Adding Spice to Undergraduate Mathematics with Sage

Proposal for a John Lantz Senior Fellowship for Research or Advanced Study
Rob Beezer, Department of Mathematics and Computer Science
DRAFT September 7, 2009 DRAFT

Sage is software for mathematics. That might not sound very impressive, but this is not software to just perform numerical approximations, but to also do exact computations, and to compute symbolically. Mathematics is infinite and continuous, while computers are finite and discrete. To create a program that can "do" higher mathematics is an incredible challenge. Sage's main competitor, *Mathematica*, has been in development for over twenty years and retails for $2,500. However Sage is free, and more importantly, it is open-source, so anybody can read the implementation of the algorithms employed, can correct bugs, and can extend the capabilities and features of the program. I propose to use a Lantz Fellowship to devote an academic year, in collaboration with members of the Sage community, to expanding and improving the usefullness of Sage in the undergraduate mathematics classroom. Specifically, this project will create Sage-enhanced textbooks for undergraduate mathematics courses.

A Swiss undergraduate contributor to my open-source linear algebra textbook made me aware of Sage in late 2007. I was *immediately* struck by the concept of an open-source program for mathematics, and very impressed by the other mature specialized open-source projects that had been brought together under the Sage umbrella. With the start of the Fall 2008 semester, and my duties as Faculty Liaison to the Thompson Hall remodel project mostly behind me, I resolved to abandon *Mathematica* to use (and learn) Sage at every opportunity. Soon I was reporting bugs in the Sage forums, and suggesting additions to the linear algebra code. I participate regularly in the Sage forums on Goggle Groups and especially the education forum. In the developer forum, I try to represent the use of the program by students or teachers, rather than just researchers. With great effort, I learned the Sage development process and started contributing code to Sage itself, and giving presentations to other developers at Sage Days workshops. At some point in this arduous and invigorating journey I became a "Sage developer," though nobody rang a bell at that moment.

The main activity of this project will be to create collections of Sage worksheets that are extensions of existing textbooks or new subject-specific supplements. Recent experience tells me that in a project of this nature and magnitude, opportunities to modify and enhance Sage itself will be inescapable (and highly desirable). Some of these are predictable and are described below, while others are impossible to predict. One of the key advantages of open-source software is the ability to modify and combine projects and tools to create something much greater than the sum of the parts. Sage just by itself is an outstanding example of this principle.

The following details activities during this project that will make Sage an even better tool for teaching, learning and understanding undergraduate mathematics. With an extended sabbatical, I can tackle this project with the confidence that it can have my full attention and be brought to a successful conclusion, while still being able to fully explore unanticipated possibilities that will arise.

Sage-Enhanced Textbooks Over a year ago, I created a demonstration project to convert mathematics textbooks into the native format of Sage's worksheets. The content of this experiment was a primer on using Sage to study group theory, based on the open-source textbook I used for Math 433 (Abstract Algebra I). The conversion of a textbook to Sage worksheest brings a variety of novel benefits to a very mature technology, the college textbook. Starting with the widely-used authoring language LaTeX, which can be subsequently converted to jsMath in an automated way, one creates a Sage worksheet that provides extremely high-quality typography viewable in any web browser using Javascript (a widely used standard). Alone this is an improvement on other approaches to publishing mathematics on the web, such as inline images for mathematics. But as a Sage worksheet, the reader also has the full power of an extensive computational environment without leaving the text. For the author, this is an incredible opportunity to place example computations, illustrating the mathematics or Sage applications, right in the text, runnable with a single click and modifiable by the reader with no overhead. Further, Sage's "interact" framework makes it incredibly easy for authors to embed interactive demonstrations, directed easily by the reader with buttons, sliders, and checkboxes, harnessing the full power and complexity of Sage under the hood. Finally, using tools already integrated into the Sage notebook, the reader can easily add high-quality annotations to their own copy of the text.

So armed with only a web browser, and a free copy of Sage (local or on a server), students can view their textbook, thoroughly explore new ideas with Sage code or interacts, learn new Sage commands, and then save all of their modifications, experiments and notes. The economics of a commercial textbook that can be easily copied, modified and saved are even more difficult than the challenges already confronting the textbook industry. This project will therefore concentrate on open-source texts. The obvious guinea pig is my open-source linear algebra text. This text has complete content and an infrastructure in place for explanations of computational aids, and so it will be very natural to integrate a complete description of Sage's outstanding and mature support for linear algebra and to present the book as a Sage worksheet.

Tom Judson's *Abstract Algebra: Theory and Applications* is another candidate for a supplement or enhancement, especially given the headstart I have with my primer on group theory. This is another mature textbook, which has just recently been released as open-source, and already has proved very popular (used at Berkeley, U of Portland, Grinell, Boston College). I am very familiar with this text, since I used it in the Fall 2008 semester and am using it this academic year in Math 433–434. It is the basis of the Sage group theory primer I used for my textbook-conversion demonstration project. I have been helping Tom Judson with his release of the text as open-source, through the `abstract.pugetsound.edu` web site. There is a nascent effort within Sage to improve the integration of the included GAP package for group theory, through a libGAP C interface, which will lead to further improvement in the capabilities of Sage for the topics in Math 433.

Bogart's *Combinatorics Through Guided Discovery* is an open-source textbook in combinatorics that teaches the subject through a series of roughly four hundred exercises. Ken Bogart died unexpectedly near the conclusion of writing this text, so his colleagues and students at Dartmouth College are anxious to see the book maintained and perhaps extended. Thus the decision to release the book as open-source. It would be natural to teach both Sage and combinatorics by adding explanations of relevant Sage commands to conduct the explorations that would illuminate the exercises. Professor David Neel is on the faculty at Seattle University, a former Ph.D. student of Bogart's, UPS Class of 1995, and a student in our first offering of Math 338, Combinatorics. David and I have had discussions about collaborating on revisions to Bogart's text, with David finishing off some of the content while I work on Sage additions and the conversion to a Sage worksheet. Serendipitously, David has a sabbatical during the 2010–11 year.

So there are three phases to textbook conversion — first creating the necessary technical tools to easily author mathematics textbooks and, in an automated way, convert them to the Sage worksheet format. My demonstration project very clearly shows the feasibility of doing this successfully. There is still work to do with the tex4ht converter, the SageTeX macro package, and support for stable cross-worksheet linking in the Sage notebook. Moreover, making these tools useful for other textbook authors will require additional work and documentation, perhaps in the form of a Sage optional package. Enhancing all three textbooks mentioned above could conceivably consume more than an entire academic year simply with the creation of new content. So there will probably need to be some selectivity in this phase, relative to related activities. The third phase of this project would be to educate and train other authors to create or modify open-source mathematics textbooks. This would happen at some time after the conclusion of the leave, possibly as the theme of a Sage Days workshop or a mini-course at the national mathematics meetings.

Enhancing Sage for an Undergraduate Audience Sage is a powerful tool for both research and education. One of my favorite features of Sage is the integration of so many mature projects for diverse specialized areas of mathematics. For years, I have wanted to use GAP in my abstract algebra course, but have never been able to surmount the elaborate syntax in a way that I thought the experience would be useful for my students in an introductory course. With Sage providing a simpler (and broadly applicable) syntax for accessing the power of GAP, I was finally able to successfully add a computational component to our course. My first major programming contribution to Sage was motivated by providing alternative output for some of the linear algebra commands that would be more understandable to a new student of the topic. So I was able to enhance the efficiency and code readability for the distinctions between a left and right kernel by refactoring portions of the linear algebra code.

Explaining Sage to students, or explaining undergraduate mathematics with Sage commands, often suggests the need for new commands, new features, alternative output, or improvements to the documentation. So as an integral part of creating Sage-enhanced textbooks, I propose to continue my work as a Sage developer with an emphasis on improving the experience of undergraduate students and teachers. I am intrigued by the possibilities afforded by the libGAP project for group theory, I am intimately familiar with the linear algebra code, and have begun to contribute to the impressive graph theory library (which is a subset of the broader area of combinatorics). As a contributor to Sage, I understand the importance of contributing to documentation, carefully reviewing code, and contributing thoughtfully in the forums, and promoting Sage to new users and developers. But nothing can replace the understanding of the code gained by contributing changes and working closely with the other developers, especially when it happens that another developer's expertise is the missing piece to one of your own puzzles. So this will improve my own understanding of the areas of Sage I will be writing about.

Interacts Sage has a simple and powerful method for quickly creating interactive demonstrations, often with graphical output. These could be a powerful addition to any mathematics textbook, providing students a way to explore a new concept with the power of Sage, but without having to learn any Sage code. Teaching multivariate calculus in the Spring 2009 semester I created many interacts for presenting three-dimensional examples in a way that we could explore vector calculus, plots of surfaces, level curves, tangent planes, approximating surfaces and other topics from a variety of viewpoints. I also used interacts created by Sage developers Jason Grout and Robert Robert Mařík, who were teaching the same course. A more polished and complete collection of these interacts would be a very useful addition to a multivariate calculus textbook. Michael Corral's incredibly good open-source textbook, *Vector Calculus*, is an ideal candidate for such an addition.

This also suggests two interesting projects for student summer research projects. One would create Sage interacts that automatically construct random, but tractable, student exercises in linear algebra. The other would be to extend Michael Corral's upcoming open-source textbook on trigonometry with Sage interact demonstrations. These projects would be an ideal mix of mathematics and programming for students with the right preparation.

Moodle and WeBWork Moodle is the open-source course-management system we are now using at UPS. WeBWork is an open-source system for presenting exercises to students online and providing immediate responses and feedback, and is explicitly designed for mathematics and science.

On the surface, it would seem obvious to couple Sage and WeBWork, and other Sage developers are interested in integrating Sage and Moodle. Mathematics software, textbooks, course management systems, homework delivery systems — all open-source — there is sure to be some combination that could totally revolutionize how mathematics courses are taught, especially if the "intelligence" of Sage was available to each component. I do not yet know what that combination is, but there is no denying the possibility. As part of creating Sage-enhanced textbooks I would be involved in these discussions and perhaps my efforts would be a part of some grander integration of these systems.

Teaching with Sage I know from writing my linear algebra textbook that many of the best ideas come directly from experiences teaching the course. I have made Sage a required part of my Math 433-434 courses this academic year, and so it will be with linear algebra in the Spring 2010 semester. This semester I have organized a weekly evening seminar devoted primarily to giving interested students an introduction to Sage. As Sage continues to evolve and improve, I expect to use it more and more in lower-level courses.

The African Institute for Mathematical Sciences (AIMS) educates students from throughout Africa, who spend nine months in residence to earn a postgraduate diploma, and then return to their home countries. The Institute is committed exclusively to open-source software so their students learn tools they can afford to use in their subsequent positions. They have recently solicited for short-term faculty to teach a course in Sage. The Institute has only recently finalized its program for Fall 2009, so our preliminary discussions about Fall 2010 have not progressed very far. If the timing works out, relative to my family responsibilities, I would find it very valuable to teach a short intense course on Sage while designing Sage-enhanced textbooks and contemplating similar changes in my own courses.

Recognition for Teaching, Professional Growth and Service This proposal is expected to describe recognition for ”outstanding teaching, professional growth and service.” I was the 2003 Regester Lecturer, and award with a selection criteria that asks for someone “who epitomizes excellence in teaching, scholarship, and leadership.” As a result of a periodic evaluation, I was named a Distinguished Professor for the 2001–06 period, another award based on XXXX. My most recent evaluation by the Faculty Advancement Committee (XXXX) read:

> Exact quote here..........

In the areas of university service, I have regularly served on university, science division and departmantal committees. Currently I am on the Professional Standards Committee and Vice Chair of our department. I recently completed two years work as Faculty Liaison to the Thompson Hall remodel project, with one intense year spent helping get a troubled project back on track. I currently serve as a faculty adviser to the Library's Scholarly Communication initiative, which of course is closely allied with this proposal

Professional Goals As it was with my decision in September 2008, my professional goals will continue to revolve around the Sage project. More generally, I am an interested participant in the “open educational resources” movement, so open-source textbooks and Sage will continue to be important to me. Contributing to Sage requires research-level abilities in mathematics, an understanding of algorithms from a computer science perspective, computer programming abilities, and is an excellent demonstration of the power of open development and open communities. It is a perfect fit for my interests and my position as a mathematics professor at a liberal arts college.

Subsequent to this leave, I will be better able to integrate Sage into my courses, both through explicit creation of new materials and through the experience gained. I will be a resource to the mathematics and computer science department, other science departments and other quantitative disciplines in the use and deployment of Sage in the classroom.

I could envision our department someday seeing the utility of a course in Sage and related tools, as a necessary part of an undergraduate mathematics major's curriculum. (I am already convinced.) So the experience gained from working closely on a project with direct educational goals should be very useful when curricular discussions turn this direction.

At the conclusion of this leave, it will remain to distribute to other authors the tools for conversions to Sage-enhanced textbooks. I will also have many new ideas for similar independent or summer research projects for students, such as the two already proposed above.

Conclusion I have learned much new about mathematics, an incredible amount about programming, and have borrowed many of Sage's open development processes and tools for my own work on open-source textbooks. I believe that my continued, heavy involvement in Sage will continue to provide direct and indirect benefits for my own professional growth as a mathematician, programmer, textbook author and most importantly, teacher of undergraduate mathematics. Furthermore, greater use of Sage will help us create students with an even deeper understanding of mathematics.