International Journal of Pedagogical Innovations

Math Software in the Classroom: Pros, Cons and Tips for Implementation

Steven D. Formaneck

Department of Management, School of Business American University in Cairo, Cairo, Egypt

Received: 19 Oct. 2012; Accepted 25 Nov. 2012

Abstract: Mathematical software in the undergraduate classroom has increased drastically over the last decade. In part, this is due to the use of the same or related software being used in industry, thus preparing students for their future careers. However, for software used in introductory courses, it can be argued that some core fundamentals may fail to be learned due to the computer performing the work for the students before they have properly absorbed the mathematical theory involved. This paper reviews the bulk of the issues and research results that have been generated in the literature. It goes on to derive its own insights into the various benefits and detriments that software can have on a student's learning. The paper then gives distinct recommendations for successful implementation of mathematical software in undergraduate education, with particular emphasis on first-year engineering and mathematics undergraduate students. Finally, further points for discussion are presented.
Keywords: Math Software, Classrooms Tips

Introduction

In today's engineering and math classrooms, the computer is becoming more and more dominant. It first started with the introduction of the calculator. Then came the more advanced graphing calculators. Finally, desktop and laptop computers using mathematical programs such as Mathematica and Maple were able to completely solve calculus and linear algebra problems in mere micro-seconds. This has caused some concern in the academic community as to what the proper use of mathematical software should be in undergraduate education. If it is used too much, students may fail to learn the key concepts of the math that they are expected to know.

Confidence and motivation are two areas in which software programs are theorized to help students learn mathematics. It is hoped that being able to code a math problem into a software program, such as Maple, and solve it using the computer can add confidence to a student in learning the subject. Unfortunately, there has been little evidence to back-up this claim in practice.

In addition to adding confidence, using software programs as part of a math-based curriculum is hoped to add motivation for students to want to learn the material better. This motivation could be from being able to solve larger 'real-world' problems that could not be done by hand. Also, since some students may be using this software in their future careers, this may be thought of as a career-related motivation to learn the software.

Even if confidence and motivation could be enhanced by using software, the question still remains if this translates into better overall performance in the classroom. This may be a question as to how the software is implemented as to how well it enhances or deteriorates a student's learning.

Since software implementation in a math-based class could potentially be a serious factor in the performance of a student, it is important to try to maximize its potential. There are many things to consider when trying to introduce and use software, such as how to ensure everyone has proper access to it and whether there should be separate tutorial sessions dedicated to its use.

Purpose

The purpose of this paper is to examine the major issues in the proper use of mathematical software and come up with its own recommendations on when, where, and how programs such as Mathematica, Maple, Mathcad and Matlab should be used when teaching undergraduates mathematical concepts, especially in the early stages (i.e. first-year) of their education.

We base our work on some of the latest literature that is available on previous theories, empirical studies, and recommendations that have been made in regards to this subject. We apply this information into our own personal recommendations as how to maximize its benefits in an undergraduate course such as calculus or linear algebra.

The overall purpose of this paper is to give a framework of suggested practices for math and engineering teachers should they wish to start, continue, or refine their personal use of mathematical software in their education of undergraduate students. Thus this should be considered a resource from which they can draw from in order to improve their overall teaching effectiveness.

Although programming languages like C and Fortran have long been used in courses such as numerical analysis and has been argued as an effective way of learning algorithmic thinking [2], higher level, math-specific programs are the main focus of this paper since they are more widely used in undergraduate math and engineering courses.

Some of the major findings from the literature are as follows:

- Computer confidence/attitudes does not relate to math confidence/attitudes (Cretchley, 2007) (Fister, 2008) (Frith, Jaftha, & Prince, 2004). Routine practice turns students off (Cretchley & Galbraith, 2002) Having students write their own programs gets them to think logically and reinforces the understanding of methods (Crowe & Zand, 2001);
- Graphical demonstration of mathematical software helped the students to understand better the mathematical problem (Lim, Tso, & Lin, 2008)
- The use of mathematical software even if it would not yield to a real tangible benefit, won't change the student interest towards math. Therefore its usage is encouraged (Lim et al., 2008);
- Software programs helped enhance students' graphical and numerical understanding (Crowe, & Zand, 2001; Guven & Kosa, 2008; Holmes, 2009);
- Teaching demands increase with technology use (Holmes, 2009; Cretchley & Galbraith, 2002);
- Software saves time but can stunt problem-solving (Runge et al., 1999);

A well-constructed workbooks or tutorials can be the key to making efficient use of the software (Tonkes, Loch, & Stace, 2005; Holmes, 2009; Gawlik, 2010).For further discussion of the major finding in the literature review conducted, see the Section 8: Appendix.

Based on the results of the literature review conducted, we have summarized some of the major do's and don't's of using mathematical software in teaching undergraduates.

Table 1. Do and don't

Do	Don't
○ Use good tutorials that any student should be able to understand; ○ Use groups to add additional help to learn the program early on; ○ Put plenty of time into ensuring all technical aspects are taken care of such as availability of the software and proper help material; ○ Take advantage of visualization tools of the software such as 3-D; ○ Have students program algorithms to enhance their understanding of logic behind the methods; Have students attack large 'real-world' projects to help them transition to eventual careers.	○ Use software for solving relatively simple problems (save those problems for the paper-based homework and examinations); ○ Use software more than a tool for supplemental learning of key concepts; ○ Make computer-based tasks so complicated that students spend more time programming than learning the fundamentals.

Conclusions

Based on our review of the literature and our ideas for applications of software in the classroom, we have summarized conclusions of the topic and some general pros and cons found in the use of mathematical software for undergraduate education.

General Conclusions of the Research

○ Research seems to be declining lately. We suspect this is due to two factors:

1. The increasing demand of technology in the workplace and at home compared to the late 1990's & early 2000's where there was still a 'ramp-up' of technology occurring.
2. The use of technology became ubiquitous to the extent that studying its effects on student learning became not of great importance. In other words technology became like the use of paper and pencil while studying

○ Usually undergrad courses in senior and sophomore years use software that will be needed in the workplace, whereas undergrad courses in freshman and junior years tend to focus on 'general studies' where learning the concepts are more the focus, so undergrad courses is where there appears to be more room for argument between how much technology needs to be incorporated in student learning.

- Most of literature comes from Australian institutes which indicates that adjustments needs to be considered during application on an international level,
- The following table summarizes a list of advantages and disadvantages of using mathematical software as an educational aid.

Table 2. Pros and cons

Pros	Cons
○ Adds flavour to a course in regards to accomplishing large	○ Students may use computers as a crutch and never fully

projects; ○ Allows students to spend more time on the core understanding and lets the computer do the simple tasks such as plotting and number crunching; ○ Prepares them for careers where they may need these programs or programming skills in general (e.g. business analyst); ○ Enforces students' learning of the logic behind algorithms as well as the graphical interpretation of the math concepts. ○ Makes learning mathematics a more enjoyable experience ○ It helps students to get over their anxiety when dealing with complex equations in the middle of solving a problem	understand the reason why the software program comes up with the solutions it does; ○ May cause students and instructors to settle for students learning 'just enough' core fundamentals of math to get by and deter students from becoming interested in pure math (i.e. non-application / theoretical mathematics); ○ Can lead to cheating (in the form of homework where they may rely on the computer to do the steps for them that they are supposed to do on their own); ○ Doesn't give students any additional mathematical confidence. ○ Make the student focus more on programming the problem and not the problem itself (Buchanan, 2008). ○ Makes the student uninterested in finding the reason behind the calculations performed (Buchanan, 2008)

Further points for discussion

As a result of our analysis, we have come up a list of questions for the future of software in the classroom:

- ○ How can we harness people's positive attitudes towards computers to help them improve their math attitude?
- ○ Now that computer technology is becoming cheaper and more portable (i.e. handheld devices such as Blackberries), how can this freedom be incorporated into math and engineering courses? Fun engaging projects such as demonstrated in (Henessy, Fung, & Scanlon, 1999) may be further extended to even more integration of classroom knowledge into real-life applications.
- ○ The use of portable devices and its making learning experience a continued process not just in classrooms (Renee, 2008)
- ○ The use of computer technology success has a strong relation with the teacher technological awareness (Newhouse, 2008)
- ○ Crowe & Zand (2001) mention that improvements are always being made in software, so it seems to be easier and easier for students to use it. Also, there are a lot of resources available online in the form of tutorials that students can freely access to help them learn the software. How much of an impact has this had on students' attitudes / performance / learning now compared to say five years ago?
- ○ Many jobs for newly graduated students incorporate 'complementary' computer skills such as database management and visual basic programming. Should these skills be included in undergraduate math courses to some degree? They could possibly be put into a separate course or peppered into other core courses as part of class projects.
- ○ Is math software a learning tool or a power tool in education? (Cretchley & Galbraith, 2002) We think it depends on the situation. If you're using it to crunch big numbers and solve a real-world problem, then it's more of a power tool. If a student's using it for visual representation of math concepts to help see connections, then it would be a learning tool. We think both have merits in the classroom.
- ○ The impact of using mathematical software along with other tools that would enforce the learning experience as a total.
- ○ The use of mathematical software in other courses other than math such as chemistry to allow the student to benefit from using such tools (Goldsmith,1997). This would have three major benefits:

1. The student would have a tangible feeling towards abstract mathematical problems as he will be able now to explore real life examples and how mathematics would be related to it. In other words he know had the chance to explore the interrelationship between various fields
2. The student will have a better hands-on experience with using mathematical tools not only in one discipline (mathematics) which would have a huge impact on his future professional career.
3. This would help to reduce the effect of one of the cons mentioned earlier which is that the student when using mathematical software does not focus on the meaning behind these numbers. Therefore in case of an applied science the numbers entered into the mathematical software would have a meaning (e.g. in Physics, 9.8 is the gravity of earth)

- ○ The growing potential for tablets (e.g. iPads) and handheld devices and their use in education. Also the increasing interest of using handheld PCs for educational purposes and its challenges (Dickerson, 2009).

- The impact of student attitude towards mathematical software and prior knowledge. For example, would the student past experience with a specific mathematical tool encourage him to use the tool more and also would it enhance his learning experience (Ocak, 2006).

References

Buchanan, M. (2008). When it's time to sit back and think again. *New Scientist*, *197*(2637), 22.

Cretchley, P., & Galbraith, P. (2002). Mathematics or computers? Confidence or motivation? How do these relate to achievement? *Proceedings of The 2nd International Conference on the Teaching of Mathematics*, Wiley, Crete, Greece, 1-6 July, Retrieved from www.math.uoc.gr/~ictm2/Proceedings/pap318.pdf, pp. 1-10.

Cretchley, P., Harman, C., Ellerton, N. & Fogarty, G. (1999). Computation, Exploration, Visualisation: Reaction to MATLAB in First-Year Mathematics. *Proceedings - Delta'99 The Second Australian Symposium on Undergraduate Mathematics*, Proserpine, Queensland, Australia, Retrieved from http://www.sci.usq.edu.au/staff/spunde/delta99/Papers/cretchal.pdf.

Cretchley, P. (2007). Does computer confidence relate to levels of achievement in ICT-enriched learning models? *Education and Information Technologies, 12*, 29-39.

Crowe, D. & Zand, H. (2001). Computers and undergraduate mathematics 2: On the desktop. *Computers & Education*, *37*, 317-44.

Dickerson, J., Williams, S., & Browning, J.B. (2009). Scaffolding equals success in teaching tablet PCs. *The Technology Teacher*, *68*(5), 16-20.

Fister, K. Renee (2008). Mathematics instruction and the tablet PC. *International journal of mathematical education in science and technology*, *39*(3), 285-292.

Frith, V., Jaftha, J., & Prince, R. (2004). Evaluating the effectiveness of interactive computer tutorials for an undergraduate mathematical literacy course. *British Journal of Educational Technology 35*(2), 159- 171.

Galbraith, P. (2002). Life wasn't meant to be easy: separating wheat from chaff in technology aided learning? (Keynote Speech) In Proceedings of the Second International Conferamce on the Teaching of Mathematics. (at the undergraduate level): ICTM2. John Wiley and Sons, Inc., USA, (CD). Heronissos: U. of Crete, Greece.

Gawlik, Christina (2010). *Making effective video tutorials: An investigation of online written and video help tutorials in mathematics for preservice elementary school teachers.* VDM Publishing.

Goldsmith, J.G. (1997). Web-Enabled Mathematical Software with Application to the Chemistry Curriculum. *Journal of Chemical Education. American Chemical Society*. Retrieved from http://search.proquest.com.proxygw.wrlc.org/docview/211931056.

Guven, B., & Kosa, T. (2008). The Effect Of Dynamic Geometry Software On Student Mathematics Teachers' Spatial Visualization Skills. *Turkish Online Journal of Educational Technology*, *7*(4), article 11.

Henessy, S., Fung, P., & Scanlon, E. (1999). Supporting Students' Learning Using Portable Graphing Tools. Proceedings of ICTMT4 Plymouth, 9-13 August 1999, Retrieved from http://www.tech.plymouth.ac.uk/maths/CTMHOME/ictmt4/P17_Henn.pdf.

Holmes, K. (2009). Planning to Teach with Digital Tools: Introducing the Interactive Whiteboard to Pre-Service Secondary Mathematics Teachers. *Australasian Journal of Educational Technology*, *25*(3), p. 351.

Lim, L.L., Tso, T.-Y., & Lin, F.L. (2009). Assessing Science Students' Attitudes to Mathematics: A Case Study on a Modelling Project with Mathematical Software. *International Journal of Mathematical Education in Science & Technology*, *40*(4), 441-453.

Newhouse, C.P. (2008). Transforming Schooling with Support from Portable Computing. Australian Educational Computing *23*(2), 19-23.

Ocak, M.A. (2006). The Relationship Between Gender and Students' Attitude and Experience of Using a Mathematical Software Program (MATLAB). *Turkish Online Journal of Distance Education*, *7*(2), article 11.

Runge, A., Spiegel, A., Pytlik Z., L.M., Dunbar, S., Fuller, R., & Sowell, G., & Brooks, D. (1999). Hands-on Computer Use in Science Classrooms: The Skeptics Are Still Waiting. *Journal of Science Education and Technology*, *8*(1), 33-44.

Tonkes, E.J., Loch, B.I., & Stace, A.W. (2005). "An Innovative Learning Model for Computation in First Year Mathematics", International Journal of Mathematical Education in Science and Technology, *36*(7), 751-59.