MICA: A Novel Direction in Undergraduate Mathematics Teaching

by H. Ben-El-Mechaiekh, C. Buteau, W. Ralph
Brock University

Pourquoi les étudiants de mathématiques à l'Université Brock dédient-ils avec entrain autant d'heures supplémentaires à leurs projets? L'honneur en revient au nouveau programme MICA (Mathematics Integrated with Computers and Applications) dont deux des principes fondamentaux sont d'encourager la créativité et l'indépendance intellectuelle et de développer les concepts mathématiques de concert avec l'usage de la technologie et des applications. La réaction des étudiants à MICA va bien au-delà des espérances du département: l'inscription a triplé, les étudiants se dévouent avec enthousiasme à leurs projets, et, selon un sondage interne, ils jugent bénéfique, avec un taux de 91.13%, l'utilisation de la technologie dans leurs cours MICA. Des exemples de projets d'étudiants se trouvent sur notre site (1).

Introduction

Why do mathematics students at Brock University choose to spend hours of extra time on mathematics projects? The reason is the new MICA program at Brock. MICA stands for *Mathematics Integrated with Computers and Applications*: a "hands on" approach to teaching mathematics, making extensive use of technology and remarkably increasing the level of students' engagement. In a recent survey of core MICA courses[i], students overwhelmingly rated the use of technology in these courses as beneficial (91.13% of responses):

On the Use of Technology in MICA Courses
(0 = not beneficial; 5 = very beneficial)
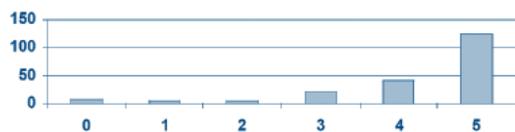
MICA is grounded in years of pioneering the integration of technology into the curriculum spearheaded by Eric Muller (**2**). It was the culmination of a full year of intensive work aimed at modernizing a traditional program sliding irrevocably on a downward enrolment curve. The consensual view of a committee consisting of two-thirds of the department faculty was to embark into a rejuvenation aimed at fostering creativity and mastery of mathematical concepts while making the best possible use of modern technology (**3**). The results are a program that is a pleasure to teach because, in part, of the keen participation of our students as well as a remarkable increase in enrolment[ii]:

Enrolment trend. MICA started in 2001.
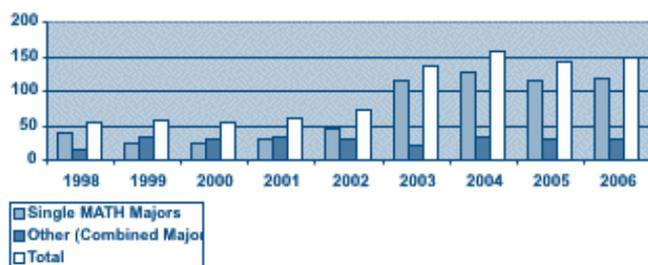
By the mid 1980s, the Department had fully integrated *Maple* (**4**) into its large enrolment service Calculus course (**5**). The following decade saw the extension of the integration of Computer Mathematical Systems and Statistical Analysis Systems to other courses and programs (**5,6**). The MICA reform consisted in introducing three core MICA courses (a total of 2.5 credits) and revising "traditional" courses in light of the MICA principles (**3**) (for example, technologies such as Maple, *Journey Through Calculus* (**7**), *Geometer's SketchPad* (**8**), and *SAS* (**9**) were fully imbedded into Calculus, Linear Algebra, ODE's, Optimization, Cryptography, Statistics, *etc*...). In the core MICA courses, students explore, model and program (in *Visual Basic.NET* (**10**), *Maple*, or C++ (**11**)) mathematical concepts.

MICA offers four streams (Pure Mathematics, Applied and Computational Mathematics, Mathematics Education, Statistics) providing ample opportunity to prepare for pure/applied graduate studies with a choice of more theoretical/applied courses or a teaching career[iii], yet also recognizing the blunt reality that for many, a B.Sc. in Mathematics is a terminal degree. Our graduates have been given the additional computational/modeling/simulation perspective; a clear advantage corroborated by the very positive feedback we regularly receive from supervisors and employers of our former students. Every year, a number of our students are accepted in outstanding graduate programs with funding from NSERC or OGS.

Why Use Computers in Mathematics Teaching?

Exploration in mathematics is severely limited by the use of purely analytical tools. The tight choreography of traditional programs around problems solvable "by hand" inevitably imposes a "canned" and regimented approach.

The importance and difficulty of creating relevant, friendly, yet challenging computer labs (and assignments) cannot be overstated. The development of technology based course materials requires insight and experience. Specific guidelines summarizing some of our experience designing effective labs can be found at (**12**).

Technology offsets the rigidity of a traditional program by providing an endless supply of problems having both analytic and computational aspects. Many of the final projects by first year MICA students are original investigations of problems of their own invention. For example, two first year students developed an unusual novel algorithm for generating Pythagorean triples. They wrote a computer program to test their conjecture and succeeded in verifying the first few cases algebraically. It was particularly gratifying to see their motivation and dedication driven more by the thrill of the discovery rather than the project's grade.

As noted, technology enables students' engagement into a new level of creative discovery; it also plays a role in preparing students for new concepts. In a recent second year MICA lab, students were writing programs aimed at preparing them for the statement of the Central Limit Theorem. When performing sums of a strongly skewed random variable, many asked with some amazement why the curve always drifted into a symmetric shape. These same students would shortly be exposed to the full treatment of the theorem, including proof, in the statistics class. But they are already interested in the result and have experienced this wonderful phenomenon using computer programs they

entirely wrote. In short, *they have experienced a situation where they naturally raised the question before being shown the result.* This principle is fundamental to the MICA philosophy.

The first MICA cohort students were given a half course in Java (**13**) prior to the first MICA course. Disaster was narrowly averted: students found the language's syntax so formidable that it overwhelmed the mathematical content – the *entry fee* was too great. This forced us to simplify the content enormously. The initial programming half course was dropped the following year, in favor of a quick introduction to Visual Basic.Net within the first MICA course. Students essentially taught themselves the language by working, in early labs, through user-friendly step-by-step tutorials. The result was a huge success. Within three weeks, the vast majority of students picked up the language and graphics and proceeded to meet the course's mathematical challenges discussed below.

The Flavor of the First Year MICA Course

MICA is built around a sequence of unique new courses emphasizing creative investigation and presentation of mathematics using computers, and in which the MICA principle of encouraging intellectual independence is fully integrated (14). The first year half-course (Winter term) is compulsory for MICA majors as well as Concurrent Education students; the presence of the latter audience justifying the strong emphasis on communication. The course meets weekly for two hours of lectures and two hours in a computer lab.

In the first part of the course, students are exposed to at least two areas of mathematics suitable for investigations and the raising of sensible conjectures that can be tested by writing programs. Examples of such starting points, accessible to students, are: the Collatz Conjecture and Prime Numbers. The class is divided into small groups and asked to raise any interesting questions about those two topics. For this session to work well, the tone has to be absolutely nonjudgmental and all speculations equally welcome. Conjectures are written on the board and we discuss the feasibility of testing them with our current level of programming skills. Each student is then asked to make his/her own unique conjecture and write a program to test it. The submitted project consists of a functioning program and a complete written report on their conjecture and findings. The program's interface is expected to be self-explanatory, visually attractive and user-friendly.
It is remarkable that for most students, this is the first opportunity in their entire intellectual life when they are asked to raise a mathematical question. Their "stiffness" at the beginning of this process is evident. But after a while, conjectures start popping up quickly and shows that students can raise interesting questions when given the chance.

The second part of the course has a very specific goal: creating a functioning encoding and decoding program based on RSA encryption useable by "spies" in the field. Students are walked through the theory of modular arithmetic, g.c.d. Euclid's algorithm, the group of integers modulo n, and Fermat's little theorem. They write programs that encapsulate each topic, in preparation for the theory and coding of the full RSA algorithm. An aim of this section is to give our students a concrete introduction to abstract algebra and number theory.

The last part of the course deals with dynamical systems. The logistic equation is covered and students write programs

exhibiting the cobweb diagram and out-putting numerical data. Once students' programs are ready, the lectures are given right in the lab so that the students can instantly use their own programs to verify the theory being developed on the board. It is a very exciting way to teach this material.

The centerpiece of this first half course is the student's final project. Students work in pairs or individually to create and test a conjecture of their own devising or, for the Mathematics Education students, to create a Learning Object to teach some aspect of mathematics. The results generally exceed our expectations. Students become very engaged in this work and the final projects are often elaborate, fascinating and a pleasure to mark. See (1) for a sample of online MICA I - III student projects.

MICA in Upper Years

MICA II - III continue building students' experience in facing complex situations not always clearly defined; working with problems that do not have unique solutions; and programming in environments that require precision. MICA courses exemplify how technology can build bridges across mathematics subject areas (3).

Students in the second year course (full year) write programs to investigate a wide variety of mathematical models of diverse types including, for example, discrete and continuous dynamical systems, stochastic models, Markov chains, empirical models, and queuing models. In addition to VisualBasic.Net, they make extensive use of Maple for the necessary analytical calculations. MICA II Students work on two main original projects for which, again, they decide on a topic.

Using the programming environments Maple and also C++ (on UNIX) because of its prevalence in research or industrial environments, the third year course (also full year) emphasizes the investigation of PDE's (such as heat flow and wave propagation) arising in applied settings and reinforces the theoretical courses in that area. Both symbolic and numerical computations of solutions are studied. For example, students work on improving or extending MAPLETs (4) for the animation of solutions to particular PDE's and exhibiting aspects of Fourier series theory. These MAPLETs are also used to explore issues such as approximation and convergence, including a detailed statistical analysis of error behavior of Fourier series. A number of such projects were successfully presented at the international Maple Conferences (4) (summers of 2005 and 2006), rubbing shoulders with presentations by researchers and expert developers of Maple applications.

Students' Response to the MICA Program

In March 2006 we surveyed [iv] students in MICA core courses on different aspects of the program, such as their appreciation of technology in mathematics courses, their learning experience, as well as their background, career goals, learning style, etc...

Students overall rated the use of technology in their mathematics courses as positively beneficial (77.74% of responses; 79.36% when restricted to mathematics majors).

More striking is the opinion of students considering future mathematics graduate studies: a whopping 88.18% of positive responses with a significant 51.82% top rating as "very beneficial".

On the Use of Technology in all Courses
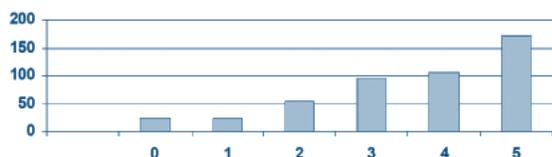
On the Use of Technology for Students Considering Graduate Studies
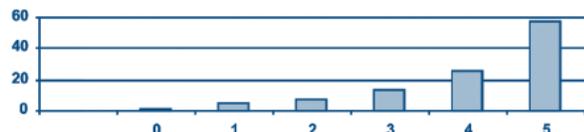
A significant 84.71% of responders describe their students' experience in the Department of Mathematics as positively satisfactory; a figure of comfort and pride for our faculty[V]. Finally it is interesting to note that 74.4% of responses from future mathematics teachers rate their learning experience as useful for their future teaching career - a very strong figure for a pedagogically savvy audience known for its critical approach to technology.

Concluding Remarks

The MICA program uses technology not only to verify and reinforce the learning of theoretical ideas but also to allow students to explore applications that are well outside the usual boundaries of a traditional mathematics program. The act of writing a computer program promotes learning because it puts the student in a feedback loop where they are constantly checking to see if the output from their program agrees with the theoretically predicted values. Students quickly discover that it is virtually impossible to write such a program without first understanding its mathematical content. In this way, students are pushed past a cursory understanding of the material to a new level of mastery. Finally, it should be noted that there is something very personal about writing programs. Our students take considerable pride in writing robust programs with user-friendly interfaces. Their level of engagement and enthusiasm for this new program has made all of the effort spent in its creation entirely worthwhile.

(1)   URL: www.brocku.ca/mathematics/studentprojects
(2)   Muller, E. (2001): *Reflections on the sustained use of technology in undergraduate mathematics education, in The Teaching and Learning of Mathematics at University Level*, D. Holton et al (Ed.), Kluwer Academic Publishers.
(3)   Brock Teaching (2001): URL: www.brocku.ca/ctl/pdf/Brock_Teaching_1.pdf
(4)   Maple, Maplesoft, URL: www.maplesoft.com
(5)   Auer, J., W., Jenkyns, T., A., Laywine, C., F., Mayberry, J., P., and Muller, E., R., (1982): *Motivating non-mathematics majors through discipline-oriented problems and individualized data for each student.* Int. J. Math. Educ.Sci. Technol., 13, 221.
(6)   Muller, E., R., (1991): *Symbolic mathematics and statistics software use in calculus and statistics education* ZDM 91/5 (1991), 192.
(7)   Ralph, B. (1999): *Journey Through Calculus*, Brookes/Cole
(8)   Geometer's SketchPad, URL: www.keypress.com/sketchpad/
(9)   SAS, URL: www.sas.com/
(10)   VisualBasic.NET, Visual Basic, URL: http://msdn.microsoft.com/vbasic/
(11)   C++, URL: www.cplusplus.com/
(12)   Pead, D., Ralph, B., Muller, E., (2007) *Uses of technology in mathematical modelling in* "Modelling and Applications in Mathematics Education", Blum W. et al (Ed.), Kluwer Academic Publishers.
(13)   JAVA, URL: http://java.net/
(14)   Buteau, C. and E. Muller (2006): *Evolving Technologies Integrated Into Undergraduate Mathematics Education*, Proceedings for the International Commission on Mathematics Instruction Study 17: *Digital Technologies and Mathematics Teaching and Learning.*

NEWS FROM DEPARTMENTS
NOUVELLES DU DÉPARTEMENT

Simon Fraser University, Burnaby, BC

Promotions: Veselin Jungic (Senior Lecturer, June 2007); Petr Lisonek (Associate Professor with tenure, June 2007); Ladislav Stacho (Associate Professor, June 2007); Jonathan Jebwab (Tenured, June 2007); John Stockie (Tenured, June 2007).

Death: Steve Thomason (Professor Emeritus April 4, 2007).

Awards/Distinctions: (nine-month-von Humboldt Foundation Fellowship at the Fraunhofer Institut für Techno- und Wirtschaftsmathematik in Kaiserslautern); Michael Monagan (2006-2007 Simon Fraser University Faculty of Science Excellence in Teaching Award).

Visitors: Keith Geddes (Canada, Computer Algebra, Sept. 2006 - June 2007); Keehong Song (South Korea, Computer Algebra, May 2007 - Feb. 2008)