Jl. of Computers in Mathematics and Science Teaching (2018) **37**(4), 309-354

Innovative CAS Technology Use in University Mathematics Teaching and Assessment: Findings from a Case Study in Alberta, Canada

DANIEL JARVIS
Nipissing University, Canada
danj@nipissingu.ca

CHANTAL BUTEAU
Brock University, Canada
cbuteau@brocku.ca

CHARLES DORAN
University of Alberta, Canada
charles.doran@ualberta.ca

ANDREY NOVOSELTSEV
University of Alberta, Canada
novoselt@ualberta.ca

In this paper, we report on a case study that focused on innovative uses of CAS technology in university mathematics teaching and assessment. The study involved a site visit to the University of Alberta campus (June 2015) during which: interviews were conducted with five mathematics faculty members and seven mathematics students; math lectures were attended; and artifacts were collected such as course outlines, software demonstrations, and assessment tools. Interviews were transcribed and the data entered into *Atlas.ti* qualitative research software for the purpose of thematic analysis. Findings center around the innovative use of the open source, CAS-based software both in the teaching (answer checking, interactive lecture demonstrations) and assessment (student-generated optimization problems, mid-terms, final exams) practices of one particular instructor who taught seven iterations of a *Mathematical Programming and Optimization* undergraduate course.

Keywords: mathematics education; technology; *SageMath*; Computer Algebra Systems (CAS); teaching; assessment

INTRODUCTION

In this paper, we examine the findings from a case study conducted within a university mathematics department in western Canada in which a special Teaching and Learning Enhancement Fund grant allowed for the creation of a new combined graduate/undergraduate level course on computer-based experimental mathematics, as well as the restructuring of an existing undergraduate course on Optimization through the creative use of available CAS technology. One particular instructor incorporated Computer Algebra System (CAS)-based software into both his teaching and assessment practices within seven distinct iterations of the same Optimization course over time. We seek to provide insights into the perceived challenges and affordances relating to technology integration at the university level.

In this introductory section we begin by providing some background information by reviewing the literature regarding the use of instructional technology in mathematics education including issues such as teacher beliefs, behaviours, and changes to curriculum and assessment practices. We then explain the university context in which funding from special grants at the University of Alberta enabled two mathematics professors to initiate rich experimentation with the open source mathematics software known as *SageMath* (Stein & Joyner, 2005) within the two above-mentioned mathematics courses.

Literature Review

A growing number of international studies have shown that Computer Algebra Systems (CAS)-based instruction has the potential to positively affect the teaching and learning of mathematics at various levels of the education system, even though this has not been widely realized in secondary schools and in higher education (Artigue, 2002; Beaudin & Picard, 2010; Bray & Tangney, 2017; Bossé & Nandakumar, 2004; Kendal & Stacey, 2002; Lavicza, 2006; Meagher, 2012; Pierce & Stacey, 2004; Somekh, 2008; Smith Risser, 2011). Following the 17th International Commission on Mathematical Instruction (ICMI) Study Conference, entitled *Technology Revisited* and held in Vietnam in December 2006, then ICMI President, Michèle Artigue (2010), shared her insights regarding the resistance of instructional technology:

> The resistance to digital technologies, the incredible recurrence of debates on topics such as the famous long division quoted by Papert in his lecture, could be re-interpreted in terms of balance between epistemic and pragmatic values. . . . Making technology legitimate and mathematically useful requires modes of integration allowing a reasonable balance between the pragmatic and the epistemic power of instrumented techniques. This requires tasks and situations that are not simple adaptation of paper-and-pencil tasks, often tasks without equivalent in the paper-and-pencil environment, thus tasks not so easy to design when you enter in the technological world with your paper-and-pencil culture. . . . [T]hinking in such terms changes one's mind, obliges one to look at educational resistances differently, and obliges one also to question the resources that, as researchers, we provide to teachers and institutions for overcoming these difficulties. (pp. 467-468)

Clearly, the incorporation of new and powerful technologies requires a careful and deliberate rethinking of mathematics curriculum, learning goals, tool/software usage, pedagogical strategies, and assessment practices at all levels of education.

> In addition to its computational power, modern technologies can help increase collaboration and bring about more of an emphasis on practical applications of mathematics, through modelling, visualisation, manipulation and the introduction of more complex scenarios. . . . For these reasons, the use of technology in mathematics education is becoming increasingly prioritised in international policy and curricula. (Bray & Tangley, 2017, p. 256)

Building upon Shulman's (1986) work on Pedagogical Content Knowledge (PCK), Koehler and Mishra (2009) have developed their own *Technology, Pedagogy, and Content Knowledge* (TPACK) model for the analysis of teacher practice:

> TPACK is the basis of effective teaching with technology, requiring an understanding of the representation of concepts using technologies; pedagogical techniques that use technologies in constructive ways to teach content; knowledge of what makes concepts difficult or easy to learn and how technology can help redress some of the problems that students face; knowledge of

> students' prior knowledge and theories of epistemology; and knowledge of how technologies can be used to build on existing knowledge to develop new epistemologies or strengthen old ones. . . . Teaching successfully with technology requires continually creating, maintaining, and re-establishing a dynamic equilibrium among all components. (Koehler & Mishra, 2009, pp. 66-67)

Another informative taxonomy for understanding the different uses of technology in mathematics instruction is the Substitution Augmentation Modification Redefinition (SAMR) model created by Puentedura (2006, 2014). Although the SAMR model has been criticized on several points (diverse interpretation/application of the model; an absence of context; an overly rigid structure; and emphasizing product over process), it remains an increasingly popular tool for practitioner reflection and planning (Hamilton, Rosenberg, & Akeaoglu, 2016). Similarly, Bruce (n.d.) has developed a detailed taxonomic model focusing specifically on varied Interactive Whiteboard (IWB) use within mathematics instruction.

Somekh (2008) described this difficult yet required paradigm shift relating to instructional technology as follows:

> The pedagogical adoption of ICT is complex and requires an integration of vision, system-wide experimentation and new roles and relationships for teachers and students. . . . The affordances of the Internet, digital photography and cyberspace are radically changing how knowledge is constructed, represented and accessed in the world outside school, and policy-makers need to acknowledge this and restructure the systems of curriculum, assessment and school organisation. (p. 458)

School level studies suggest that beyond the availability of technology, teachers' beliefs and cultural influences are key factors in technology integration into mathematics teaching and learning. In categorizing barrier types as either first-order (external) or second-order (internal) in nature, Ertmer et al. (2012) have provided a helpful set of related definitions:

> First-order barriers were defined as those that were external to the teacher and included resources (both hardware and software), training, and support. Second-order barriers comprised those that were internal to the teacher and included teachers' confidence, beliefs about how students learned, as well as the perceived value of technology to the teaching/learning process. Although first-order barriers had been documented as posing significant obstacles to achieving technology integration . . . , underlying second-order barriers were thought to pose the greater challenge. (Ertmer, Ottenbreit-Leftwich, Sadik, Sendurur, & Sendurer, 2012, p. 421)

Teacher beliefs represent deeply-held assumptions and values relating to education, and these beliefs are not easily changed. As Bray and Tangney (2017) note, "In order to achieve an environment that facilitates technology usage in an inquiry-based, constructivist manner, a change in the pedagogical approach and the learning experience of the students is required, and this is fundamentally dependent on the actions and beliefs of teachers" (p. 257).

In contrast to the growing body of research focusing on CAS technology use at the secondary school level (Connors & Snook, 2001; Fey, Cuoco, Kieran, & McMullin, 2003; Haapasalo, 2013; Kieran & Drijvers, 2006), there is relatively little parallel research at the post-secondary level (Buteau & Muller, 2014; Decker, 2011; Martinovic, Muller, & Buteau, 2013; Rosenzweig, 2007; Stewart, Thomas, & Hannah, 2005; Tobin & Weiss, 2016; Thompson, Ashbrook, & Musgrave, 2015; Thompson, Byerley, & Hatfield, 2013; Tall, 2013). This is particularly true in the area of student *assessment*, where powerful technology tools such as CAS computer software and CAS-enabled calculators have rarely played a part in formal evaluation in undergraduate mathematics courses (Heidenberg & Huber, 2006; Sevimli, 2016).

Lavicza's comprehensive study (2008a, b) featured an online survey of 1100 mathematicians as well as interviews with 22 mathematicians in three countries, namely, Hungary, United Kingdom, and United States, which examined mathematicians' beliefs/conceptions regarding CAS and its instructional potential. Findings showed some similarities, but also notable differences, between university- and school-level research findings (e.g., use of CAS in one's research being the greatest factor influencing the use of CAS in one's teaching).

Building on the findings from Lavicza's international work, the team of Jarvis, Buteau, and Lavicza implemented a mixed-methods research study to examine individual and systemic CAS usage in undergraduate mathematics instruction. This research program involved an extensive literature review, a national survey of Canadian mathematicians, a multi-site case study of two technology-enhanced mathematics departments (Canada; United Kingdom), and the hosting of two national workshops at premier Canadian research institutes in both Quebec (in French) and Ontario (in English). Based on their findings, they concluded that: Instructor beliefs regarding the nature of mathematics learning, required curriculum/assessment changes, the use of technology in one's own research, and the availability of resources are among the complex set of factors that affect the degree to which technology is implemented within undergraduate university mathematics courses. The Canadian survey of over 300 participating mathematicians clearly indicated that many professors were using CAS in their instructional practice (69%), and also reinforced the Lavicza finding that the greatest factor influencing the use of CAS in one's post-secondary mathematics teaching was the use of CAS in one's own research (Jarvis, Lavicza, & Buteau, 2012).

TLEF Project and McCalla Professor in Science Chairship

Dr. Charles Doran, professor in the Department of Mathematical and Statistical Sciences at the University of Alberta (UA) and Site Director of the Pacific Institute for the Mathematical Sciences (PIMS), received internal funding by way of the *Teaching and Learning Enhancement Fund* (TLEF). In conjunction with the TLEF funding (2013-16), he was also named to the position of McCalla Professor of Science Chair. As part of this latter recognition, Doran had created an integrated teaching and research plan which involved the writing and delivery of a new upper level, joint (graduate and advanced undergraduate) computing and mathematics course entitled *Computing in Mathematics: Research via Experimentation* (MATH 497). Project funding also allowed him to focus research on an existing third year *Mathematical Programming and Optimization* (MATH 373) course in which *SageMath*, an open source mathematics software program, was being used by a Post-Doctoral Fellow, Dr. Andrey Novoseltsev, in new and creative ways in terms of mathematics teaching and assessment. A case study by Jarvis and Buteau was approved and conducted at the University of Alberta, the results of which form the basis of this paper, with a particular focus on what was being undertaken by Andrey in the undergraduate MATH 373 course.

SageMath Software

When asked why they chose to use open source Sage[1] software and to continue to develop their own math apps using Sage, rather than using other available commercial software and apps, Charles explained the following factors:

> Sage has the advantage of moving us away from proprietary systems, and that's good and bad. The good is that we can start fresh and build from the ground up so that we can control

[1] Created by Dr. William Stein, *SageMath* (http://www.Sagemath.org/) originated at the University of Washington, but now represents an international project with many developers in dozens of different countries. *SageMath* is a freely available, open-source mathematics software system licensed under the General Public License (GPL). Since participants in this study, including the instructors that were interviewed, commonly refer to the software as simply Sage, we have used this shorter title throughout the paper for consistency.

> everything about the look, and the feel, and the capacity. It's problematic when we're talking about a course, say, in differential equations where the applied mathematicians have always used Maple, or a course in the engineering school, or for engineering students really where they're so immersed in the MATLAB world that they don't want to leave that setting. . . . Mostly, I just think the price is right, the community is like-minded individuals who want to make everything better, and who are really willing to work with you to do so, and the overall vision set by Stein is solid, I mean, you know, he's looking ahead to both scaling up and tailoring the resources.

Andrey produced approximately 12 applets (i.e., very small applications, often utility programs that perform one or a few simple functions) for different mathematics courses (see Figure 1), which represented one of their important goals for the TLEF project. These apps were uploaded to a shared drive so that other professors could have access to them, and so that students through different courses also had access to them, and they tracked how often and when these applets were used.

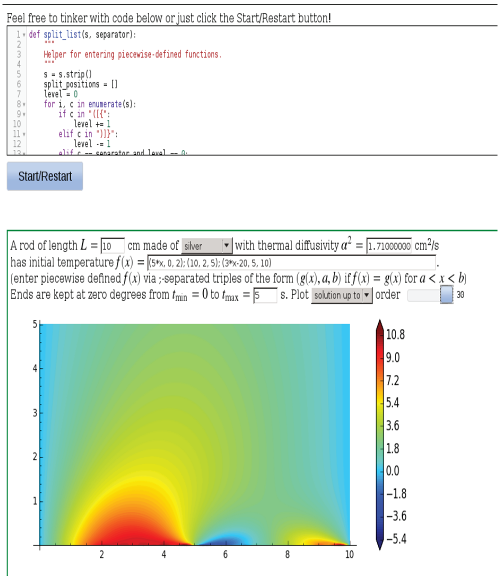

Figure 1. Heat equation applet created by Novoseltsev using Sage software.

In discussing unintended consequences of the TLEF project, Charles noted that, in his opinion, the biggest unexpected aspect was figuring out where and how to host the actual apps for student and instructor access.

> Where do we put it? Do we put it on the machine in the PIMS office, which was our original idea, or host it somewhere on campus, and that led to all sorts of problems involving access and students ID issues. . . . [T]here was this development of Sage Cloud, and Andrey realized rapidly that we could actually take the environment that the students were working in and simply have them do it on their browsers, in the Cloud. . . . Finances being what they are—everyone would like to see it go somewhere free. . . . So, I would say that the technical hardware aspects of this story are ones that I had not expected. . . . I thought that would be dead easy. We'd buy a machine, we'd plug it in, Andrey would spend a week tinkering with it, and we'd be done. I was very naïve about that.

In discussing the choice of Sage software, Andrey explained his rationale: "I like the fact that *SageMath* software is free for students, and that they can also access it from home. . . . I strongly believe in the open source approach for mathematics software . . . you have the option to look at the code and to fix the bugs—try to make improvements." Once Sage was established as the software of preference for the TLEF initiatives, specific plans began on how to incorporate it into the new *Research via Experimentation* course and into the existing *Mathematical Optimization and Linear Programming* course in terms of both potential instructor and student use of the open source software.

Mathematical Programming and Optimization (MATH 373) Course

The *Mathematical Programming and Optimization* (MATH 373) course had already existed within the department, and had been taught a number of times by Andrey (2011-13) before the TLEF funding began to be used for this initiative in 2014. For sake of general context, what follows is the Course Description from the Andrey's syllabus from the Spring 2014 term.

> MATH 373: Introduction to optimization. Problem formulation. Linear programming. The simplex method and its variants (revised simplex method, dual simplex method). Complementary slackness and duality. Extreme points of polyhedral sets. Theory of linear inequalities (Farkas Lemma). Post-optimality analysis. Interior point methods. Applications (elementary games, transportation problems, networks, etc.).

Pre-requisites for this course were listed as first year linear algebra course, as well as any 200-level math course. In other words, this third year level course required students to have some background in mathematics, but not any programming experience. Course objectives and expected learning outcomes for MATH 373 were described in the syllabus as follows:

> After taking this course, you should be able to formulate a linear programming problem and convert it to the standard form(s); understand the structure of dictionaries of the Simplex Method and their relation to the original problem; perform steps of the Simplex Method (and its variants) and understand why they lead to the solution; use relations between dual problems to efficiently verify optimality of solutions and to construct certificates of solutions; detect inconsistent and redundant inequalities in a system; modify optimal solutions to take into account changes in constraints and objectives. You will also develop a general sense of what optimization problems are, see "linear algebra in action," and pick up basics of using math software and typesetting mathematical expressions in LaTeX.

In interviewing Charles, Andrey, other mathematics instructors, and mathematics students at the University of Alberta, we quickly became convinced that our primary focus would be on the MATH 373 course and how Andrey had developed strategies involving technology for teaching and assessment through seven different iterations of this course. Follow-up interviews allowed us to inquire into how the MATH 373 course had changed in further offerings of the course subsequent to the TLEF funding window and to our campus site visit in Alberta.

RESEARCH QUESTION AND METHODS

The main research question that guided this study was as follows: What are the perceptions (e.g., benefits, barriers, other observations) of key stakeholders (i.e., project leader, instructors, students) regarding the implementation of technology-enhanced (SAGE applets) mathematics courses which involve new forms of curriculum and assessment practices?

The case study research project involved 14 interviews with the following participants: Dr. Charles Doran, TLEF Project Leader (twice); Dr. Andrey Novoseltsev, the Course Instructor (twice); two other math faculty members; six students; a course grader; and a former UA student who had become a math professor at another institution. These interviews took place both in Ontario during the Canadian Mathematics Society (CMS) Winter conference in December 2014, and (mostly) during a site visit to the University of Alberta campus in early June 2015. The interview questions were semi-structured (i.e., open-ended) and were designed and implemented according to qualitative case study standards (Creswell, 2013; Denzin & Lincoln, 2005; Yin, 2009). Follow-up email correspondence with Charles and Andrey provided further data for the study.

During the site visit, artifacts were also collected regarding course outlines, software applets, course assignments, and assessment tools. The interview data was entered into *Atlas.ti* qualitative research software for the purpose of data organization and thematic coding of the interview transcripts. Thematic analysis was used with data and this involved a process of coding with the following six phases to create established, meaningful patterns: familiarization with data, generating initial codes, searching for themes among codes, reviewing themes, defining and naming themes, and producing the final report. During this process, 34 distinct codes were originally identified, and from these, seven larger code groupings were established: Assessment, Instructor, Learning, Optimization, Sage Software, Technology, and the TLEF Project. Drawing upon the content found within these thematic strands, we have organized this paper primarily around the chronological delivery of the MATH 373 course, followed by subsequent discussions relating to mathematics teaching, learning, assessment, and departmental sharing.

While the actual names of Drs. Doran and Novoseltsev are being used directly in this paper, by permission, the names of students that were interviewed during the site visit shall be replaced with the following alphabetical order pseudonyms, as per the letter of informed consent: Akemi, Brittany, Cheng, Dawn, Ezra, Felix, and Guang. In addition, the code names Xavier and Zack will be applied to two other mathematics instructors. This study was approved by Nipissing University Research Ethics Board in Ontario, and was similarly approved by the University of Alberta, the host institution.

FINDINGS

In this section we begin by describing the seven iterations of the *Mathematical Programming and Optimization* (MATH 373) course in which Andrey incorporated technology into his teaching and assessment practices in ever-changing ways, and with a variety of intended and unintended results. We then discuss how this technology was perceived by participants as affecting the teaching, learning, and assessment of mathematics course content.

Seven Iterations of the Technology-Enhanced Course

The *Mathematical Programming and Optimization* (MATH 373) course offered within the Department of Mathematical and Statistical Sciences at the University of Alberta was taught seven times by Dr. Andrey Novoseltsev from 2011-15, before, during, and after our case study site visit took place. Based on interviews with Andrey and his students and colleagues, as well the examination of course related documents, we track the progression of technology use within this particular course, in terms of both teaching and assessment practices that were being developed and monitored throughout these course sessions.

During the interviews, Andrey described his first use of Sage technology in the MATH 373 course, having taken place during Fall Term of 2011. At this early stage, Sage software was being used primarily by Andrey as a powerful calculating machine for checking answers, and occasionally for class demonstrations during this first iteration of the course. By the fall of the next academic year in 2012, Andrey was now beginning to create applets using Sage in advance of his lectures so that these applets could then be used to expedite lengthy calculations, based on student suggestions during class. The imagery from his computer screen was projected onto a screen using a data projector, and thus provided his students with immediate feedback on their volunteered ideas. Andrey describes his satisfaction with this process as follows:

> I was coding and preparing parts that we needed before classes. I would show the computations in class . . . much faster than if I was doing it by hand, or on the board. What I really liked was that I was able to get input from students, "What do we do next?" And I could follow their suggestions even if they were wrong because the algorithm would show you that they had done something wrong.

Beyond classroom demonstrations in which Sage was used as a powerful calculation and exploration tool, Andrey also began making Sage worksheets with additional related commentary available to his students online following classes.

In the third iteration of the course, Andrey continued to use a strategy during lectures in which he asked for student input while using Sage software applets and digital worksheets to calculate or model particular questions. The major shift in this third iteration was that he began to have students create their own personalized linear programming problem for the second assignment (see Appendix A), and to then use these problems to apply different linear programming methods throughout the remaining course assignments, such as *A3: Simplex Method* (see Appendix B).

> Students had to each come up with their own word problem at the beginning of the course that would be convertible to a linear programming problem, and that would have a sufficient size and sufficient number of constraints, and enough variables. . . . What I was doing was giving them a template to help them set up a problem . . . I recommended that they read through several problems and go online and just get a feel for what kind of word problems could be converted to these linear programming problems that we were working with. A lot of them were about manufacturing problems, where you are producing so many goods, from so many ingredients, then you want to maximize your profits when you know the price of ingredients and what the product is. Mathematically, they are not all that different. . . . Throughout the course, they would apply new techniques that we were covering to the same problem that they had originally posed in Assignment 2.

Although students would now be working on their assignments individually, Andrey did begin using an approach that involved the grouping of students, using the department's Learning Management System (LMS), for the purpose of online assignment sharing and peer review and commentary. Overall, he found that this grouping and sharing system worked well, although there were some who failed to engage. Andrey did allow for some variation in groupings, telling his students: "If you are particularly unhappy in your group, and think that you put a lot of effort into it but your group members did not, then you can try to see which groups you like and I will move you there."

In terms of assignment submissions, Andrey had not required his students to use Sage in their assignments. In so doing, he allowed students to solve their problems either by hand, or by using other proprietary software such as *Maple* or *MATLAB* that they may have learned in other math courses. However, what he did require was that they submit their draft and final versions of all assignments as typeset Portable Document Format (PDF) files, in order to avoid scanned, hand-written solutions which were often more difficult for fellow students and the grader to read and understand. In other words, students needed to type out their answers using commercial software such as *Microsoft Word* with built-in *Equation Editor*, or with open source options such as *Sage*, *LaTeX*, or *Libra Office* to render the alpha-numeric and math symbol characters.

During the fourth installment of the MATH 373 course in the Fall 2013 term, Andrey decided to now *require* that his students use Sage for certain assignments. According to Andrey, very little knowledge of computer programming was actually needed to learn how to use Sage applets: “Basically, students only need to know how to enter command lines in Sage. There is no real programming involved. . . . If any students have taken any introductory Computer Science courses here, then they would know Python, and Sage uses Python.”

Access to the Sage software was also now changed to an online approach, with passcodes provided so that students could access the software either from within the university campus, or off-campus, as long as they had Internet access.

> I was hosting a separate installation of Sage on one of our university computers, and so students would connect up from wherever. . . . By looking at IP addresses accessing the site, I could see that most of them were working outside campus. . . . It was very good to have a system that allowed them to do that.

Another important note about the fourth installment of the MATH 373 course was that Andrey made the significant choice of introducing required Sage software use into formal course tests.

> One of the complaints before that was, “We’re using all this stuff to solve problems with Sage, but then on tests we don’t use this knowledge.” . . . I didn’t want to sacrifice too much time, so the tests were 40 minutes each. Usually they had two problems, and so for this test which involved the computer lab, one problem was such that a computer was not particularly useful, so they really needed to write their own arguments . . . they could use the computer to play with, or to verify their work, but the solution had to be hand-written. . . . The other question was one where the computer was extremely useful, and I don’t think that problem was even doable by hand.

In terms of logistics for this test, students were seated at computer lab desks that allowed enough flat space to write on their papers, as well as to access the computer keyboard. The lab that Andrey used could seat 120 students, so the approximately 70 students in his class were able to be physically spread out, which reduced the potential for copying and chatting. In order to address the issue of Internet browsing and peer communication during the test, Andrey adopted a number of additional creative strategies. First, he arranged to have the main server shut down for the lab so that students could not access the LMS system which housed their shared homework assignments, even from the nearby washrooms. He also required students to switch their browser windows to full-screen mode, as opposed to just maximized windows, so that they could not access any other programs during the test. Finally, he customized the style of the digital worksheet used on the test so that it had a distinctive colour pattern which was pale enough that it did not interfere with the students' ability to read it, but could be easily monitored visually from the back of the computer lab, ensuring that all students were indeed working on the prescribed Sage worksheet.

Based on the overall success of the technology-based test experience, Andrey began preparing a final examination that would also feature a computer lab experience that required the use of Sage software. To help reduce the number of questions for technical assistance during the exam, he more carefully prepared students in this regard prior to the exam date by making related procedural announcements in class and via email messages.

> They were supposed to work on their assignments this way, so I think there was no excuse for them not to know how to do it on the final exam. . . . I strongly recommended to them to press the Save button on the top of the worksheet during the exam. . . . At the end of the exam, they would just shut it down so that all access would be terminated, and then later on I would be able to log in as the administrator to this installation of Sage on my office computer, and then I could see the worksheets for assessment. Sharing these instructions with my students beforehand greatly calmed them down. I still had a few questions during the exam, but it went quite smoothly.

The final examination was written in a smaller computer lab that had an opened partition wall and adequate seating available for 60 students at individual computer stations (i.e., some students had dropped the course by the time of the final examination). Two proctors were enlisted from the math department to help monitor the exam, and Andrey moved back and forth between the two sides of the lab to answer questions about the material. Andrey explains the content of this first exam:

> There were about eight problems on the final exam, and half of them were supposed to be done with computers—some completely on computers; others with part to be done on computer, and another part done by hand; and some problems which mostly were to be done by hand. So, again, I did not tell them which ones had to be done by hand, or with the computer, but I think it was quite clear when they should use it.

Up until this point, Andrey indicated that he had been giving his students very simple problems with simple steps, so they could do it themselves by hand, or that he had given them slightly more complicated problems, but had asked them to do just one or two steps. But what he was doing by Fall 2013 was giving them bigger problems involving random coefficients and problems that one simply *could not do by hand*. Andrey would require students to solve these problems completely, usually requiring five or more steps, and then answer questions about analyzing the results. Clearly, this was a rather bold and uncomfortable move for Andrey, to venture into the area of computer-assisted undergraduate mathematics final examinations. Based on this first experiment, and on observed student reactions and achievement, Andrey would prepare to repeat this approach to summative assessment in the same course the following spring term.

According to Andrey, very little changed with the fifth installment of the Optimization course. Enrolment had increased to 100 students (up from 50 only a few years before), 90 of which remained in the course after the drop deadline. Due to this larger number of students and the unavailability of the large computer lab for multiple sessions, Andrey decided to once again replace the three term tests with one larger mid-term exam, followed by the final exam at course end.

Quite a number of students still seemed to be having difficulty in understanding the technical aspects of how to use the LMS system to access, complete, submit, and share assignments online. To address this perennial problem, Andrey adopted two new strategies. First, he offered extra, optional computer labs at the beginning of the course which he tried to schedule so that students could attend independent of their respective timetable constraints. In these sessions, he would offer individual or small group assistance on how to gain access to the system, how to enter commands and obtain outputs, and how to save their work. With 90 students, and with the course being offered in the Spring/Summer Inter-Session at a more rapid than normal term pace (i.e., fewer weeks), the instructor found that he just could not schedule enough of these optional labs during the first three days of the course, so he then made the decision to also begin creating and sharing screencasts of this material in which he offered step-by-step instructions as well as text-based files that would reinforce the new procedural learning regarding LMS navigation and Sage worksheet access and manipulation.

The assignments for the course still involved the review of the LMS/Sage guidelines in the first assignment, as well as the creation of a student-generated optimization problem in the second assignment that would be revisited using different methods of analysis throughout subsequent assignments. Students were once again asked to submit drafts of their assignments online, to review the drafts of their peers within assigned groups, and to submit final copies of assignments for grading. Both the mid-term and final examination again involved a combination of question types including those that demanded full, hand-written responses; those for which Sage could be used for exploring and checking; and those that absolutely required that Sage software be used to complete them.

Dr. Novoseltsev did not teach the MATH 373 course in the Fall Term of 2014, but did return to the course in the Spring Term of 2015. It was during the implementation of this sixth version of the course that our site visit took place, shortly following the mid-term examination. By 2015, Andrey had created more and improved screencast video tutorials for familiarizing students with the Learning Management System (LMS) and with course requirements involving Sage digital worksheet technology. Overall, he found the screencast sharing approach to be more effective than scheduling small groups of 10 students at a time in lab spaces early in the course since all students were then able to view and review the instructional videos, with pausing as needed, and at their leisure wherever Internet access could be found. The instructor continued, as always, to maintain posted office hours, often making himself available in the computer labs as well. For certain assignments he continued to require students to tackle problems that involved quite large calculations that were often impossible to do by hand, thus encouraging the use of the Sage technology. In other words, he believed in a balance between reinforcing hand-written calculation skills along with the regular and targeted use of Sage technology where appropriate.

Students were again required to create a unique, personalized optimization problem for their second assignment that was difficult enough in nature that it would require fairly complex formulas, and which could be revisited and applied to the different methods being learned throughout subsequent assignments. Students also wrote both a mid-term examination (see Appendix C) and a final examination that involved Sage applications. When Andrey last taught MATH 373 in the Fall 2015 term, he noted that there were not many changes made to his teaching and assessment practices for this final iteration. He reported making some further adjustments and improvements to the various applets to allow for customization of output by certain interested colleagues, and including some choices to match different existing mathematics textbooks.

Having normalized, over four years, the use of Sage software for class lecture demonstrations and explorations based on student input; for use in student-generated and revisited optimization problems; and for required use in course assignments and assessments, Andrey had become more and more convinced of its relevance and utility for this particular course. Table 1 shows an overview of the advancement of technology-related strategies and tools that Andrey had introduced over the seven installments of the *Math-*

ematical Programming and Optimization (MATH 373) course as described above.

Table 1

Overview of how Sage software was incorporated into the MATH 373 course over time.

Installment	Sage Technology Used in Instruction	Sage Technology Used in Assessment
1. Fall Term 2011	▪ Wrote simple code to check his own lengthy calculations during the course ▪ Minimal use for class demonstrations ▪ Gave students access to it; very few used it	▪ Used applets to prevent exam question calculation errors in marking written papers ▪ Grader used to mark student work
2. Fall Term 2012	▪ Sage applets written for most topics ▪ Demonstration: Frequently used Sage in class lectures to incorporate student suggestions and to provide immediate feedback on overhead screen	▪ Sage worksheets with additional related commentary created and made available to students via the university LMS for completing assignments and for review ▪ No grader used to mark student work; cheating noticed on submitted assignments
3. Spring Term 2013	▪ Student-generated linear programming problems are required, and are to be used throughout the course in various modules ▪ Sage used to explore these problems	▪ Students read/respond to fellow student problems online in assigned peer groups ▪ Sage optional for assignments (could also do by hand, or using other CAS software)
4. Fall Term 2013	▪ Sage moved to online platform with secure passcodes making it more accessible ▪ Sage now required for some assignments	▪ Sage required for three course tests, which led to implementation of strategies to prevent cheating during test writing in computer lab ▪ Required Sage use on final exam in lab
5. Spring Term 2014	▪ Extra, optional computer lab tutorials offered to small groups of students to familiarize them with Sage and LMS ▪ Created a step-by-step video for this also	▪ Three term tests with Sage replaced with one mid-term examination using Sage ▪ Mid-term/Final exams feature hand-written, optional Sage, required Sage use questions
6. Spring Term 2015	▪ Created more and improved online videos for LMS and Sage technical instructions	▪ Short exercises for students to explore included in his LMS posted lecture notes ▪ Mid-term/Final exams feature hand-written, optional Sage, required Sage use questions
7. Fall Term 2015	▪ Created additional applets for demonstration and teaching purposes	▪ Mathematicians at other universities become interested in Simplex Method applet ▪ Mid-term/Final exams feature hand-written, optional Sage, required Sage use questions

Technology in Teaching and Learning

In this section, we will explore the emergent theme of technology in teaching and learning by focusing first on participant perceptions relating to Andrey's instructional strategies, and then on their perceptions regarding the actual understanding of mathematical concepts in light of increased technology use within this specific undergraduate mathematics course.

Participants in the study were asked to describe the instructional strategies of Andrey in terms of his general teaching methods and the perceived effectiveness of these practices. During the site visit, the two researchers were able to attend several of Andrey's lectures in person to observe these methods and the overall classroom environment.

Instructor Policy on Classroom Technology Use

In order to deal directly with the issue of student distraction with social media and the Internet, Andrey had adopted a policy of not allowing any cellphones or laptops in his classroom during regular lectures.

> **Brittany:** He doesn't allow technology in class—no phones or laptops. He'll stop the class and say, "Put it away." You do not lose marks, but it's just embarrassing. . . . Yes, I agree with his policy. . . . I don't have social media—I don't have that stuff, so it's not distracting to me but I know if I had it, it would be.
>
> **Cheng:** I personally agree with my professor, like, you shouldn't have your phone out in class. . . . . Or what's the point of coming to class? So, I do agree with my teacher—put the phone away and listen to the lecture.
>
> **Dawn:** Oh yes, definitely it is effective. Some profs say, "I allow you to use your laptop, but just don't go on Facebook and don't use social media," but it still happens. So this kind of rule where it's like, "No technology whatsoever," just reinforces the learning environment. You can't really take notes on a laptop, especially in a math class. You're not going to enter each variable, so I think that for this class, it's completely fair and it helped a lot.

One interviewed student took issue with this policy, but most were actually quite satisfied with this directive, whether or not it affected their own personal behaviours in class.

Instructor Knowledge and Pedagogical Practices

In observing Andrey's classroom lectures while on-site, we noted that he primarily taught from his desk, with a computer and keyboard connected to a data projector, and using a screen to show the Sage outputs, and also writing on a whiteboard as needed. When asked about their perceptions of the instructor's knowledge of the subject matter and software, students offered insights into Andrey's combination of alternating whiteboard and keyboard work during most class lectures.

> **Brittany:** He knows the material, because he doesn't have notes with him—he does it by memory. . . . He's usually teaching

> something new on Sage . . . And dealing with technology difficulties, he's always answering questions. . . . It's something I appreciate because he actually engages . . . It's not just, "Here's a program, go figure it out." He actually kind of walks you through it, and makes sure you understand. . . . If someone asks what happens if you change, like, a constraint he will actually go and show you, and then you get to see it in class. . . . He has really detailed instructional videos that he personally uploads on eClass, and those were really helpful.
>
> **Cheng:** He has to be really familiar with the software, because if he has difficulty to access it, we're going to be super confused. I think the professor has to be an expert. . . . I think he did a really good job.
>
> **Dawn:** He knows when to stop and bring it to the computer, and then show us the real-time example. Again, for theorems and for definitions he knows he should write these down in word form, and that's how we understand it better. . . . I think that he did a pretty good job of organizing the format of the course . . . it was well executed.

Students clearly viewed Andrey as a professor who not only knew the mathematical content of MATH 373, and application of this content using Sage technology, but also one who had found a positive way to combine whiteboard notation, student interaction (questions/suggestions), and keyboard applet demonstrations for effective class lecture sessions. In comparing lecture notes that he use to use to the "messier, real-world" math problems eventually adopted, Andrey noted:

> A big problem why lecture notes are considered only artificial and disconnected from real examples, is that real-world problems tend to give too many variables, and too many constraints, and so they are very difficult to solve using our by-hand calculation techniques—you just get drowned in this matrix. . . . So, you need this kind of powerful software like Sage to handle these messier, real-world problems. . . . They still need to know the steps that they need to follow to get the solution, but there are some steps where it's not complicated from a conceptual point of view in any way, but just involves maybe a lot of addition and multiplication.

When asked how he thought this balance of using lecture notes and technology explorations during Andrey's classes affected student learning, Charles maintained that "Clearly the students know more. I mean they have much better understanding of what's going on in applying optimization in doing it this way then they would have the other [hand-written] way."

Mathematics Learning with Technology

Instructors and students both had a great deal to say about how they felt instructional technology impacted the actual learning of the mathematical content of the course.

> **Cheng:** I have to use Sage every day to review, and also to preview, and to learn things. . . . In Sage, we can actually see the process. Sometimes when you do it by hand, you're not actually sure what you did, if it's right or wrong. . . . I believe that for optimal solutions, we need to use something like computers. . . . It's not that the problem cannot be done by hand, it just seems useless and a loss of time to do it by hand.
>
> **Dawn:** It helps you understand, it helps you practice what you've learned in terms of the actual course content. It actually helps you put it into real-life usage. I'm pretty sure that mathematicians don't sit at their desks all day computing these commands by hand. They actually would use computers, and so it kind of gives you a bit of a sneak peek into possibilities of a future based on mathematics. . . . An Optimization course could happen without technology, yes, but I feel like it wouldn't be as efficient—and I don't think it would be as fair because you're not getting 100% out of the math, do you know what I mean?
>
> **Ezra:** I couldn't imagine learning this course without any current technology. . . . Computers are better in some areas—far better than human beings. . . . we can shift the focus to the understanding part.

Using Sage for complex problem answer checking and for class lecture demonstrations both represent instructional strategies that are frequently encountered at the secondary and university levels in the research literature. What is far more rare, particularly at the post-secondary level, is the use of technology in formal assessment practices. It is to this area that we now turn, as we focus on how participants interpreted Andrey's use of Sage in his various assessment strategies.

Technology in Assessment

The role of technology, particular the use of Sage software, clearly evolved over the seven iterations of the MATH 373 course taught by Dr. Novoseltsev, both in the ways in which he conducted class lectures (as discussed above), and also how he approached all forms of course assessment including assignments, tests, mid-terms, and even final examinations. Charles was particularly pleased to see Andrey take such initiative in incorporating the technology in these various ways over time.

> In the context of Optimization [MATH 373], it's gone from traditional assessment to entirely computer-based and Sage-based assessment. . . . The kind of questions that one could ask in an optimization course, and expect the students to work out, either on homework or the exam, went from being atypical for applications of optimization in the real-world to being typical. . . . Andrey did develop some sort of project component as well for optimization. . . . I'm pretty sure it really is the case that the kinds of question being asked now are much closer to real-world applications of optimization. . . . Given the nature of what optimization does, the successful optimization student is the one who can run big optimization problems on the computer, because no one does the Simplex Method by hand.

When asked about their general perceptions of their instructor's use of Sage technology in his formative and summative assessment methods, students shared both approval and some minor elements of frustration regarding their experiences. In what follows, we will look at comments regarding three specific assessment strategies that Andrey had developed: (i) introductory assignment and screencast video tutorials; (ii) assignments that involved student-generated, revisit linear programming problems; and (iii) mandatory Sage use in course tests and on mid-term and final examinations.

Introductory Assignment and Screencast Video Tutorials

Andrey had created an introductory assignment that introduced basic navigation of the Learning Management System (LMS) and familiarization with the Sage interface and common pre-existing commands. He would later go on to create comprehensive screencast video tutorials for his students in subsequent versions of the course:

> **Dawn:** When I first got that assignment I was really scared. I thought I was going to drop the course because I just was not

> familiar with the Sage format. . . . The video screencasts were very useful. . . . I really did like them. I made sure to watch all of them.
>
> **Felix:** For me, it was a bit too easy. I'm fairly comfortable with software, so the commands were fairly straightforward. But for the class, I think, for some people who weren't as comfortable with software, or maybe even [computer] algebra software, maybe that [Assignment 1] was worthwhile for them, just to get their feet wet.

While the need for extra support at the beginning of the optimization course obviously varied among students with different backgrounds in both computer science and mathematics, the archived Sage screencast video tutorials and digital lecture notes and worksheets appeared to be useful resources for all students.

Student-Generated, Revisited Optimization Problems

Beginning in 2013, Andrey had introduced the idea of the student-generated optimization problem for Assignment 2 (see Appendix A) that would then be revisited by students in subsequent assignments throughout the course with new methods being applied to the same problem (see Appendix B example). The origin of, and unintended consequences proceeding from, the introduction of this particular learning strategy were both equally fascinating. Ironically, student plagiarism was the actual original reason for this move to student-generated problems so that they would all be different and thus require independent and unique solutions. In discussing this strategy, Charles noted some original doubt about Andrey's idea: "I was amazed that he was able to make that work because I thought that students would make terrible choices with their problems, and somehow they'd be stuck with them and it wouldn't work well." Andrey himself had initial reservations about these assignments, particularly with regard to the grader's role:

> What I think, or hope at least, is good for students is that they can work on problems that they come up with on their own. . . . They have a chance to work on something that is of interest to them, in creating an interesting problem. So, I think that usually the challenge with such problems is that they are very difficult to grade because the grader would not be able to just pick up speed in order to grade 10 problems. . . . However, because there is no need for the grader to check any mathematics or computations . . . then it's actually much easier to grade it.

The student grader, Akemi, indicated the apart from requiring additional time to read them, the problems were fun to assess:

Akemi: Students made up a problems themselves. . . . The only challenge is that you really need time to read them. It really takes me a lot of time to review those assignments, because every problem is different. . . . If they set up the correct variables, if they make the right constraints—it really takes a lot of time. . . . Of course it's beneficial for students. They make up the problems, so that means that they really need to understand everything.

The six students that were interviewed all shared very positive memories regarding their personalized, self-generated optimization problem that they had created and shared with their peers for Assignment 2 in the course.

Brittany: I like the second assignment. It was fun. It was kind of time-consuming but that's because I wasn't as used to Sage yet. . . . The third assignment was using the Simplex Method. We used the function that we had already made from the second assignment, and how to use dictionaries. . . . Duality was the fourth assignment. . . . You use the same problem again. Then he does the same thing in class where he has his example of corn and barley and he continues to bring it through all the different methods. . . . I liked using the same problems throughout the course. . . . You really get to know that problem and everything about it, and how to use all the methods with it.

Cheng: So, for my problem we're planting trees—apple trees, pear trees, and palm trees—and making profits, and we use fertilizers and water. It took me around an hour. What I did, I just wrote down the equations first, and based on the equations, I formed my work. . . . I had a friend who came up with a problem to sell candies. But this guy put $10 000 into making the candy, and he only made $200 profit, so it didn't make any sense. So, making them up is kind of hard. You have to relate it to a real-life problem, and then you will have something that works.

Dawn: Our professor actually gave us a little bit of a warning and said, "Make sure your problem is interesting because you're going to be working with this for the rest of the semester." . . . I really loved formulating my own problem. . . . It was about refrigerated coffee beverages. So, one is going to be decaf or light caffeine, and then one was going to be heavy caffeine. It was pretty generic. My different constraints involved the amounts of caffeine to milk ratio that you have to do, and how the sugar should be equal in both

amounts. I had so much fun making it up. I remember at the end of it, I ran downstairs and showed my dad. He was like, “Okay, great—go away!” It was a fun assignment—I think it was my favourite.
Ezra: I liked it that way because by doing that [reading other problems] you can have a better understanding of the theme, rather than just having a single problem. . . . My project was about—I like food, so I imagined being an owner of a barbecue house and tried to—it was kind of like improvising, but I tried to put as much as I could imagine into this problem—it was funny for me—different raw materials, and selling strategies, and pre-orders.

Felix: I basically just took the framework of what he wanted, like just a linear problem. . . . It was about Jurassic Park II, and you had to pick which dinosaurs you wanted. Well, it was really simplistic. I had different types of dinosaurs and I said, well, the guys in Finance figured out which dinosaurs make how much money . . . from the increase in attendance from this. So, like, if you have a T-Rex, you get this much money per year. . . . Then you have space and land requirements and water—so, you have a limited amount of space, so you can’t just have any number of dinosaurs in the park. And then, I imposed various constraints, so you have to have certain amount of land type for the walking ones, and then you have the water ones, and then the flying ones. . . . The thing I like about that is that you build familiarity with the numbers and the solutions, so every time you take the right steps, you reach something that’s recognizable to you, instead of doing all new problems with new numbers each time.

Guang: It wasn’t very hard to come up with my own problem because what I did was I made a standard form problem, and then I put a word problem into it. . . . Have you seen the movie *“Harold and Kumar go to White Castle?”* It’s a comedy movie. Basically, these three guys drive into Mexico. They’re given twelve hours total and each of them only drives four or two hours, and they drive different speeds on the roads. I have a country road, a highway, and off-roading. It’s to see how far they can get. That was my word problem. . . . For every assignment there are new methods you’re using, so you already know what’s going on there, and it helps you understand.

We specifically noticed a few things during these particular responses: how students became unusually animated while describing their own optimization problem, using words such as "fun," "funny," or "favourite"; how Cheng, Felix, and Guang could not stop smiling or laughing, for example, when they were telling us about fruit orchards, Jurassic Park dinosaurs, and Harold and Kumar's Mexican driving adventures, respectively; and how all of these students indicated that they both appreciated the chance to share their problems with other students and to receive feedback on their own work. In trying to prevent plagiarism and to avoid the use of monotonous and simple problems, Andrey had stumbled upon a learning and peer assessment strategy that seemed to not only greatly increase student engagement, but also may have deepened conceptual knowledge insofar as it served to reinforce concepts in meaningful and authentic ways throughout the course.

Mid-terms and Final Examinations

Andrey's desire to eventually incorporate Sage software use into tests, mid-terms, and even final examinations clearly took him into new and unfamiliar territory in terms of what might be considered normal or commonplace for a third year, undergraduate mathematics course at the University of Alberta. In fact, these types of efforts—the significant use of CAS-based technology in formal mathematics assessment at the university level—are relatively rare within the related literature. Andrey explained his own anxiety surrounding the first use of Sage in a mid-term examination (see Appendix C), as well as the access glitch that occurred that day and how he managed to cope with the situation:

> The beginning of the mid-term exam was terrifying because I could not start the test because the servers could not be accessed. It took me probably five minutes to fix it. . . . I was mostly worried about how stressed those students might be. In principle, I had plans that if something did not go well, then they would work on paper. The mid-term is during their regular class time, so it's 70 minutes long. They had three questions. One question was just completely on paper, a conversion of a word problem into formulas. . . . Another problem was about the graphical interpretation and solution of the Simplex Method. My hope there was that they would use Sage to play with different values, and see how to construct those examples, but definitely, with the techniques involved there, they would be able to do it by hand. A third question required them to solve a relatively big problem using the Simplex Method. There

> were some questions about this third problem on the paper, but the actual problem was situated on the computer, and so they *had* to solve it using the computer. . . . We had had migration of IT services and those things did not migrate over completely, so I was just not made aware of this. Certain ports were locked, and so when I tried to look at it earlier, it was fine, but the test was in a different building, so it didn't work there that day. . . . I gave them an extra five minutes at the end of the test, since we had had this discombobulation with the server at the beginning.

Students also shared with us their own reflections on the mid-term examination that had recently taken place and which involved both the optional and required use of Sage for certain questions within a computer lab environment.

> **Dawn:** I think these three questions really reflected our last three assignments. . . . That's what I was really surprised about because it made me feel a little bit better as I was going through the mid-term because I was familiar with the format. . . . I think it's really fair. . . . I don't think it's fair to get a student kind of use to that format, where you can just do the assignments online on your computer, but then when it comes to your mid-term, you have to switch that mentality and go straight back to pen-and-paper. I don't think that really tests what you've learned. So, I think if professors want to stick to programming and using computers, then a mid-term and final should follow that format.

> **Brittany:** The computer component is important because no one, like engineers and analysts, does things by hand. Especially with 50 variables—how are you going to do that by hand? But you should still know the theory behind it . . . but the [software] aid is helpful for real-life applications.

> **Ezra:** I think it's necessary. If you depend on Sage heavily throughout the whole course, and you are not allowed to use it on the exam, that wouldn't be fair. . . . I think it's a natural outcome.

> **Felix:** If you're going to do the Simplex Method by hand, you're probably going to do it wrong. . . . Yeah, look, I don't see any other way you could test this, other than not using technology and making it really simple . . . but the exam would be too easy. And if you just, straight up, give someone a Simplex Problem, or a couple

> of problems, and then just tell them to do them by hand, it would just take way too long, and people would make too many small mistakes. . . . So, yeah, I think it was fair. . . . the questions test your understanding with and without the technology.

These interviewed students seem to have felt that the Sage-based mid-term examination was "fair" and appropriate in terms of mirroring the assignment work, involving real-world problems that required powerful software for solving, and balancing hand-written response sections with other questions that clearly did require the software. Notwithstanding, it was interesting to listen to other students, such as Cheng and Guang cited here below, where, despite also approving of the mid-term format in general, they use the phrases "not going to assess my knowledge," and "You don't actually engage with the material." Clearly, in their minds, the hand-written solutions are somehow more strongly correlated with "legitimate" mathematical knowledge, understanding, and engagement; whereas the use of the technology is somehow viewed as perhaps a type of perceived short-cut, or at least a lower form of understanding or engagement with the mathematics.

> **Cheng:** Basically, if you know Sage, you can do the first and second questions just by trying it in Sage. . . . If you did it by hand, then this question would take at least 10 minutes, but with Sage I just put in the code . . . Personally, 70% of the test is not going to assess my knowledge. I don't need to know anything about the course. I can just use Sage to do it. . . . Well, to be honest, I think the mid-term was good. What we had learned, at least we had a test on that, right. But I believe that it didn't test all of my knowledge about optimization problems.
>
> **Guang:** You don't actually engage with the material. . . . The first three questions can be solved with a graph because they have so few variables. . . . You just type in some kind of solution and see if the 2-D plot is what you're looking for—if not, you can try some other numbers. . . . Technology has its place, but maybe not the whole exam.

If Computer Algebra System (CAS) technology use in formal examinations (as opposed to just using scientific calculators) was relatively new to these students, it is little wonder that they were not easily able to connect the use of these powerful technologies to their impressions of "real" or "authentic" learning and understanding. Surely this becomes a highly significant point for both instructors and their students when discussing technology use in formal assessment, for it represents an aspect of one's belief system that allows, or disallows, such methods to be validated and implemented.

At the time of the site visit at the University of Alberta, the final examination had not yet been written but was only a few weeks away. Andrey discussed in detail his plans for the final exam, which would again involve Sage use.

> I'm trying to make it a 2-hour exam, but allow them three hours to write it, to give them some extra time so that they are not stressed. It will have problems that involve Sage, and so it will benefit from using computers. I will also try harder to make students show that they understand what the computer is actually doing. . . . And I really want students to demonstrate that they know how to set up the starting point for this situation, and then explain what steps they can do on it. If they want to, they can use computers to solve the problem and get the answer, but the task is not just to get the final answer, but to show how you can get to that certain point. . . . They will have seen the exam problems before, because I have taken them from the lecture notes, which they do have access to on eClass during the course. . . . I have not heard about other examples of using technology to this extent on exams in big courses. People have used it with maybe 10 or 20 students, but not for 60 or 90.

For MATH 373, Andrey was teaching the only section of this course, and thus had more flexibility in terms of what he was able to do with the final examination. In contrast, he noted that courses that involve many sections, and often a designated faculty "Course Captain" who oversees the organization of multiple sections, are usually much more prescriptive.

> In the multi-section courses, we have a very concrete syllabus, which shows which sections we need to cover, in what order, and that's it. Sometimes Course Captains will ask that we emphasize this particular part, or maybe also include some other topics. So, we don't really have much freedom. . . . Some classes have consolidated exams where they all write it together. Usually each instructor is asked to contribute one or two problems on the particular topic, and then that instructor usually grades those problems as well. . . . If the exam is consolidated, then it is one person grading all of the exams for the same problems. Some courses are not consolidated because they are so big. For example, Calculus has over 2000 students, so there are 15 sections. . . . There are multiple exams that have to all be different, and the exams that are run at the same time are still consolidated. . . . For multi-section courses I did not try to use computers in any way, and I frankly just don't see how it could be possible, for final exams, or for any other shared assessment with more than one section.

After four iterations of the MATH 373 final examination being written in a computer lab, and involving the use of Sage software for at least some of the questions, Andrey had obviously become convinced that this was not only a doable strategy but one that more accurately assessed his students' ability to understand and solve problems using technology. It is of course possible that some, or perhaps even many of the students that did not volunteer to be interviewed during the site visit were of the opinion that Sage technology was not considered a positive addition to both the course work and assessment tools. However, based on the comments made by the seven students that we did speak with, we can say with some certainty that they not only found the Sage-enhanced assessments to be a "fair" situation, but also one that represented an important and timely approach to this particular topic in light of the mathematical content and available technology.

DISCUSSION

In this section, we will discuss some of the broader issues around technology in mathematics teaching and assessment; how these novel uses of Sage software were shared with other faculty; and some suggestions for related future research.

Technology in Mathematics Teaching, Learning, and Assessment

Clearly there is still no international consensus around the role and effectiveness of instructional technology in the learning of mathematics, particularly with regard to powerful CAS-based tools such as Sage. In studying nearly a decade's worth of mathematics journals, Smith Risser (2011) concluded that there are a few key reasons why healthy skepticism abounds.

> More than twenty years after the introduction of the first handheld graphing calculator the mathematics community appears to still be struggling with the use of technology in the teaching and learning of mathematics. . . . The arguments against technology use centre on three main issues: whether technology should change the focus of mathematics curriculum, whether technology use changes how students conceptualize mathematics, and whether the benefits of technology outweigh the costs. These arguments provide a revealing look at what some mathematicians fear are the negative effects of technology use on the learning of mathematics. (Smith Risser, 2011, p. 97)

One of the most interesting aspects regarding the role of technology in mathematics learning is whether or not, and if so in what specific ways, the use of the technology directly impacts the understanding of existing mathematics curricula content. For some mathematicians, they appear to equate "true understanding" exclusively with the learning of traditional algorithms and hand-written solutions and proofs. To them, technology may be a helpful way to reinforce this understanding through answer checking, demonstration, or even exploration, but it is perceived as supplementary. For example, note how Zack, who had taught the Optimization course once previously, and who had included Maple software explorations in his lectures and course assignments, praises the use of powerful digital tools, yet distinguishes these from real "understanding":

> **Zack:** What's nice about using technology is that you can ask them some interesting and very difficult computational problems that you couldn't do in a reasonable amount of time with paper-and-pencil. . . . Students should learn about the technology that's available. . . . that's why this technology is fantastic. Why shut it away? They should learn how to use it, but it shouldn't be used as a replacement for understanding. . . . I think that most professors encourage students to use the technology as a checking tool and as a confidence builder, so that what you're handing in is actually correct, but it shouldn't be used to replace your understanding of the material.

In contrast, other mathematicians and educational researchers are of the opinion that the use of CAS technology not only is intrinsically tied to the understanding of existing mathematics curricula content, but that it in fact can open up whole new ways of thinking about content that were previously inaccessible, hence actually serving to expand the very possibilities of mathematical knowledge and understanding. Verillon and Rabardel's theory of *Instrumental Genesis* (Trouche, 2004) speaks to this idea of how the development of ICT, together with its usage, has led to both "instrumentation," meaning a person is able to use the instrument, and also "instrumentalisation," meaning that the tool actually shapes the actions and the character of the knowledge constructed with the tool (Haapasalo, 2013, p. 87).

Beyond just performing quicker calculations, Tobin and Weiss (2016) likewise concluded that handheld CAS-enabled calculators actually allow for expanded ways of exploring and thinking about mathematical phenomena:

> The best way to use this technology would be to write a curriculum around [the] expectation that technology like it will always be available in [the] future and that the focus on learning should shift

> to contexts, to applications, to learning of concepts that transfer well. It seems that we are far from being able to do this at present. . . . The different outputs CAS calculators can offer also lead to interesting discussions on alternative forms of solutions (e.g., in anti-differentiation) or different methods of solution (e.g., in differential equations). These surely promote thinking about the solution, not merely obtaining it! (p. 40)

Xavier, a friend of Andrey's and an instructor at another university in the city who had been experimenting with Andrey's applets in his own math teaching, noted the following example of how Sage allowed for increased understanding:

> **Xavier:** So, take the vector field tool. You can write down the algebra for a vector field. It's very difficult to get an intuitive grasp of what that algebra means. The visualization tool lets you see the vector fields—sort of what directions. If there's sort of straight motion, curving motion, spiraling motion—and that way, when they start to do calculations, if they're integrating a line integral through that vector field, they can actually see whether or not they expect it to be positive or negative, whether they expect it to be large or small, they can see whether or not you're moving with, or tangential to that field. So, I think, to get intuition for the problems before they do them, and then to have a bit of a check for intuitively using this—doing what it should be doing, based on the visualization.

The vector field tool (see Figure 2) represented an example of an applet that underwent revisions based on peer feedback.

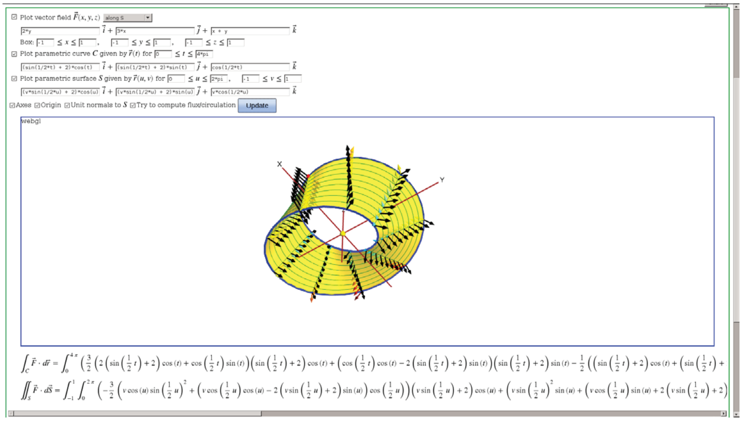

Figure 2. Vector field applet created by Novoseltsev using Sage software.

Dr. Novoseltsev had made significant revisions to the *Mathematical Programming and Optimization* (MATH 373) course in terms of how Sage software had been used in both his teaching style and assessment practices. Charles, the TLEF Project leader, reflected on the gradual and purposeful changes that colleague Andrey had made to the MATH 373 course:

> This is a very unique situation because Optimization is a course which has always suffered by the fact that you cannot do real world style examples by hand, you just can't. By its very nature, it involves systems of integral equations that cannot be solved in your head, and you can illustrate on the board with so few variables that it seems meaningless. On the other hand, what Andrey realized right away was that by using Sage, and also by integrating both the algebraic manipulation capacity of Sage, and the visual capacity, the actual ability to show the graphs, to show the boundary lines for the various Optimization constraints, that you could really take it to the next level. You could have the students see how it worked in a sort of live demonstration with real variables, real numbers in class. So, that's what I think he did at first, and then he realized that the students wanted to actually play with this, and use it for their homework, and so then he created something that they could do for their homework. And then he began to realize there was a disconnect between the course that the students experienced in the classroom and on homework, and their exams. They went from having this marvelous tool that they knew how to use to having to sit there in an exam room and do everything by hand, and it just didn't make any sense. So, then he took the great leap of just trying to make the entire course Sage-based, and now it is. . . . He actually has them in a computer lab, the software is all loaded on the machines, they know how to use it, and it's all done that way.

Andrey had unmistakably transformed the Optimization course over time and through multiple iterations of the course, in terms of pedagogical and assessment strategies involving the purposeful use of Sage software. You can see this first affecting the way applets were used for illustrating and co-developing (with student input) key concepts in the classroom setting; then in the impact on the way the students worked with interactive applets for homework and course assignments; and, finally, through the inclusion of required Sage technology use within formal course tests and exams. One could easily argue that he had thus demonstrated all four levels of Puentedura's (2006) Substitution, Augmentation, Modification, and Redefinition (SAMR) model. Based on our relatively small participant sample, Andrey's supervisor and his students were appreciative of his math knowledge, classroom methods, supportive attitude, and novel assessment practices. To the external observer, he had also thus demonstrated key characteristics of Koehler and Mishra's (2009) Technology, Pedagogy, and Content Knowledge (TPACK) model insofar as his reflective teaching practice provided ample evidence of these qualities.

Whether or not this novel approach could be shared with other faculty members in such a way as to promote similar practices in MATH 373, or in other mathematics courses, represented a completely different issue. It is to these challenging areas of faculty sharing and departmental, or system level reform that we now turn.

Issues in Departmental Sharing

Notwithstanding its non-traditional technology use and instructional methods, *Mathematical Programming and Optimization* (MATH 373), offered within the Mathematical and Statistical Sciences Department at the University of Alberta, was also a popular course because it fulfilled requirements for other departments across the university. According to Charles, those other departments are, of course, more applied then we are, and I can't imagine they will have any complaints that their students actually know how to implement optimization algorithms. I mean, you can give them a problem and they will go to Sage and they know how to set it up, how to run it, as opposed to just doing it all by hand.

Dr. Doran also explained the inherent difficulty in promoting departmental changes regarding instructional practices, in light of academic freedom, as well as the realities of multi-section courses and how they are traditionally operationalized.

> I can tell you right now that the objectives involving dissemination beyond the select courses with a few professors has not been successful, and it's not for want of trying, but there's no departmental mandate that says everyone must make an attempt to do this, and there's even a struggle to convince Course Captains to mandate it in an individual multi-section course. . . . I think a lesson I've learned is if you really want to have a project like this have the impact it should, then you need to have some kind of departmental mandate on top of just a willingness, or an openness to try it. It's chicken or egg, right? You have to have the successful apps—you have to have it working first to convince people that

> it should become the new model, and that they should take the time to learn it, or to adapt to it. In that way, it's like any other innovation in education.

Similarly, Andrey described his desire that the growing number of Sage applets that he had been creating would eventually be used by other mathematics colleagues in their own respective teaching and assessment practices:

> Well, the hope was to use Sage applets in other courses, and to encourage others to come up with ideas. So, they can either go straight to the applets and write them themselves, or I also could try to help them with whatever they requested. For some courses, instructors are interested in doing something with computers, but there are already so many simple tools out there which they can use. . . . But whenever you need some non-trivial mathematics such as symbolic integration, for example, which you can do in a specialized way on CAS, then you really need something bigger and more powerful, and something bigger usually cannot work in a browser.

Xavier, the former University of Alberta student now holding a faculty position across town, had adopted Andrey's Sage applets in his own mathematics teaching and was regularly in contact with him. Note how he describes Andrey informally attempting to reach out to him by way of encouraging him to try out these applets and teaching methods.

> **Xavier:** What I was going to do was I was going to incorporate them into my lectures, and I was going to encourage, but not require students to use them for assignments. . . . I use to draw a lot of stuff on the chalkboard and my drawing is not great, so I was quite excited. . . . Sage is fantastic because it's free and open source. . . . I'm responsible for Calculus 1 through 4, as well as the Differential Equations course, so I was quite interested in what he was doing. It was a great meeting. We went for coffee. He told me what the project was about . . . gave me the links, gave me the permissions, and said, "Use it for your classes and get back to me. Let me know how it works."

Xavier further explained how that because he was in a smaller institution, he was responsible for most math courses and thus enjoyed certain freedoms such as full control over assignment and exam content and assessment, unlike colleagues in larger institutions such as UA where Course Captains ultimately were given responsibility for many of these decisions. His particular story is an interesting case because we feel he clearly represents a mathematics professor "in transition" in terms of his beliefs regarding instructional technology tools. In what follows, note how he celebrates the software, yet admits to still not being ready to change formal assessment practices, not even allowing standard calculators during in-class exams:

> **Xavier:** I have a bias, because of my particular background in mathematics, for elegant problems that can be worked out without complicated numbers, but that's because I'm a pure mathematician and not an applied mathematician. . . . So, it encourages me to throw in more applied questions with real numbers, where you actually need CAS to do it. . . . You asked, did I change my assignments, and the answer is, not really. . . . I made use of the visualization at different points, so in my lecture organization there was quite a bit of change. . . . I'm fine with students using technology. I encourage them. . . . For my in-class exams there are no electronics—no calculators. I design the exam such that there's no super difficult multiplication or division. . . . I'm somewhere along that learning curve.

According to Andrey, interested colleagues such as Xavier could quite easily and quickly be mentored in how to use Sage software for the types of teaching, learning, and assessment practices that he had experimented with in his courses: "It's just how to use the basics of Sage—how to publish worksheets, how to insert comments, how to efficiently convert and access files for grading. . . . I think that if someone was sold on the usefulness of this, one day would be enough to get them up to speed."

Like with any entertaining murder mystery, the adoption of powerful, CAS-based software tools such as Sage in undergraduate mathematics teaching involves at least the following three key factors that are ultimately established by the watchful detective: motive, means, and opportunity. Mathematics instructors, through observation, reading, or word of mouth, must come to *believe* that these tools are actually useful and beneficial to student learning; they must have *access to software* that is both affordable for them and their students, and which is easily learned and implemented; and they must be supported by their colleagues, Course Captains, and Department Chairs insofar as they are *permitted and encouraged to experiment with these strategies* within their teaching and assessment practices (Bray & Tangley, 2017; Ertmer et al., 2012).

In our previous research on systemic change, we have found that in two particular cases (UK, Canada) where technology had become a significant part of a university mathematics department's *modus operandi* and shared teaching philosophy, this lengthy transformational process required the presence of a number of key elements which included: a dedicated core group led by a committed advocate in a position of influence/power;

a strong and shared incentive for change; strategic hiring processes; an administration which supports creative pedagogical reform and well-considered risk-taking; and, a continuous and determined revisiting of the original vision and purpose (Jarvis, Lavicza, & Buteau, 2014, p. 117).

Oates (2011) contends that, "The effective integration of technology into the teaching and learning of mathematics remains one of the critical challenges facing contemporary tertiary mathematics" (p. 709). He reports on the technology implementation occurring at the University of Auckland, proposing a detailed taxonomy for describing and comparing technology use within individual courses and departments that identifies a complex range of factors, summarized under six defining characteristics (i.e., access, assessment, organizational factors, mathematical factors, staff factors, and student factors) of an "integrated technology mathematics curriculum (ITMC)". The survey on which his taxonomy was based drew upon the input of 56 colleagues from international tertiary institutions involved in the teaching of undergraduate mathematics. In conclusion, he highlights the urgent need to revisit curricular content and assessment practices:

> With respect to assessment, both pedagogical consistency, and the impact of CAS on examination questions, are seen as particularly significant issues. . . . For content, the findings reported here support the complexity of assessing the values of topics, and support the overall conclusion that a re-examination of the changing pragmatic and epistemic values of specific topics, and the goals of mathematics education, within a rapidly evolving technological environment, remains a pressing challenge for undergraduate mathematics educators. (Oates, 2011, p. 720)

The ability to change curricular content, teaching practices, and assessment strategies, as a negotiated part of the reform process clearly requires a sustained and long-term commitment by faculty within any mathematics department. Through the targeted application of project funding in these two math courses in which Sage was heavily adopted, Charles and Andrey had together developed resources and processes at the University of Alberta that clearly showed promise in terms of student learning and engagement. Despite certain inherent obstacles and limitations, the sharing of these ideas with peers, both within and beyond their own institution, was already beginning to bear fruit as colleagues began to show an interest.

> Although many teachers are still struggling to achieve meaningful technology integration in their classrooms, . . . recent changes in access, student characteristics, and curricular emphases may provide some much needed impetus in moving teachers' efforts forward. Our hope is that these changes, together with

> modifications to professional development and district technology plans, will coalesce into a perfect "technology integration" storm that continues to empower more and more teachers to use technology in ways that prepare our students for the future they will inherit. (Ertmer et al., 2012, p. 434)

Dr. Novoseltsev had clearly taken bold new steps regarding the implementation of technology within his curriculum and assessment planning at the post-secondary level. While such pedagogical experiments were fraught with both first- and second-order barriers, as described above, Andrey had indeed provided an informative example of perseverance and progress. More specifically, the adoption of personalized, revisited, and highly engaging optimization problems, along with carefully developed summative assessment tools (midterms/finals) that required CAS-based technology use in ways parallel to those with which his students were familiar, together demonstrate a level of sophistication and accomplishment well worth sharing.

Future Recommendations

Based on this study, future recommendations would include the development of a centralized website via which mathematics instructors such as Dr. Novoseltsev could continue to post and share freely-available, open source Sage applets; technology enhanced assignment and assessment samples; and perhaps even print or video-documented classroom success stories.

Further national surveys of Canadian mathematicians (see, for example, Jarvis, Buteau, & Lavicza, 2014) regarding their beliefs and practices would be helpful; as would further case studies in other Canadian mathematics departments where instructional technology has been widely adopted at the university level (Jarvis, Lavicza, & Buteau, 2014). More specifically, a research study focusing on whether or not, and how these powerful, CAS-based tools actually affect student content learning in mathematics would no doubt be of great interest to university faculty, department chairs, and administrators.

Acknowledgements

We gratefully acknowledge that this research was made possible through internal research funding support offered by the Offices of Research Services at Nipissing University, Brock University, and the University of Alberta.

References

Artigue, M. (2002). Learning mathematics in a CAS environment: The genesis of a reflection about instrumentation and the dialectics between technical and conceptual work. *International Journal of Computers for Mathematical Learning, 7*(3), 245-274.

Artigue, M. (2009). The future of teaching and learning mathematics with digital technologies. In L. H. Son, N. Sinclair, J. B. Lagrange, & C. Hoyles (Eds.), Mathematics education and technology: Rethinking the terrain—The 17th ICMI Study (pp. 463-475). New York, NY: Springer.

Beaudin, M., Picard G. (2010). Using symbolic TI calculators in engineering mathematics: Sample tasks and reflections on a decade of practice. *International Journal for Technology in Mathematics Education, 17*(3), 69-74.

Buteau, C., & Muller, E. (2014). Teaching roles in a technology intensive core undergraduate mathematics course. In A. Clark-Wilson, O. Robutti, & N. Sinclair (Eds.), *The mathematics teacher in the digital era: An international perspective on technology focused professional development.* Berlin: Springer.

Buteau, C., Jarvis, D. H., & Lavicza, Z. (2014). On the integration of Computer Algebra Systems (CAS) by Canadian mathematicians: Results of a national survey. *Canadian Journal of Science, Mathematics and Technology Education, 14*(1), 35-57.

Bossé, M. J., & Nandakumar, N. R. (2004). Computer algebra systems, pedagogy, and epistemology. *Mathematics and Computer Education, 38*(3), 298-306.

Bray A., Tangney B., (2017, November). Technology usage in mathematics education research: A systematic review of recent trends. *Computers & Education, 114*, 255-273.

Bruce, C. (n.d.). *The IWB Framework.* Trent Math Education Research Collaborative. Retrieved from http://tmerc.ca/iwb-framework/

Connors, M. A., & Snook, K. G. (2001). A technology tale: Integrating hand held CAS into a mathematics curriculum. *Teaching Mathematics and its Applications, 20*(4), 171-190.

Creswell, J. (2013). *Qualitative inquiry and research design: Choosing among five approaches* (3rd ed.). Thousand Oaks, CA: Sage.

Decker, R. (2011). The Mathlet Toolkit: Creating dynamic applets for differential equations and dynamical systems. *International Journal for Technology in Mathematics Education, 18*(4), 189-194.

Denzin, N. K., & Lincoln, Y. S. (Eds.). (2005). *The Sage handbook of qualitative research* (3rd ed.). Thousand Oaks, CA: Sage.

Ertmer, P. A., Ottenbreit-Leftwich, A. T., Sadik, O., Sendurur, E., & Sendurer, P. (2012). Teacher beliefs and technology integration practices: A critical relationship. *Computers & Education, 59*(2), 423-435.

Fey, J. T., Cuoco, A., Kieran, C., McMullin, L., & Zbiek, R. M. (Eds.). (2003). *Computer algebra systems in secondary school mathematics education*. Reston, VA: National Council of Teachers of Mathematics.

Haapasalo, L. (2013). Adapting assessment to instrumental genesis. *International Journal for Technology in Mathematics Education, 20*(3), 87-93.

Hamilton, E. R., Rosenberg, J. M., & Akcaoglu, M. (2016). The Substitution Augmentation Modification Redefinition (SAMR) model: A critical review and suggestions for its use. *TechTrends, 60*(5), 433-441.

Heidenberg, A., & Huber, M. (2006). Assessing the use of technology and using technology to assess. In L. A. Steen (Ed.), *Supporting assessment in undergraduate mathematics* (pp. 103-108). Washington, DC: The Mathematical Association of America.

Jarvis, D. H., Lavicza, Z., & Buteau, C. (2012, July). Computer Algebra System (CAS) uSage and sustainability in university mathematics instruction: Findings from an international study. *In proceedings of the 12th International Congress on Mathematical Education (ICME-12).* Seoul, Korea.

Jarvis, D. H., Lavicza, Z., & Buteau, C. (2014). Systemic shifts in instructional technology: Findings of a comparative case study of two university mathematics departments. *International Journal for Technology in Mathematics Education, 21*(4), 117-142.

Kendal, M., & Stacey, K. (2002). Teachers in transition: Moving towards CAS-supported classrooms. *Zentralblatt für Didaktik der Mathematik (ZDM), 34*(5), 196-203.

Kieran, C., & Drijvers, P. (2006). The co-emergence of machine techniques, paper-and-pencil techniques, and theoretical reflection: A study of CAS use in secondary school algebra. *International Journal of Computers for Mathematical Learning, 11*(2), 205-263.

Koehler, M., & Mishra, P. (2009). What is Technological Pedagogical Content Knowledge (TPACK)? *Contemporary Issues in Technology and Teacher Education, 9*(1), 60-70.

Lavicza, Z. (2006). The examination of Computer Algebra Systems (CAS) integration into university-level mathematics teaching. In C. Hoyles, J.B. Lagrange, L. H. Son, and N. Sinclair (Eds.), *Online proceedings for the 17th ICMI Study Conference, Hanoi University of Technology* (pp. 37-44). Hanoi, Vietnam.

Lavicza, Z. (2008a). The examination of Computer Algebra Systems (CAS) integration into university-level mathematics teaching. *Unpublished Ph.D. dissertation,* The University of Cambridge, Cambridge, UK.

Lavicza, Z. (2008b). Factors influencing the integration of Computer Algebra Systems into university-level mathematics education. *International Journal for Technology in Mathematics Education, 14*(3), 121-129.

Martinovic, D., Muller, E., & Buteau, C. (2013). Intelligent partnership with technology: Moving from a mathematics school curriculum to an undergraduate program. *Computers in the Schools, 30*(1-2), 76-101.

Meagher, M. (2012). Students' relationship to technology and conceptions of mathematics while learning in a Computer Algebra System environment. *International Journal for Technology in Mathematics Education, 19*(1), 3-16.

Oates, G. (2011). Sustaining integrated technology in undergraduate mathematics. *Science and Technology, 42*(6), 709-721. doi:10.1080/0020739X.2011.575238

Puentedura, R. (2006). Transformation, technology, and education [Blog post]. Retrieved from http://www.hippasus.com/resources/tte/

Puentedura, R. (2014). Building transformation: An introduction to the SAMR model [Blog post]. Retrieved from http://www.hippasus.com/rrpweblog/archives/2014/08/22/BuildingTransformation_AnIntroductionToSAMR.pdf

Pierce, R. L., & Stacey, K. (2004). A framework for monitoring progress and planning teaching towards the effective use of Computer Algebra Systems. *International Journal of Computers for Mathematical Learning, 9*(1), 59-93.

Rosenzweig, M. (2007). Projects using a Computer Algebra System in first-year undergraduate mathematics. *International Journal for Technology in Mathematics Education, 14*(3), 147-149.

Sarvari, C. (2005). Experiences in the integration of CAS into engineering education at the University of Pecs. *Proceedings of the 4th CAME Symposium*. Roanoke, VA. Retrieved from http://www.lkl.ac.uk/research/came/events/CAME4/CAME4-topic2-ShortPaper-Sarvari.pdf

Schurrer, A., & Mitchell D. (1994). Technology and the mature department. *Electronic Proceedings of the 7th ICTCM*. Retrieved from http://archives.math.utk.edu/ICTCM/VOL07/C002/paper.pdf

Sevimli, E. (2016). Do calculus students demand technology integration into learning environment? Case of instructional differences. *International Journal of Educational Technology in Higher Education, 13*(37), 1-18. doi:10.1186/s41239-016-0038-6

Shulman, L. (1986). Those who understand: Knowledge growth in teaching. *Educational Researcher, 15*(2), 4-14.

Smith Risser, H. (2011). What are we afraid of? Arguments against teaching mathematics with technology in the professional publications of organisations for US mathematicians. *International Journal for Technology in Mathematics Education, 18*(2), 97-101.

Somekh, B. (2008). Factors affecting teachers' pedagogical adoption of ICT. In J. Voogt & G. Knezek (Eds.), *International handbook of information technology in primary and secondary education* (pp. 449-460). Boston, MA: Springer.

Stein, W., & Joyner, D. (2005). Sage: System for Algebra and Geometry Experimentation. *ACM SIGSAM Bulletin, 39*(2), 61-64.

Stewart, S., Thomas, M. O. J., & Hannah, J. (2005). Towards student instrumentation of computer-based algebra systems in university courses. *International Journal of Mathematical Education in Science and Technology, 36*(7), 741-750.

Tall, D. (2013). The evolution of technology and the mathematics of change and variation. In J. Roschelle & S. Hegedus (Eds.), *The Simcalc vision and contributions: Democratizing access to important mathematics* (pp. 449-561). Boston, MA: Springer.

Tobin, P. C., & Weiss, V. (2016). Teaching undergraduate mathematics using CAS technology: Issues and prospects. *International Journal of Technology in Mathematics Education, 23*(1), 35-41. doi:10.1564/tme_v23.1.04

Thompson, P. W., Ashbrook, M., & Musgrave, S. (2015). *Calculus: Newton meets technology.* (eBook)

Thompson, P. W, Byerley, C., & Hatfield, N. (2013). A conceptual approach to calculus made possible by technology. *Computers in the Schools, 30*(1-2), 124-147.

Trouche, L. (2004). Managing the complexity of human/machine interaction in computerized learning environment: Guiding students' command process through instrumental orchestrations. *International Journal of Computers for Mathematical Learning, 9*(3), 281-307.

Weida, R. (1996). Computer laboratory implementation issues at a small liberal arts college. *Electronic Proceedings of the 9th ICTCM.* Retrieved from http://archives.math.utk.edu/ICTCM/VOL09/C051/paper.pdf

Yin, R. K. (2009). *Case study research: Design and methods* (4th ed.). Thousand Oaks, CA: Sage.

APPENDIX A: MATH 373 ASSIGNMENT 2 (SPRING 2015)

Chapter 2 Assignment: Problem Formulation

1. Word Problem: Compose a word problem. You should start by practicing on some other problems (e.g. from course notes) and you may use them for inspiration, but please formulate your own problem in your own words and with your own numbers - do not look at other problems while writing yours. Taking "Dog Food" problem and replacing "Dog" with "Cat" throughout the text does NOT count at your own problem! Refrain from looking at others' submissions until you have posted your own. Your word problem should "make sense" to people who have never heard of Math 373 and optimization and be written in proper English, ask someone to proof read it, especially if English is not your native language. Points will be taken off for typos/mistakes/unclear sentences! While it may seem harsh, this is the problem that you will be working on for the rest of the term and that means your group members will have to deal with it as well—be kind to them and write accurately.

2. Linear Programming Problem: Formulate an LP problem corresponding to your word problem. Make sure to explicitly describe all involved decision variables: what do they represent and in what units are they measured. Explain the physical meaning of each constraint and how do you derive it (e.g., $3C + B \leq 1500$ means that the total amount of fertilizer used cannot exceed the available amount).

Once you are done, input your problem into Sage.

```
A = ([2, -1], [1, -1], [1, 0],)
b = (-1, 2, 1)
c = (1, 5)
P = InteractiveLPProblem(A, b, c)
P
```

3. Standard Form: Convert your problem to standard form. You are free to use Sage to do it automatically in a single step, but please explain in words what has to be done (not "use this command" but rather "multiply the second inequality by -1, replace the third equation with two inequalitites, etc."). In standard form your LP problem must involve at least 5 decision variables and at least 4 constraints (not counting sign restrictions or constraints involving a single variable only). There is no upper limit: if you are so adventures that it gets difficult to enter all the coefficient or display output, talk to me and we'll try to figure out how to deal with such a problem. If your problem has too few variables/constraints, go back to the beginning and make it more interesting! Of course, if you change your word problem, you have to adjust its conversion to an LP problem accordingly. There is no need to keep the "old" problem around. In addition, it would be nice if the numbers of constraints and variables are different for your problem. (This will help you to avoid confusion in duality theory.)

These commands have to return "True":

```
P.n_variables() >= 5
```

```
[r.nonzero_positions() >= 2 for r in P.A().rows()].count(True)
>= 4
```

It is not strictly required, but it would be better if this command returns "True" as well:

```
P.n_variables() != P.n_constraints()
```

4. Feasible Set: Adjust your problem (both words and formulas!), if necessary, to make sure that the feasible set is non-empty (i.e. the problem is feasible) and has at least 4 vertices, i.e. the following command should give "True":

```
P.feasible_set().n_vertices() >= 4
```

5. Solution: Use Sage to find the optimal value and an optimal solution for your problem.

What do these numbers mean in terms of your original word problem?

APPENDIX B: MATH 373 ASSIGNMENT 3 (SPRING 2015)

Chapter 3 Assignment: Simplex Method

1. Word Problem: Start with the word problem you have composed last time: your submission should include the word problem, description of decision variables, and formulation as an LP problem. (No need to keep derivation of each constraint or explanation of conversion to standard form.) If the initial dictionary of your problem is feasible, tweak the problem (both the word and formula versions, so that they continue to match) a little to make the initial dictionary infeasible and force you to go through the auxiliary problem phase! You still should make sure that your problem has a feasible set with at least four vertices, the number of decision variables is at least five, and the number of constraints involving two or more variables is at least four. Make sure also that your problem is bounded, so that you do have the optimal value and at least one optimal solution!

2. Simplex Method—Feasible Problem: Use the Simplex Method to find ALL optimal solutions and ALL BASIC optimal solutions of your problem! (If your problem has a lot of basic optimal solutions, find at least 3 of them.) You may want to watch the "Recovering from Wrong Choices" screencast on eClass for how to "fix mistakes." It is OK to use decimal approximations if precise computations look too ugly, but you need to keep at least 5 digits for each number. i.e. you should use RealField(20) or higher. Beware of approximations issues, however! See "Approximation Issues" worksheet. You are NOT allowed to submit work invoking run_simplex_method() command and I do not recommend using it at all until you have solved the problem yourself.

If your solution uses less than 4 iterations of enter-leave-update steps (combined for auxiliary and original problem), go back to your word problem, make it more complicated, and adjust all other steps as necessary. No need to include the old "simple" version in your submission.

3. Simplex Method—Infeasible Problem: Take the word problem you have been working on and slightly modify its constraints/parameters in such a way that the problem becomes infeasible (e.g., for Corn and Barley the requirement to grow at least 2000 acres of corn would do the trick.). Provide below your modified word problem and its formulation as a LP problem.

You can quickly check if your modified problem is indeed infeasible via

```
P.is_feasible()
```

Apply the Simplex Method to this problem to prove that it is infeasible. (You cannot just invoke run_simplex_method().)

4. Degenerate Dictionaries: Have you encountered any degenerate dictionaries while working on this assignment? If yes, give a clear reference to it. If no, explain whether it is possible for your problem to have degenerate dictionaries.

APPENDIX C: MATH 373 MID-TERM EXAMINATION (SPRING 2015)

Instructions: Points WILL be taken off if you deviate from any of the following instructions:
1. Fill in the information above, including “Desk:” from a plaque along its top side.
2. Authenticate on the lab computer (you are not allowed to use your own device).
3. Start Mozilla Firefox web browser (not Chrome or Internet Explorer).
4. Go to [University of Alberta based url] (https:// is important!)
5. Press F11 to switch to full screen.
6. Log in to your account. You will see no worksheets—do not make any!
7. Once the test starts, make your own single copy of the published test worksheet.
8. You are not allowed to start or use any other program, access any other web site, create any other worksheets, or share/publish your test worksheet.
9. You are not allowed to use any run_... or possible_... commands.

This exam consists of 4 pages (including this title page) with 3 question(s). You can use any calculator without wireless capabilities. TURN OFF AND PUT AWAY ALL OTHER ELECTRONIC DEVICES. You can use one 2-sided sheet of notes in your own handwriting (do not submit it). You may not use any other notes or your own scratch paper. If you run out of space on the problem page, please use the back of the previous problem (which should be conveniently located on your right). Ask for more paper if it is still not enough. You must show your work on the exam paper with explanations in plain English. If a problem asks you to use a specific method, you MUST use this method. You may get zero credit for any other solution, even if it is correct. Each of the 3 questions are worth 10 marks, for a total of 30 possible marks. Good luck!

1. Consider the following LP problem with a “mystery” constraint:

$$\begin{aligned} \max \quad & x_1 + x_2 \\ & 3x_1 - x_2 \geq 3 \\ & -x_1 + x_2 \geq 1 \\ & Dx_1 + Ex_2 \leq F \\ & x_1, x_2 \geq 0 \end{aligned}$$

(a) Give an example of the last constraint for which the problem is feasible, but there are no optimal solutions or explain why it does not exist.
(b) Give an example of the last constraint for which there are no feasible solutions or explain why it does not exist.
(c) Give an example of the last constraint for which (3; 5) is an optimal solution or explain why it does not exist.

2. Your company produces 4 types of fertilizer: A, B, C, and D. To produce 1 kg of fertilizer A you need 300 g of potash (P), 400 g of phosphate (H), and 300 g of nitrogen (N). To produce 1 kg of fertilizer B you need 300 g of P, 300 g of H, and 400 g of N. To produce 1 kg of fertilizer C you need 500 g of P, 200 g of H, and 300 g of N. Finally, to produce 1 kg of fertilizer D you need 400 g of P, 400 g of H, and 200 g of N. Suppliers can provide 40 kg of P, 40 kg of H, and 30 kg of N per day. Net profit is \$20, \$40, \$50, and \$30 per kilogram of A, B, C, and D respectively. Formulate an LP problem for maximizing the profit of your company. Make sure to clearly describe all decision variables, their units, and the physical meaning of each constraint. No need to simplify constraints and/or objective. You can also get 2 bonus points (but no more than 100% for the whole exam) if you find the optimal value and all optimal solutions using Simplex Method!

3. Solve the LP problem provided in the Sage worksheet using the (\ Regular”) Simplex Method and, based on your solution, write down the following information. Entering and leaving variables on each step:

Step	1	2	3	4	5	6	7	8	9
Entering									
Leaving									

The optimal value (or explain why it does not exist):
An optimal solution (or explain why it does not exist):
All optimal solutions (or explain why there is only one or none):