Examining the Role of Mathematics in the IT Curriculum

Joy Gasson
Otago Polytechnic
joy.gasson@op.ac.nz

ABSTRACT

What is the role of mathematics within the Information Technology curriculum? Do students need calculus? Should mathematics be taught at all? The rapidly changing nature of IT in recent decades calls for a serious reconsideration of curricula, with a particular focus on those traditional elements that may have become outdated. In this paper, we argue that the role and requirements of mathematics for IT has changed and that a greater emphasis is now needed on IT-relevant and contextualised pedagogy. Our curriculum is outlined and examples of content and approach are discussed in terms of their alignment with current ACM recommendations and the needs of our students. This review and student feedback has validated our provision of a broad, rich mathematical experience that is relevant and applied to careers in the IT industry.

Keywords: ACM/IEEE Computer Science Curricula 2013, mathematics, discrete mathematics, calculus, CS education, computing, information technology, curriculum, computational thinking, community of practice, work-ready, industry, narrative

1. INTRODUCTION

Mathematics is a fundamental component of all Computer Science and Information Technology degree programmes. As a knowledge domain, mathematics is both broad and deep; covering such diverse areas as number and measurement, geometry and topology, logic and proofs, algebra, probability and statistics, calculus and analysis. Each institution must decide the coverage, depth and place of mathematics within their course to meet the needs of both the institution and their students. The requirement of the educator within each institution is to understand the diverse needs of their student cohort, and to scaffold the learning to contribute to the requirements of their graduate profile.

The ACM/IEEE Computer Science Curricula 2013 (CS2013) provides extensive curricular guidelines and a timely opportunity to examine our local curriculum in terms of currency, relevance and legitimacy.

2. MATHEMATICS, COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

Mathematics is the theoretical foundation of computer science. The connections between mathematics and computer science are "deep and beautiful" (ACM Computer Science Curricula 2013, p. 49). Mathematics provides the framework for reasoning and solving of computing problems (Wing, 2006; Baldwin, Walker and Henderson, 2013). However, there is a gap between the mathematics that students are taught in CS/IT courses and the mathematics they will use as IT professionals (Ralston, 2005). The majority of CS/IT students do not need a traditional theoretical approach to learning mathematics. The majority of CS/IT students are preparing to work in the IT industry developing software and hardware solutions and maintaining computer systems. They need practical mathematical skills that are directly applicable to their work. It is not a question that IT professionals do not use mathematics, but rather recognising the areas of mathematics that they do use. It is perhaps a problem of perception: in engineering, mathematics is used explicitly and is easily recognisable. In computing, mathematics is used more implicitly in critical analysis, logical reasoning and problem solving. To show students the power and relevance of mathematics, educators need to connect theory to practice, to provide content that is relevant to the field, and to provide rich satisfying experiences of mathematics.

Computing is a young and developing field, and the role of mathematics continues to be pivotal to computing. The role of mathematics education in CS/IT education also continues to evolve. While the place of calculus in the curriculum has changed over time (Ralston, 2005), the role of mathematics has remained prominent. Baldwin, Walker and Henderson (2013) warn that mathematics will assume greater importance and that graduates will be expected to apply mathematical tools in the course of their careers.

To identify the most valuable elements of mathematics for CS study, it may be useful to consider the cognitive principles that underpin the learning of mathematics. Wing (2006) proposed a framework in which she identified six core properties of computational thinking:

A conceptualisation of the problem that involves thinking at multiple levels of abstraction,

- A fundamental skill that is performed and utilised, rather than learnt by rote,
- A human problem solving approach that is not replicated by machines,
- A way of thinking that inherently draws on the foundations of both mathematics and engineering, yet complements and extends these disciplines,
- Is about ideas and concepts, not about producing hardware or software products,
- Is fundamental to 21st century living, so that everybody everywhere needs to have access to and appreciate its power.

Computational thinking, as proposed by Wing (2006), encompasses the reasoning and problem solving that are inherent in the study of mathematics. Using a computational thinking approach could provide an interesting new avenue that could empower, broaden, deepen and extend learning in mathematics in CS and IT programmes. It could provide a

This quality assured paper appeared at ITX 2014, incorporating the 5th annual conference of Computing and Information Technology Research and Education New Zealand (CITRENZ2014) and the 27th Annual Conference of the National Advisory Committee on Computing Qualifications, Auckland, New Zealand, October 8-10, 2014. Mike Lopez and Michael Verhaart, (Eds).

direction and impetus to the design of mathematics curricula in CS and IT programmes.

3. THE ACM/IEEE COMPUTER SCIENCE CURRICULA

The 2001 ACM/IEEE Computing Curricula Computer Science Guidelines recommended that discrete mathematics should be taught early in a degree programme so that students could understand how mathematical tools can be applied in practical contexts. It was suggested that Calculus should be taught if and when it was needed. This distinction continues to be made in CS2013.

The CS2013 was published 20 December 2013 and builds on previous ACM/IEEE curricular statements. It is moving away from theoretical mathematics and statistics towards achieving a more applied focus. This reflects an increasing awareness of preparing students for a future beyond our tertiary institutions. It emphasises that curriculum design needs not to be highly theoretical but instead to be responsive to what is of use to students and what the majority will use in their careers. For educational institutions, whose primary aim is training graduates for the workforce, CS2013 provides a useful guidelines and curricula direction.

The CS2013 highlights an inclusive curriculum and the mathematical requirements that all CS/IT students will need during their studies. The CS2013 suggests that the only mathematical requirements should be discrete mathematics, mathematics that is directly relevant for all students. The statement emphasises a need for "a general facility with mathematics" (p.49), "mathematical maturity and clarity of mathematical thinking" (p. 50) which suggests a rich experience of mathematics. It points to a mathematics course that is practical and applied in an authentic IT context. The CS2013 offers not only a guide to curriculum but also pedagogy for teaching that curriculum. There is recognition that a wide range of institutions will use the guidelines and that implementation will generate local institutional curricula.

The CS2013 statement warns that requiring calculus for entry to CS or IT courses is an unnecessary barrier. The use of mathematical examples in first programming courses could also be construed as alienating for many students. It could account for the historically low pass rates in first year CS courses (Robins, 2010). In a recent survey of New Zealand secondary teachers, and their readiness to implement new computing assessment standards, teachers were asked a series of mathematical problems. (For example, what is $\log_2 1024$? If $n = 10$, what is $n(n - 1) /2$? What is 2^{10}? Convert the hexadecimal number 1A to decimal?) Their ability to answer these mathematical questions was used to gauge their confidence to teach the new computing standards (Thompson, Bell, Andreae and Robins, 2013). The assumption that mathematics is an indicator of ability to teach computing at Year 11 (aged 15 years) level shows how pervasive the attitude is. Perhaps the use of mathematics in this way provides a filter so that only those with the greatest mathematical facility are accepted into the computer science classes. Such an attitude does not serve the interests of our students or the IT industry. It excludes students who might make real contributions to the field of computing. Mathematics continues to be important for theoretical computer science, but is no longer central to research and development or many applied fields. This focus may also account for the gender bias in the CS tertiary population which is just as damaging to the industry as a little weakness in calculus.

If tertiary institutions cannot provide inclusive, broad and appropriate education that meets the needs of the CS/IT industry, tertiary institutions will become marginalised and alternative work ready training will be engineered.

4. THE EFFECTIVE LEARNING AND TEACHING OF MATHEMATICS

Learning, cognition and context cannot be separated into parts but are interrelated and interdependent. Effective teaching of mathematics should be considered as a coherent system that encompasses the classroom atmosphere, the learning tasks, the mathematical tools and the classroom discussions. The teacher's role is to select and adapt learning tasks and resources that focus on the key mathematical ideas and concepts and will enhance students' proficiency. Learning is occasioned by the teacher, emphasising the teacher's role as proactively supporting the development of each student's understandings and reasoning (Anthony and Walshaw, 2007).

Educational literature stresses learning as a process of acquiring knowledge within a meaningful context as opposed to the rote learning of isolated facts (Mayer, 2002). In the context of CS education, this means that if the mathematics "is not taught in conjunction with a computer science experience, or it is taught long before the concepts and theory are applied… then the effectiveness of the course is lost" (Wolz and Conjura, 1994, p. 223). By teaching mathematics within a computing context, student will not only gain the mathematical tools but also learn to think mathematically (Henderson et al, 2001).

Lave and Wenger (1991) explain that learning should not only be a shared, social process that contributes to the gaining of knowledge, but that through this process the learners form a "community of practice" that allows "the connectedness of knowing". This means that within the classroom environment, by sharing understandings and practices an IT context, we can help our students begin to gain the knowledge, language and mores of the IT industry. "Learning how to use a tool involves far more than can be accounted for in any set of explicit rules. The community and its viewpoint, quite as much as the tool itself, determine how a tool is used" (Brown, Collins and Duguid, 1989, p.33). It is important for the credibility of qualification and the employability of our graduates that we provide our students with the relevant skills and knowledge to be effective practitioners, and to acculturate them in the standards, ethics and practice of the IT industry.

5. MATHEMATICS AT OTHER INSTITUTIONS

Mathematics is a foundational course of computer science and information technology programmes. Many institutions require a first year paper taught from within their Mathematics department. For example, at Otago University, Computer Science majors are required to take a first year Mathematics, Statistics or Computational Mathematics (Otago University, 2014). At Auckland University of Technology, students must take Applied Statistics, Differential and Integral Calculus, Algebra and Discrete Mathematics, Mathematical Concepts, a course that includes sets, functions, series, limits, calculus, probability and statistics (Auckland University of Technology, 2014). These traditional approaches to mathematical requirements contrast with the "Beauty and Joy of Computing" course at University of California, Berkeley, which offers non-CS majors a contextualised introduction to computing and includes mathematical topics such as recursion and algorithmic complexity. This course has become one of their most popular, has now been opened to CS majors and is credited with "bringing computing to traditionally under-represented groups in computing, i.e. women and ethnic minorities" (University of California, Berkeley, 2014). At Carnegie Mellon University there is a course titled

“Mathematical Foundations for Computer Science”. This course is targeted at first year students and uses logic, sets, induction, functions, and combinatorics as a context to teach the methods of mathematical proof. It is their approach to teaching this content that sets them apart. “Unlike more traditional courses, we use experimentation and collaboration as ways to gain better understanding of the material” (Carnegie Mellon University, 2014). This range of interpretations of the mathematical requirements in the curriculum provides a framework through which to examine our own curriculum.

6. EXAMINING OUR CURRICULUM

Otago Polytechnic provides a three year Bachelor of Information Technology degree where students receive focussed practical instruction. The degree culminates in an industry based project, where students work in small groups for a real client. We attract students from diverse work experiences and educational backgrounds. Many of the older students are retraining to start new careers often bringing with them extensive and relevant work experiences. On graduation they are employed as system administrators, programmers, database administrators, and various IT support roles. Feedback from employers indicates that our graduates are valued for their work-ready industry skills. Cheryl Adams, from Intergen says "We have a great relationship with the Polytechnic and know the graduates we hire will have skills that are current and relevant for this industry" (Otago Polytechnic, 2014).

Since its inception in 1993, there has been a mathematics course in the first semester of the Bachelor of Information Technology programme. In 2004, we identified that the calculus portion of the mathematics course did not meet the needs of the students while studying or in their future employment. We created a Numerical Methods course in discrete mathematics, the knowledge and tools that underpin and are applicable to the field of information technology (Gasson, 2004; Gasson, 2005). In 2009, the three separate mathematics modules were combined into a single 20 credit paper.

6.1 Our Curriculum

The current implementation is titled “Maths for IT” and taught in two two-hour classes over 16 weeks in the first semester of the degree. The current curriculum covers binary numbers, digital logic, number theory, encryption, algorithms and network topology, as detailed in the schedule below.

Table 1: Class Schedule

Class	Topic
1	Numerical Notation
2	Indices
3	Binary Number System
4	Binary Arithmetic
5	Set Theory
6	Digital Logic Gates
7	Digital Logic Gates
8	Coding Systems
9	Counting Techniques
10	Coordinate Geometry
11	Trigonometry
12	Number Theory
13	Modular Arithmetic
14	Check Digits and Random Number Generators
15	Caesar and Affine Code
16	Hill Code
17	RSA Encryption
18	Elliptical Curve Encryption
19	Algorithms and Trace Tables
20	Functions, Hash Tables
21	Iteration, Fractals
22	Recursion and Induction
23	Searching/Sorting Algorithms
24	Computational Complexity
25	Network Topology: Traversability
26	Shortest Path
27	Shortest Connection

Each class is introduced in the context of its relevance to Information Technology. This approach is used to justify, encourage and motivate the learning. It is also hoped that students will see the usefulness of mathematics as a tool rather than as an isolated body of knowledge.

For example, Modulo Arithmetic

In Class 13, the mechanics of modulo arithmetic are introduced, so that encryption calculations can be confidently executed in later classes.

In Class 14, further practice in modulo calculations is provided and in particular using negative numbers. The lesson uses two applications as examples of modulo arithmetic:

- The calculation of check digits in bar codes and in ISBN numbers. The purpose of this class is to provide practice in modulo arithmetic calculations. Confidence in the arithmetic manipulation, particularly negative modular numbers, is necessary for the encryption calculations that are used in Classes 15, 16, 17 and 18. The class is started by discussing scanners at the supermarket checkout, their significance to the customer and to the company. We talk of their link to the central database. All students have a point of reference with supermarkets and are easily hooked into the discussion. Often there are students in the class who work at local supermarkets and can extend the discussion with extra information or comments. The history of the UPC and EAN-13 bar coding systems and their use around the world is outlined. Check digits are manually calculated using the two systems. Conversion of UPC codes to EAN-13 is shown. Calculation of ISBN numbers provides further practice in manipulating modular numbers. This also allows an opportunity to introduce a number of additional texts that provide alternative descriptions of course content, interesting background to the course and are readily available to our students in our library.
- The creation of a set of pseudo random numbers.John von Neumann’s famous statement “Anyone who considers arithmetic methods of producing random digits is, of course, in a state of sin.” (MacHale (1993) quoted in MacTutor History of Mathematics) provides contentious opening to the discussion of whether random numbers can be generated by a computer. The linear and multiplicative congruential methods are used to calculate sets of random numbers. Suitability for use as a set of random numbers and how it can be tested is outlined. The calculation of random numbers is referred to later in the course when looking at recursion.

For example, Functions

In Class 20, students are introduced to:

- Function notation, arrow diagrams, definitions of functions, one-to-one and onto relationships and inverses. The concept one-to-one relationships is introduced in the context of database tables; inverses in the context of encryption and decryption.

- Construction of hash tables using function notation.The first tables use linear probing in the event of collisions. The problems of clustering are discussed. Double hashing is presented as a solution to decrease clustering. The concept of collisions is tied to collisions within networks.

6.2 Our Teaching Approach

A curriculum encompasses not only the content to be covered but also the teaching approach and delivery.

Our curriculum caters for a broad range of students, from those with little secondary mathematics background to those with university mathematics papers. Many students come with significant experience of failure in mathematics and the ensuing emotional debilitating effects. To counteract any negativity, we employ a variety of pedagogical techniques that have been shown to be successful in the teaching of mathematics: an ethic of care, relationship building, social nurturing and confidence building, valuing student contributions, working in groups, and caring about the development of mathematical proficiency (Anthony and Walshaw, 2007). The focus of each class is on the development of mathematical understanding, proficiency and reasoning by providing on-going discussions and ensuring active participation. The pace is measured to ensure that there are opportunities for group work yet sufficient space for individual thinking.

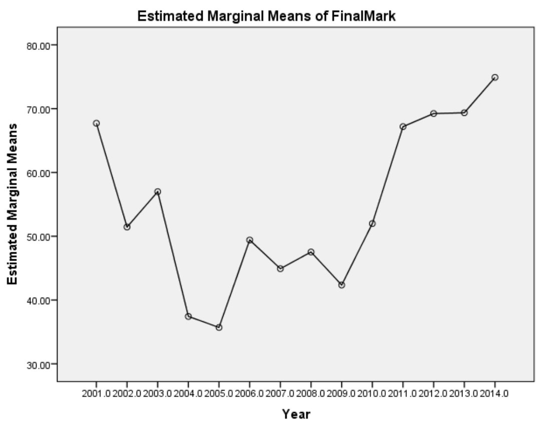
There is emphasis is on a language rich, teaching narrative, the articulation of not only the methodology but a massaging and soothing of the experience. By developing dialogue within the classroom, students refine and restate their understandings. They make sense of new or confusing information (Drake 2006). Narrative allows meaning to grow and change (Fincher, 2012). It provides a platform to play with the rational and emotional reactions to a mathematical problem and work constructively toward a solution.

By recognising and exploiting the interconnectedness of content and approach in a classroom environment, students are provided with authentic learning experiences where they can gain confidence and extend their mathematical understandings.

6.3 Student Performance and Feedback

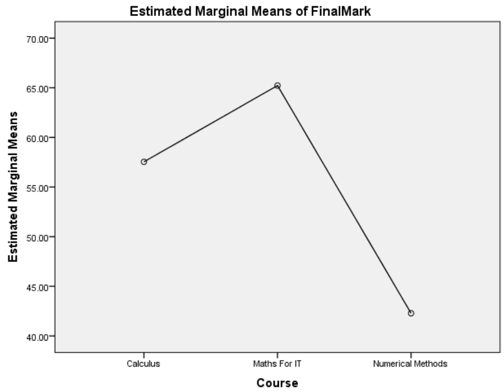
Completion rates over the last three years have been measured at 88%, 86% and 88%, which is consistently above the 80% target for the institution and compare well to other first semester papers.

The estimated marginal means of the final mark have been calculated against year and course type:



Final Mark			
Year	**Mean**	**N**	**Std. Deviation**
2001	67.7086	81	23.65944
2002	51.4409	127	35.24577
2003	56.9886	88	31.31101
2004	37.4306	72	29.72684
2005	35.6986	73	31.07459
2006	49.4118	51	31.08065
2007	44.9138	58	31.61073
2008	47.5254	59	34.20040
2009	42.3544	79	35.82902
2010	51.9861	72	30.97045
2011	67.1786	56	20.36316
2012	69.2373	59	22.63225
2013	69.3529	51	17.77732
2014	74.9024	41	17.99417

There is a significant effect of year on final mark (F=11.638; p<.001),



Final Mark			
Course	Mean	N	Std. Deviation
Calculus	57.5419	296	31.87509
Maths For IT	65.2258	279	24.57266
Numerical Methods	42.2857	392	32.57767
Total	53.5744	967	31.77050

There is a significant effect of course type on final mark (F=50.512; p<.001)

These results show that the mean pass rate for Calculus was 57.5% and Maths for IT is 65.2% and continuing to trend upwards. An initial attempt at Numeric Methods in the intervening years was less successful (mean pass rate of 42%) which highlights the difficulties inherent in making pedagogical change, in introducing more challenging content and the need for constant monitoring and fine tuning of curriculum.

Student feedback is collected via course evaluation surveys and informal interviews. Among the patterns observed are:

The contextualised approach appears to make mathematics more accessible to learners who have struggled with it in the past. Some student comments were:

“I used to hate maths until I enrolled in this course”, "I never thought that I could enjoy or understand maths",

"I struggled with maths at school and this course has been a revelation to me",

"It really helped me become less intimidated by math",

"It made someone who had a history of being poor at math and disinterested, interested and involved",

"This course is really interesting and enjoyable".

The course continues to be challenging for some:

"I have struggled with maths my whole life and felt that this paper was the most challenging aspect of the course so far",

"A very challenging but rewarding class",

"You definitely have to go to the classes to understand the content",

"It is a very challenging topic to cover".

Students identified aspects of the course that were meaningful or useful to their studies:

"It is obviously useful and relevant to the IT industry",

"It aided logical thinking. It certainly got me into the programming mind-set",

"It was really interesting, especially fractals",

"I liked learning new ideas with ways to solve problems",

"It was very helpful to understanding of networks and programming",

"I liked being able to talk among the class during a lesson, helps to figure out the math problem by group input",

"I can see how things I have been learning in class relate to future programming classes."

The course presents mathematics in a way that many students have not experienced before. While it is challenging for some, it contains aspects that are included to challenge the most able, yet there is sufficient that is achievable by all. The overriding aim is to provide a course that presents mathematics as a relevant and useful instrument in their current studies and their future employment.

7. CONCLUSION

Critical curriculum review is an on-going and vital component of any educational programme. The recent publication of the CS2013 guidelines has provided and excellent opportunity to reflect the programme against the ACM/IEE curriculum. The process has validated the shift we made from calculus to discrete mathematics in 2005 and our selection of topics that are engaging, relevant and contextualised. With each iteration, we continue to tweak, adapt and enrich the mathematics course. We aim to provide a broad rich educational experience that prepares our students within three years to enter the IT industry as a professional.

Future directions will include a more extensive review of mathematics curricula from other institutions, a survey of the specific use of mathematics throughout our programme and further analysis of the data collected from the "Maths for IT" course.

8. REFERENCES

Anthony, G. & Walshaw, M. (2007). *Effective pedagogy in mathematics/pāngarau: Best evidence synthesis iteration.* Retrieved from www.minedu.govt.nz.

Association for Computing Machinery (2013). *Computer Science Curricula 2013.* Retrieved from www.acm.org/education/CS2013-final-report.pdf

Auckland University of Technology. (2014). *Computer and Mathematical Sciences*. Retrieved from http://www.aut.ac.nz/study-at-aut/study-areas/computer-mathematical-sciences/undergraduate-degrees

Baldwin, D., Walker, H. M., & Henderson, P. B. (2013). The role of mathematics in computer science. *ACM Inroads 4*(4), 74-80.

Brown, J. S., Collins, A., & Duguid P. (1989). Situated cognition and the culture of learning. *Educational Researcher, 18*(1), 32-42.

Carnegie Mellon University. (2014). *School of Computer Science.* Retrieved from http://www.csd.cs.cmu.edu/education/bscs/index.html

Drake, C. (2006). *Using teacher narrative to understand teachers' uses of curriculum materials.* Retrieved from www.acm.org

Fincher, S. (2012). *Using narrative methodology.* Canterbury: University of Kent Press.

Gasson, J. (2004). Aligning our first year maths course to better meet student needs. National Advisory Committee on Computing Qualifications, (NACCQ). Proceedings of the *17th Annual NACCQ Conference*, Christchurch. 6 - 9 July. p. 495.

Gasson, J. (2005). Why calculus isn't relevant in tertiary information technology programmes. *New Zealand Mathematics Colloquium Contributors' Abstracts,* Massey University, New Zealand, 5 -7 December 2005.

Henderson, P. B., Baldwin, D., Dasigi, V., Dupras, M., Fritz, J., Ginat, D., Goelman, D., Hamer, J., Hitchner, L., Lloyd, W., Marion, B., Riedesel, C., & Wlaker, H. (2001). *Striving for mathematical thinking.* Retrieved from www.acm.org

Lave, J. & Wagner, E. (1991). *Situated learning: Legitimate peripheral participation.* Cambridge: Cambridge University Press.

MacTutor History of Mathematics. (n.d.). John von Neumann. Retrieved from www-history.mcs.st-and.ac.uk/ Mathematicians/Von_Neumann.html

Mayer, R. E. (2002). Rote Versus Meaningful Learning. *Theory Into Practice, 41*(4).

Otago Polytechnic. (2014). *We Love Otago Polytechnic.* Retrieved from www.op.ac.nz/study/business-tourism-and-it/computing-and-it/

Ralston, A. (2005). Do we need any mathematics in computer science curricula? *ACM Inroads 37*(2), 6-9.

Robins, A. (2010). Learning edge momentum: A new account of outcomes in CS1. *Computer Science Education 20*(1), 37-71.

Thompson, D., Bell,T., Andreae, P. & Robins, A. (2013) The Role of Teachers in Implementing Curriculum Changes. *SIGCSE'13*, March 6–9, 2013, Denver, Colorado, USA. 245-250.

University of California, Berkeley. (2014). *The Beauty & Joy of Computing*. Retrieved from http://inst.eecs.berkeley.edu/~cs10/fa14/

University of Otago. (2014). *Majoring in Computer Science*. Retrieved from http://www.cs.otago.ac.nz/student/major.php

Wing, J. M. (2001). *Mathematics in the computer science curriculum.* Retrieved from www.acm.org

Wing, J. M. (2006). Computational thinking. *Communications of the ACM. 49*(30), 33-35. Retrieved from www.acm.org

Wolz, U., & Conjura, E. (1994). Integrating mathematics and programming into a three tiered model for computer science education. *Inroads - The SIGCSE Bulletin, 26*(1), 223-227.