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
Due: October 12, 2009
DRAFT September 2, 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 to several interconnected projects, in collaboration with members of the Sage community, expanding and improving the usefullness of Sage in the undergraduate classroom.

I became aware of Sage in late 2007 through some correspondence with a Swiss undergraduate contributor to my open-source linear algebra textbook. I was immediately struck by the idea 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. We were fortunate to have William Stein, Sage's project leader, visit us to give a departmental seminar on Sage in Spring 2008. 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 use (and learn Sage) at every opportunity. At the end of that fall, I contributed to the Sage project a 14-page primer on using the program to learn group theory, a result of integrating Sage into my Math 433 course. Soon I was discovering and reporting bugs in the Sage forums, and suggesting additions to the linear algebra code for use in my linear algebra textbook. It was not long until I was learning the Sage development process and contributing code and fixes to Sage itself. I gave a talk at Sage Developer Days 1 in June 2008 about my linear algebra textbook and plans to tightly integrate Sage, and gave a Sage linear algebra tutorial at Sage Days 15 in May 2009. At some point in this arduous and invigorating journey I became a "Sage developer," though nobody rang a bell at that moment.

I participate regularly in the Sage forums on Goggle Groups and especially the "sage-edu" group, mostly devoted to topics related to using Sage in undergradute mathematics courses. In the "sage-devel" group, devoted to the actual design and development of the program, when appropriate I try to represent the use of the program by students or teachers, rather than just researchers. One of the key strengths of Sage is that it is a powerful tool for research and for teaching and the community is equally supportive of both purposes.

Sage has become a very big part of my courses and my professional growth. As I introduce it to more students, and in more introductory courses, I also hope to prepare some students to make it a central part of their independent projects and summer research projects. I have learned an incredible amount of new mathematics through my own contributions, and by participating in Sage's rigorous review process verifying the work of other contributors. (Who knew there was an $O(n^4)$ algorithm to compute the characteristic polynomial of a matrix *without ever dividing*!) This proposal is about tackling five large interconnected projects that will make Sage an even better tool for teaching — contributing new features for use by students at UPS and around the world, while further leveraging the contributions of the several hundred other Sage developers. The projects described below are ones that cannot be easily accomplished through sporadic work in the evening and on the weekend, and they will also rely on coordinated work with other Sage developers and other open-source projects. With an extended sabbatical, I can tackle these larger projects with the confidence they can have my full attention and be brought to a successful conclusion, while being able to fully explore new possibilities that arise from the connections between the projects.

Sage-Enhanced Textbooks Over a year ago, I created a demonstration project to convert mathematics textbooks into the native format of Sage's worksheets. This 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 with no overhead. Further, Sage's interact framework makes it incredibly easy for authors to embed interactive demonstrations, directed by the reader with buttons, sliders, and checkboxes, harnessing the full power and complexity of Sage under the hood. Finally, the Sage notebook has powerful tools for adding text to a notebook and excellent support for the LaTeX language, so the reader can easily add high-quality annotations to their own copy of the text.

This approach has been tried before with calculus texts in the *Calculus & Mathematica* curriculum, which is powered by *Mathematica*. The difference now is that the software is free and the interface requires only a standard web browser. So for starters, students can work with a local copy or anywhere there is an internet connection, rather than needing to visit the computer lab to study. Also, the economics of a commercial textbook that can be easily copied and easily modified 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. This is another mature textbook, which has been very recently released as open-source, and already has proved very popular (used at Berkeley, U of Portland, Grinell, Boston College). From the Fall 2008 semester I have a good start on describing the use of Sage's group theory commands and will expand my knowledge of Sage in Spring 2010 when I first use the book in Math 434 (second-semester Abstract Algebra). There is a nascent effort within Sage to improve the integration of the included GAP package, through a libGAP C interface, which will lead to further improvement in the capabilities of Sage for the topics in Math 433 (first-semester Abstract Algebra).

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 this activity — 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 relies on other open-source software, with some minor custom tools. These standard tools should be easy to modify and extend to make the whole process routine for other authors. Enhancing all three textbooks mentioned above could require an entire academic year of content-creation. So there will need to be some selectivity in this phase, relative to other projects. The third phase of this project would be to educate and train other authors to create or modify open-source mathematics textbooks using the tools created. This could be the theme of a Sage Days workshop or a mini-course at the national mathematics meetings at some time after the sabbatical.

Enhancing Sage for an Undergraduate Audience Sage is a powerful tool for research and for 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. 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 my introductory course.

In linear algebra, commercial competitors like MATLAB prefer a numerical approach to the topic and sometimes provide misleading or incorrect answers when students expect exact results. But Sage's extensive support for computations with rational numbers, provided by packages like IML and PARI, fits nicely with the aim of our course giving a first exposure of the topic to mathematics and science majors. And if they need accurate computational tools later in their careers as scientists, Sage provides that capability through packages like SciPy and NumPy. Still, Sage sometimes presents results in formats better suited for a research project rather than a student new to the subject. Sage's answers are not incorrect, there is just room to provide alternatives more appropriate for the beginner. My first major programming contribution to Sage was to enhance the efficiency and code readability for the distinctions between a left and right kernel, with an eye towards avoiding misunderstandings in this area for students.

So a second project, definitely intertwined with the textbook content creation described above, is to continue to contribute code and reviewing to the Sage code base itself, with the purpose of enhancing the student (and faculty) experience in undergraduate courses. I'm intrigued by the possibilities afforded by the libGAP project, 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.

Interacts As mentioned above, Sage has a simple and powerful method for quickly creating interactive demonstrations, usually with graphical output. Teaching multivariate calculus in the Spring 2009 semester I found this an extremely valuable tool 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. Typically, I would create a demonstration the evening before class, post it on the Sage forum, and overnight Jason Grout or Robert M would improve on it before I used it in class the next day. It would be worthwhile to go back and improve these demonstrations further and organize them into a collection that others could profitably use in their own courses, and to perhaps initiate similar collections for other courses, such as linear algebra.

Moodle and WebWork Moodle is the open-source course-management system we are now using at UPS. I know next to nothing about it. WebWork is an open-source system for presenting exercises to students online and providing immediate responses and feedback, with explicit support for mathematics and science, again in large measure through support for LaTeX and jsMath (note: not sure about accuracy of last phrase). There is a community project to create vast libraries of mathematics exercises, the National Problem Library, which now has 12,000 problems. Again, I am largely ignorant of WebWork's mission and capabilities. But I noticed a genuine enthusiasm among people I met, and presentations I attended, at the national meetings this past summer in Portland. Indeed, the primary collegiate professional society for mathematics, the Mathematical Association of America, has now taken a leadership role in the WebWork project.

There are repeated discussions in the Sage forums about integrating Sage with Moodle. On the surface, it would seem even more obvious to couple Sage and WebWork. There is already some integration between Moodle and WebWork (e.g. WebWork scores can migrate automatically to Moodle). Open-source math software, open-source textbooks, open-source course management systems, open-source homework delivery systems: there is sure to be some combination that could totally revolutionize how mathematics courses are taught. I do not yet know what that combination is, but there is no denying the possibility.

There have been discussions within the Sage community about a workshop bringing together members of the Sage, Moodle and WebWork projects for several intensive days learning and understanding each others' systems with an eye towards just such an integration. So as another project, I would like to learn as much as possible about these systems to the point where I could understand the technical challenges of integrating one with the other. At a minimum, I would be able to comment, test and critique efforts in this area, and perhaps be able to take a more direct role relative to other projects described here.

Teaching with Sage I am steadily increasing my use of Sage, and expectations for my student's use of Sage, in courses such as Abstract Algebra, Linear Algebra and Multivariate Calculus. As Sage continues to evolve and improve, I expect to use it more and more in lower-level courses. As part of my planning for this leave, I have been in contact with representatives of the African Institute for Mathematical Sciences (AIMS) about their solicitations for a Sage Lecturer. AIMS students come from throughout Africa, spend nine months in residence studying for a postgraduate diploma, and then return to their home countries. The Institute is committed to giving its students experience exclusively with software that is open-source so they know tools they can afford to use in their subsequent positions. Their courses are short and intense (similar to the Colorado College model), so a short visit (six weeks?) might be an excellent opportunity to gain additional skills in presenting Sage to an undergraduate audience. A desire to have family members accompany me on such a trip might complicate matters, and it is not clear such a position will materialize, but it is part of my planning. At the time of the last reply, AIMS was still trying to fill the position for October 2009, so there was not much urgency on their end to consider next fall.

This semester I have organized a weekly evening seminar devoted primarily to giving interested students an introduction to Sage. I also expect to use all three of the above textbooks in my own courses, as these topics comprise the majority of my non-calculus teaching. Improved Sage interacts will be extremely useful in my teaching of multivariate calculus. 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.

Conclusion Big finish explicating all of the ways this proposal meets the purposes detailed in the official description of the Lantz Fellowship.