks` for individual exercises that contain simple calculations and examples on how to connect some programs has been developed and provided during the course. Some such `notebooks` have been made available prior to the exercise, some thereafter. The latter simply because students are encouraged to first try to work out the exercises by themselves before running a 'blackbox'. Third, along with the provided `notebooks` the focus of the exercise has in some cases been slightly shifted. Some exercise material was changed such that it now requires students to reflect more upon the choices of parameter (for some programs and analysis steps) and their impact on the results. Furthermore, the pre-made `notebooks` and programs enable now to set more focus on the analysis of the data products of each data reduction step. Fourth, to ease the work tasks for the students during class, more briefing and debriefing sessions for each major exercise was implemented, discussing not only the expected outcomes but also the software tools and what to pay attention to during the exercise. In previous courses, it was typically expected that students would inform themselves about the exercise prior to class as part of the preparation. However, it turned out that students rarely do that. Fifth, group-work has been more strongly and repeatedly emphasized during class. While this has always been an integral part of the course (students are required to write a report in a group), more effort has been made to more directly teach students how to work in a group. Particularly, discussions with the students about what is expected of a group member (generally speaking) and what are the advantages and disadvantages of group work. Sixth, in accordance with the discussions on group-work, for the first time, discussions about possible feelings involved in being a member of a group in a course with several groups have been made. This has been implemented for one reason, to increase the relatedness amongst the students and between students and lecturers by setting up a more trusted environment for the students.

Outcome: Student evaluation

The project is the first phase of addressing and improving the satisfaction of students and lecturers in a classroom and, hence, increase the efficiency of learning. The project here has been of an experimental nature and thus, mere subjective measures to evaluate the outcome have been used. Such subjective measures are provided through student evaluations and lecturers reflections. For a better comparison with previous course evaluations, no changes to the general content of the requested feedback have been made than otherwise given by the institute.

The students overall responded positively to the dedicated efforts in diminishing the efforts spent on programming and computational issues. Indeed, the classroom atmosphere ended up being less stressful and students engaged more in reflections on their results. Sixteen out of the twenty students have evaluated the course and provided feedback. Eleven students felt that the academic level of the course is suitable. Same number of students also felt that they have acquired the competencies of the course objectives. All students taking part of the evaluation agreed that the teaching material was relevant to the course and that they received relevant feedback on their oral and written work during the course.

To the question *What was good about the course and why?* the students also responded positively to the efforts. In particular, the students liked the course structure of short lectures (of about 1 hour) at the beginning of the class and felt that it helps to have dedicated time for the exercises. The students also appreciated the course material and described it as 'easily accessible, relevant and useful'. It was also pointed out that students really appreciated the group work. The students could work more efficiently as they realized (and we emphasized) that there is no competition amongst them. This really helped with creating an environment where good communication among the students and with the lecturers were possible. The students said that particularly, the constant communication between the members, was the main tool with which any potential problem was faced. Furthermore, they said that they had the chance of sharing their way of thinking and as a result could improve themselves and understand their tasks better.

Outcome: Lecture reflection

From a lecturer's perspective, the students seem to have responded well to the previously described efforts. Particularly, the prepared installation material, and the one-week session on `python` basic programming have indeed had the intended affect that less time on installation issues were spent in the remainder of the course. Some of the pre-prepared programs and `jupyter notebooks` have also largely been used. It has been interesting to observe, that students that were already familiar with `python` programming could easily write their own programs while students unfamiliar with any kind of programming even struggled with what was prepared. On the group work aspect, it appears that students indeed were helping another more within, and sometimes even across the groups. Overall, the conscious efforts made on fostering a respectful and trusted environment has indeed led to more student interaction and engagement as well as student-lecturer discussion. This is noticed as technical assistance from the lecturer related to solving computational issues was less often requested. Rather, students would ask questions more related to exercise results and the overall procedures.

Caveats and improvements

The main goal of this exploration is to increase student and lecturer satisfaction by alleviating issues related to computational problems, and hence, frustration. The expected outcome is that students would learn more about data processing and develop a deeper understanding of it. Furthermore, it is expected that students stay more engaged in learning in the classroom. According to student evaluations, on a subjective base, this has largely been achieved. However, the study is limited to only one course and a small number of students (twenty students) including quantitative evaluations. Thus, future studies are needed. The evaluation scheme may also be more directed towards the specific set of improvements to better isolate which of the new strategies are most relevant. Nevertheless, some elements can be addressed.

One is related to what has been attempted to improve already in this project, and that is the alleviation of computer frustration by providing more detailed material. While 9/16 students strongly agreed and 6/16 students agreed that the teaching material was relevant to the course, there has been comments in the written feedback from the students saying 'it will be better if more instructions and materials about how to use Python can be given to students before the course begins' as well as 'getting a good installation guide written for the needed modules'.

Another aspect is related to the strongly emphasized group work in the course. While overall students appreciated the group work efforts and could see the benefits of it, some groups did not as well. Comments from the students are like ‘maybe give people the option of working alone -- I myself ended up doing the entire project alone while my group gets credit. If I had the option, I would have worked alone’ as well as ‘the exercises should not be split between group members, as it is then difficult to compare results’.

Summary and discussion

This is a short exploratory study to increase student and lecturer satisfaction in a computational heavy exercise driven astronomical course. The main goal of the project is to reduce computer and programming challenges for students to free more time to be spent on the essential course ILOs. It is anticipated that students can learn more about data processing, deepen their understanding of the astrophysical problems and stay more engaged in the classroom. The entire course comprises different didactical strategies, theories, and philosophy, such as flipped classroom model, group work, formative feedback, and self-determination theory. Six different changes and improvements to existing strategies have been explored, all oriented towards reducing computational challenges and fostering better communication and assistance amongst students. The goal has largely been achieved, and students indeed were more satisfied during the class, helped another more within groups and struggled less with computer program installations. This led to a calm work climate. Dedicated efforts to stimulate a trusted environment also came to fruition.

However, further studies are required to provide a more qualitative analysis. This will also allow to evaluate, if the positive student response is due to the set of students that participated in the course or can be generalized. The written student feedback clearly demonstrates that more work on the computational aspect is necessary. However, it remains unclear to what degree these efforts can and should be localized to this specific course. Furthermore, it is also unclear whether elaborate didactical teaching methods can aid with this issue. As discussed in previous section 2, computer developments are fast paced in both, technology, and programming. Training students in using computers as modern replacements of pocket calculators or typewriters may be insufficient. Furthermore, training students on computers and programs that are quickly outdated may not be fruitful either. Instead, students may be required to develop a deep conceptual understanding of computer systems and programs and to learn different kinds of competencies that allow them to quickly orientate and re-adjust themselves in the computationally and technically rapidly evolving realm of all STEM fields. However, such competencies cannot just be acquired as a byproduct of computationally intensive courses with other ILOs. Dedicated student training is necessary, which may need to be implemented in STEM field curricular already in the first year, if not even already at high-school level.

Acknowledgements. We would like to thank Dr. Lise Christensen, Prof. Ian Bearden, Dr. Ricardo Karam, Dr. Frederik Voetmann Christiansen and Dr. Lotte Ebsen Sjøstedt for helpful feedback and suggestions. This work is supported by a VILLUM FONDEN Young Investigator Grant (project number 25501).

References

Barry Issenberg, S., Mcgaghie, W. C., Petrusa, E. R., Lee Gordon, D. & Scalese, R. J. (2005). Features and uses of high-fidelity medical simulations that lead to effective learning: a beme systematic review. Medical teacher 27, 10–28.

Mazur, E. (1996). Science Lectures: A relic of the past?, vol. 9, 13–14.

Strelan, P., Osborn, A. & Palmer, E. (2020). The flipped classroom: A meta-analysis of effects on student performance across disciplines and education levels. Educational Research Review 30, 100314.

Herreid, C. F. & Schiller, N. A. (2013). Case studies and the flipped classroom. Journal of college science teaching 42, 62–66.

Velegol, S., B., Zappe, S. E. & Mahoney, E. (2015). The evolution of a flipped classroom: Evidence based re-commendations. Advances in Engineering Education 4, n3.

Bishop, J. & Verleger, M. (2013). Testing the flipped classroom with model-eliciting activities and video lectures in a mid-level undergraduate engineering course. In 2013 IEEE Frontiers in Education Conference (FIE), 161–163.

Thai, N. T. T., De Wever, B. & Valcke, M. (2017). The impact of a flipped classroom design on learning performance in higher education: Looking for the best "blend" of lectures and guiding questions with feedback. Computers & Education 107, 113–126.

Albert, M. & Beatty, B. J. (2014). Flipping the classroom applications to curriculum redesign for an introduction to management course: Impact on grades. Journal of Education for Business 89, 419–424.

Roach, T. (2014). Student perceptions toward flipped learning: New methods to increase interaction and active learning in economics. International review of economics education 17, 74–84.

Lax, N., Morris, J. & Kolber, B. J. (2017). A partial flip classroom exercise in a large introductory general biology course increases performance at multiple levels. Journal of Biological Education 51, 412–426.

Kim, M. K., Kim, S. M., Khera, O. & Getman, J. (2014). The experience of three flipped classrooms in an urban university: An exploration of design principles. The Internet and Higher Education 22, 37–50.

Yough, M., Merzdorf, H. E., Fedesco, H. N. & Cho, H. J. (2019). Flipping the classroom in teacher education: Implications for motivation and learning. Journal of Teacher Education 70, 410–422.

Shute, V. J. (2008). Focus on formative feedback. Review of educational research 78, 153–189.

Buczynski, S. (2009). Formative feedback stimulates students' thinking and provides teachers with information to guide future instruction. The Professional Association in Education. USA: Phi Delta Kappa International 2.

Fluckiger, J., Vigil, Y. T. Y., Pasco, R. & Danielson, K. (2010). Formative feedback: Involving students as partners in assessment to enhance learning. College teaching 58, 136–140.

Boud, D. & Molloy, E. (2012). What is the problem with feedback? In Feedback in higher and professional education, Routledge, 1–10.

Molloy, E. & Boud, D. (2012). Changing conceptions of feedback. In Feedback in higher and professional education, Routledge,11–33.

Boud, D. & Molloy, E. (2013) Feedback in higher and professional education. Understanding It and Doing It Well 2013.

Ryan, R. M., Deci, E. L. et al. (2002). Overview of self-determination theory: An organismic dialectical perspective. Handbook of self-determination research 2, 3–33

Deci, E. L. & Ryan, R. M. (2002). Self-determination research: Reflections and future directions.

Ryan, R. M. & Deci, E. L. (2017). Self-determination theory: Basic psychological needs in motivation, development, and wellness. Guilford Publications.

Deci, E. L. & Ryan, R. M. (2002). The paradox of achievement: The harder you push, the worse it gets. In Improving academic achievement, 61–87.

Levesque, C., Zuehlke, A. N., Stanek, L. R. & Ryan, R. M. (2004). Autonomy and competence in german and american university students: A comparative study based on self-determination theory. Journal of educational psychology 96, 68

Reeve, J. (2006). Teachers as facilitators: What autonomy-supportive teachers do and why their students benefit. The elementary school journal 106, 225–236.

Reeve, J. & Jang, H. (2006). What teachers say and do to support students' autonomy during a learning activity. Journal of educational psychology 98, 209.

Haerens, L., Aelterman, N., Vansteenkiste, M., Soenens, B. & Van Petegem, S. (2015). Do perceived autonomy-supportive and controlling teaching relate to physical education students' motivational experiences through unique pathways? distinguishing between the bright and dark side of motivation. Psychology of sport and exercise 16, 26–36.

Vasconcellos, D. et al. (2020). Self-determination theory applied to physical education: A systematic review and meta-analysis. Journal of educational psychology 112, 1444.