Motivating Mechatronics and Energetics Students: A Specialized Approach to Applied Mathematics.

Anna Nováková and Viktor Mészáros

Budapest University of Technology and Economics, Institute of Mathematics, Department of Geometry, and University of Debrecen, Faculty of Informatics, Department of Mathematics

ABSTRACT

For STEM students, gaining an intuitive and deep understanding of mathematics is critical, because only then will they be able to successfully integrate it into their own field of study. The main obstacles of this are lack of motivation to learn mathematics, and students not being used to apply the gained knowledge. To discover solutions to overcome the presented difficulties, we developed a new program as a nonmandatory part of the Mechatronics and Energetics BSc courses at TU Budapest. Our aim is to augment the Calculus II. course in a way that accelerates the ability of application and motivates the students in gaining better understanding of mathematics. In the new course, we follow the fields taught in Calculus II., and help the students in understanding the most important concepts by looking at them from new, graphical perspectives. We also encourage them by discussing applications related to their field of interest – for example robot kinematics or neural nets. We test their knowledge by mini-quizzes, three homework assignments and one group project. The innovative course is of a great popularity – while being non-mandatory, 53 students applied, and the attendance rating is exceptional. Students complete assignments on time and are eager to work on their projects. In our concept paper, we will discuss the methodology and outcome of the course. In our experience, applying this concept would be very beneficial for all STEM students, so we are planning to develop the equivalent of the course to augment Calculus III. as well.

1 INTRODUCTION

1.1 Challenges of Mathematics education at TU Budapest

In recent years, the students accepted into our institute tend to be very inhomogeneous in their mathematics skills. This poses a great challenge to us, as there is no opportunity of differential training or talent management in the strict timeframe of the curriculum. Furthermore, while being very varied in terms of talent, undergraduates of every skill level are harder to motivate than before - students often neglect progressing in mathematics, because they are not getting the instant gratification of building useful systems, which engineering undergraduates tend to look for. To counter this effect, our institute launched a new project in the spring of 2019, aimed at motivating the students to dig deeper into mathematics by new means of education. This was started by observing the learning environment and the feedback on our current courses.

1.2 The negative feedback on traditional education

It is a common complaint of our colleagues teaching engineering subjects that both graduates and undergraduates are having hardships applying mathematics in their field of study. The main reason for this is that students have failed to capture key concepts and abstractions of mathematics in their former studies; they only practiced the operative performance of the demonstrated exercises, and this was enough to get them through their exams. Students tend to separate their mathematics subjects into theory and practice - and they often pay little to no attention to the former, because they feel it is not practical, therefore offers no benefits for them.

1.3 Importance of solid fundamentals in mathematics

However, students thinking learning mathematics has nothing to offer them could not be further from the truth. The most important mission of our education system is to help our students gain the knowledge needed to be competent and succeed in their careers. As our environment rapidly changes, we must rethink our approach to education; in today's knowledge-based society, the individual's economic competitiveness is based on how applicable and flexible his/her knowledge is. This means that there is a lesser demand on encyclopaedic knowledge, and more on skills and know-how that can be adopted rapidly to the ever-changing work environment. Coupled with the digital revolution, which means any data can be acquired with incredible ease, it is trivial that our new work environments put such emphasis on life-long learning and self-education that is unmatched in history. However, there are skills that are incredibly hard - for even the most talented students- to learn individually: mathematics is one of them. In addition, understanding mathematics is a gatekeeper of higher engineering concepts, so in order to help our students progress in their engineering careers, we must aid them in acquiring the key abstractions in mathematics and other fundamental subjects - that even the best students have problems learning themselves - but to manage this successfully, we need to overcome their rejective attitude towards theory and mathematics in general.

1.4 Getting students involved - informal learning

Today's students tend to be less resistant to the delay of gratification; this is the main reason of them neglecting calculus and other fundamentals. In these subjects, they are not getting the confidence boost of gaining a directly applicable skill, as these are traditionally based on the concept of formal learning, which, and its informal counterpart, is described below:

- Formal learning is independent from application. It is autonomous, as it serves no purpose other than education. Formal learning was designed in order to be more efficient: which efficiency measured by how well it serves the purposes of work, life and progress [1].
- Informal learning is based on heteronomy: it has other purposes than mere education, unlike formal learning.

The approach above also states that formal and informal learning are in a symbiotic relation, and only through the latter can the former contribute to the gaining of knowledge. The traditional practice was that undergraduates undergo formal education in their first few semesters, and only then can they complete their gained skills with informal learning; but this is getting harder and harder to carry out properly, because of reasons stated before. In order to get students motivated to learn mathematics, especially calculus, formally, we must motivate them with giving them opportunities to apply their knowledge instantly. This motivation boost, paired with the informal learning done while solving practical problems, should guide our students - of all talents - to pursue learning formal mathematics with greater enthusiasm, to their great benefit, as shown above.

2 EXPECTATIONS

The vital challenges and demands of the new learning environment - raised by students, our colleagues and the industry - can be summarized as bellow. A new approach must focus more on students learning the key abstractions and concepts in mathematics, in order to make applying the learnt theories in new fields of practice easier for them.

The new course should motivate the students to dig deeper into mathematics by showing them how crucial mathematics is through practical examples taken from the industry or personal experiences.

To augment their theoretical training, students shall also acquire computing skills and techniques in the course; with which they will be able to solve problems by themselves.

Our new course should apply the concept of informal learning - "learning while doing" - in order to condition the students to apply their knowledge.

3 OUR CONCEPT

The idea of an innovative mathematics course, fulfilling the criteria stated above, was brought to life by us in '18's early autumn. The system of requirements and the vision of the course was raised by Benedek Forrai, the curriculum and methods were worked out together by the writers of the paper.

The lessons of the course shall follow Calculus II., focusing on fields that require more attention, in order to motivate constant learning, and simultaneously making it easy for students to prepare for their main course.

In the new course, the students should be presented new, and if possible, graphical ways of understanding the concepts learned in the main course. A graphical approach makes it easier to visualize problems that students would consider mundane or "not practical" before, and it also accelerates to ability to pair the engineering concepts with mathematical concepts.

The students should be presented the crucial importance of understanding mathematical theory deeply and intuitively in engineering through personal experiences or shared experiences. This would counter the ever-present criticism of our graduated students, who state that they don't understand the need for "plain, boring theory".

In order to prevent students from trying to learn everything just before the exam - as they often do with non-mandatory subjects - the students should only be obliged to do assignments and projects alone and in groups. This way, we also apply the theory of informal learning, while developing essential hard and soft skills - following their mandatory subjects, these students have had no projects in their studies before.

In their assignments, the students shall also be required to complete tasks that require self-education through the internet. This should help them realize that "why learn it, I can google it" is not always a viable answer.

4 EXECUTION

4.1 The framework of execution

Traditionally, the top of mathematics courses in TU Budapest are accredited courses dedicated to talented students, who wish to pursue mathematics further - even compete in mathematics - and take their knowledge to a higher level. These courses, just like the one introduced above, follow and augment the fields of the main, mandatory mathematics courses, and are managed by Brigitta Szilágyi. On the other hand, these courses only attract students who are very confident in their knowledge - around the upper 15-20th percentile of engineering students are interested in this way of learning. This approach leaves many students, who could profit from such a course, unmotivated to expand their mathematics skills, because they - as engineering students often think - are not interested in "pure" mathematics, they only care about building things. In the February of 2019, when one of these "traditional" augmenting courses started, we launched a parallel course with a new, innovative curriculum fitting the criteria above, named "Applications of Multivariable Calculus".

4.2 The structure of our new course

Just like it's talent management counterparts, "Applications of Multivariable Calculus" yields 3 credits, and is taught in one practice per week, for one semester. Expectations from students differ from traditional, however - because we wanted to follow the guidelines stated before - instead of taking one exam, assignments were done by the students. These assignments are listed below:

Individual assignments: students must complete 3 homework assignments, all of which require around 10-15 hours of work to complete, and require both mathematical, engineering and problem-solving skills. Students are also required to write small summaries of the most vital concepts in their own words; this hones their communication skills and also deepens their understanding. The assignments always contain a task that produces something of value, for example analysing a song or recognizing a face with a camera; this motivates students while widening their perspective further in their field of study. Each assignment is rated on a 0-20 scale, and students also receive a feedback summary - which is unprecedented in our institution.

Group assignments: students also complete a group project until the end of the semester, in groups of 3 to 5, which they present orally in the last lecture. The problem they solve in their project is chosen by them, but it needs to be relevant both to the curriculum of the course, and to their field of practice as well; this helps them in recognizing mathematical problems in engineering later.

Attendance: mandatory on the 70% of the lectures and is monitored through a small "Kahoot" quiz in the end of every lesson. The quick tests also help the students in capturing the vital concepts [2], [3], [4]. Attendance was not a question, nearly all of the 53 students who took the course attended to 90% of the lessons.

4.3 The structure of one lesson

We always start our lectures with a quick summary of what we will show the students, and we also motivate them with some applications in the beginning. After that, we follow up with deepening the mathematical foundations and describing the technical background, then we apply these theorems in practice together with the help of computers; we write small programs that solve the discussed problems, which the students follow on their own laptops. The problem solving is coupled by the sharing of our experiences, stories related to the field. This is of a great importance: in their traditional training, due to lack of time, students won't be given such examples, and are often left wondering what they will use the gained skills for. Once or twice every lesson, the guided programming sessions are intersected by "mini-tasks", smaller, easier problems that students solve individually, and the 5 fastest students are rewarded. Lectures are ended with a "Kahoot" quiz, which is a great opportunity to wrap up the session's curriculum, and is always a fun, informal ending note of our lessons - and the students to step on the "podium" are also rewarded.

4.4 Media and software used

4.4.1 Python

We use the Python 3.6 programming language with our students to develop solutions to practical problems - like image processing tasks or processing of signals for example - in order to show the importance of mathematics in application. We chose this language because it is easy to teach and learn, and students are not shown any Python in their mandatory programming classes - despite it being one of the most common languages out there, especially in fast-growing fields like data science and machine learning. Python was developed with code readability and coding speed in focus, and is best in small (<200 line) scripts, this makes it our ideal choice.

4.4.2 NumPy, SciPy

More complex matrix operations, Fourier-transformations and the like are not implemented in basic Python; in order to evade having to program these from scratch, we used the freely downloadable NumPy and SciPy libraries. These provided great "toolboxes" for us to work with and made it easier for our students to experience success. These libraries are also widespread in the industry and knowing them well is a sought-after competence.

4.4.3 Notebook environments

The Python 3.6 programs are written in notebooks, not in scripts. A notebook environment enables us to augment our programs with formatted text, LaTeX functions, links and images, greatly improving understandability and presentability. The notebook method is rapidly expanding in popularity in the engineering and scientific community.

4.4.4 YouTube channels

As stated above, we wanted to show graphical ways of understanding mathematical concepts for our students - this was inspired mostly by the great 3blue1brown channel, which we also promoted in the course. The content on this channel is of such quality and so well thought out that many students began following it themselves, to their great benefit. We were inspired by other channels too.

4.4.5 GitHub

The code written on the lessons, the homework assignments and other pieces of code of importance or interest were uploaded to a GitHub Repository. GitHub is the leading code hosting website having over 26 million public repositories. Most students won't meet this service in their mandatory lessons, but on our course, they are given an opportunity to use and master the website.

4.5 Detailed curriculum

The parallel mandatory mathematics course ("Calculus II.") has three main points in its curriculum: Linear Algebra, Function Series and Multivariable Calculus are taught to first year Energetics, Mechatronics and Industrial Design students in their second semester. We found that linear algebra requires the biggest abstractions, and is more often thought impractical by the students, so we put more emphasis on Linear Algebra and its applications in our new curriculum. In the first few weeks of the course, we paid attention to let the students pick up the pace of the course, and get familiar with Python and the environments used, so we put more emphasis on syntax, basic algorithm design, and describing Jupyter notebooks in detail. After this, we started to follow the program of the main mathematics course. We made a small by-pass too: two lectures were held in the field of Graph Theory, because this important part of mathematics could not fit into the mandatory mathematics courses. The educational program week by week is presented below:

1. Introduction: A quick Python tutorial to take the first steps, and a detailed explanation of the platforms used (Jupyter and GitHub). Mini-tasks: implementation of a signum function in Python, calculating the decimal value of bits representing floating point numbers with Python.
2. The NumPy package, basic matrix operations: a quick intro into the usage of the most commonly used Python library. Matrix and hypermatrix operations. Applications: Numerical integration and noise filtering. Mini-task: calculating the approximation of definite integral of a function with dot products.

Homework assignment I.: the basics of neural networks, calculating the output of a neural net dedicated to recognizing the numbers of the MNIST dataset with matrix operations. Calculating the distance of Bence Sipos's walk by Bence Sipos's phone's accelerometer data, by filtering and numerical integration.

3. Image processing I.: a quick summary of vision technology (from analog to digital). Importance and relevance of computer vision, with industry applications. Introduction to the OpenCV imaging library. Representation of grayscale and colour images in computers: colour spaces RGB, BGR and HSV. Cropping images in NumPy. Thresholds, colour detection. Mini-task: translation from RGB to HSV in OpenCV and Python.

4. Image processing II., Convolution and matrices: Quick revision of colour spaces with a custom GUI program written in class. Colour filtering. Handling videos in OpenCV. Convolution, as a matrix operation, and its relevance in image recognition and signal processing. Edge detection with 2D convolution. Mini-task: RGB to grayscale conversion, finding the matrix to perform edge detection.

   Homework assignment II.: Face-recognition with Haar-Cascades. "Smart TV" application: play/stop a video, whether the webcam of the computer detects a face or not.

5. Matrices - an intuitive approach: Matrices as representations of linear transformations. Matrix multiplication, as composition of linear transformations. Determinant, as the constant with which areas change through a linear transformation. Instead of a "mini-tasks" a "mini-project" is done by students: forming teams of three, they develop a program that displays the famous "Game of Life" in Python, using 2D convolution.

6. Robots and matrices: the basics of industrial robots. Industrial robots and automation in Hungary. Direct and inverse kinematics, and their mathematical representations. Coordinate transformations and matrices. Nonlinear transformations and matrices: performing translation with a matrix: the Denavit-Hartenberg coordinate representation in robotics.

   Projects: team selection and gathering of ideas for the students' projects.

7. Finite element method - and linear equations: presentation of Attila Kossa, the colleague of great authority of Department of Applied Mechanics, about the Finite Element Method's significance in engineering with practical examples, the brief concept of FEM, emphasizing the role linear equations play. After Attila's presentation, we showed the students the graphical intuition behind solving linear equation with Cramer's rule.

8. Graph theory I.: Main concepts and theorems of graph theory. Implementing graphs in Python in an object-oriented way. Mini-task: calculating the sum of degrees in a graph in Python.

   Projects: finalization of problems (chosen by students)

9. Graph theory II.: Dijkstra's algorithm in Python. Revision of graph theory, then a 45 minute "mini-project": visualizing plane routes from online data.

10. Fourier analysis: the significance of frequencies in engineering: music, radio signal demonstration with an RTL-SDR receiver. Control systems and their oscillation. The sound of machining centres, robot vibrations. Difference between Fourier series and the Fourier transform, visualization of a Fourier-series with Python graphics. Fourier series and Fourier transforms from the linear algebra perspective. FFT. Images and

FFT. Demo: online spectrogram. The role of harmonics in the sound of instruments. The working principle of the Shazam music recognizer app. Mini task: recognizing famous pieces music from spectrograms.

Homework assignment III.: Analysis of online news pages with graph theory. Programming a graphical representation of the Fourier-transform in Python, converting downloaded music to a spectrogram, then analysing it.

11. Multivariable Calculus in practice: Deep Learning applications.: presentation of [], PhD student of [institute]. The basics of feed-forward neural nets (revision). Showing the significance of multivariable calculus in backpropagation, the backbone of deep learning applications. Gradient descent methods: stochastic, batch and mini-batch.
    Solving the infamous XOR-problem with a Python script using NumPy.

12. Presentation of projects.: The students present their solutions in a quick pitch. They are rated by us and their classmates.

5 EXPERIENCES, RESULTS, SUMMARY

5.1 Feedback and Survey

Throughout the course, we ran an anonymous feedback system, and conducted a survey on the 12. week of the semester: we asked students studying Calculus II. 26 questions, in which we inquired about their experiences with mathematics and programming in their curriculum. Students were either attending the new course (group “A”) or not (group “B”). 61 students answered our survey, 37 from group “A” and 24 from group “B”.

According to this survey, the attending (“A”) students earned notes in Calculus I. in an even distribution, while the other group (“B”) had slightly more students in the higher percentiles. The other statistics show similarities, so we can say that the two groups had almost similar abilities and knowledge.

We asked the students rate their knowledge of Calculus II. on a scale of 10; the “A” group had a higher average by 5 percent.

We also asked them whether Calculus II. helped them in their engineering subjects, such as mechanics or mechatronics. Group “A” has given noticeably higher score for this question.

According to the collected data, the majority of the students did not have any previous programming knowledge in Python or other programming languages except C (this is mandatory for the faculty), and in group “A”, they felt that their programming skills are improved greatly throughout the course.

We also inquired if students would like to participate in a similar course the next semester: more than 54 percent of the currently non-attending students considered joining, and only 5 percent of the attending group “A” would not continue.

They also rated the "general usefulness" on a scale of 10. The distribution is shown in Fig. 1.

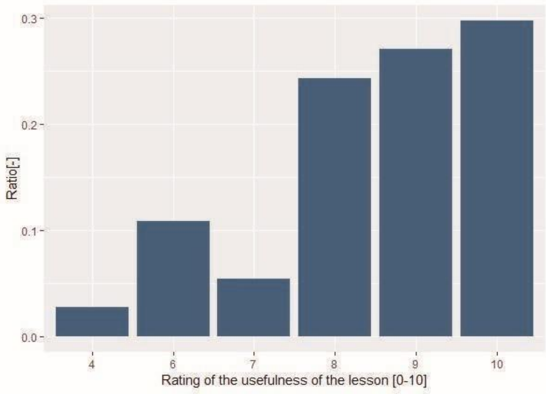
5.2 Summary and acknowledgements

The students' feedback conformed our assumptions that our new course helped them in understand the concepts behind Calculus II. better, and demonstrating real word applications greatly motivated them in their studies.

The extraordinary high participation and attendance also verify the need of continuing this practice the next semester in parallel with Calculus III. and developing a course for Calculus I. as well. As an ending note, let us present some of the students' opinions from the survey:

"...I consider this course very useful. I have been talking about it with my friends, and we would definitely like to continue this kind of learning if possible."

"You guys are great! If not for you, I would have never got the meaning behind linear



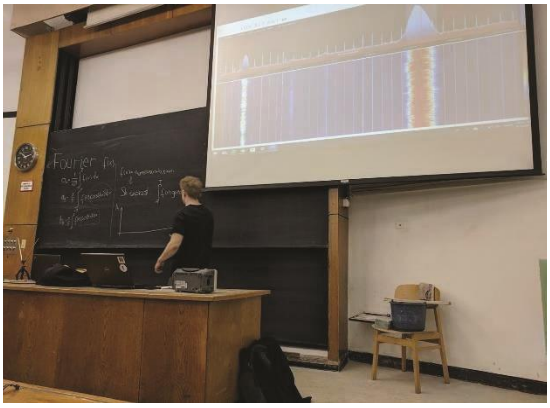


Fig. 1. The rating of the lesson's usefulness *Fig. 2. Software defined radio demo*

algebra!"

REFERENCES

[1] Kálmán, A. (2013), Tanulás és életpálya, Typotex Kiadó, www.tankonyvtar.hu/hu/tartalom/tamop412A/2011-0023_Tanulas_es_Scorm/adatok.html Downloaded (2019).

[2] Roediger, H. L., III, Karpicke, J. D. (2006a). The power of testing memory: Basic research and implications for educational practice. Perspectives on Psychological Science, 1, 181–210.

[3] Roediger, H. L., III, Karpicke, J. D. (2006b), Test-enhanced learning: Taking memory tests improves long-term retention, Psychological Science, 17, 249–255.

[4] Keresztes, A., Kaiser, D., Kovács, Gy., Racsmány, M. (2014), Testing promotes long-term learning via stabilizing activation patterns in a large network of brain areas, Celebral Cortex 24 (11): 3025-3035